PALATABLE GAME DEVELOPMENT: LESSONS LEARNED FROM FOODBOT FACTORY, ACCESSIBILITY, AND AUDIO GAMES

A GAME DEVELOPMENT PERSPECTIVE ON THE CREATION OF FOODBOT FACTORY, RECOMMENDATIONS FOR STUDENT GAME DEVELOPERS, AND A FOODBOT FACTORY AUDIO GAME FOR PEOPLE WITH LOW VISION AND BLINDNESS

BY

ROBERT SAVAGLIO

A Thesis Submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the Requirements for the Degree of

MASTERS OF SCIENCE IN COMPUTER SCIENCE

FACULTY OF BUSINESS AND INFORMATION TECHNOLOGY
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY
(ONTARIO TECH UNIVERSITY)

OSHAWA, ONTARIO, CANADA

April 2023

THESIS EXAMINATION INFORMATION

Submitted by: **Robert Savaglio**

Masters of Science in **Computer Science**

Thesis title: Palatable Game Development: Lessons Learned from Foodbot Factory, Accessibility, and Audio Games

An oral defense of this thesis took place on April 11, 2023 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Peter Lewis
Research Supervisor	Dr. Bill Kapralos
Examining Committee Member	Dr. Alvaro Joffre Uribe Quevedo
Thesis Examiner	Dr. Loutfouz Zaman

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

To address a lack of serious game work focusing on the development itself, this thesis describes recommendations for developers based on the development of *Foodbot Factory*, a nutrition-based serious game. Additionally, to push audio game and accessibility research forward, the development and testing of a *Foodbot Factory* audio game are described. Twenty participants played through *Foodbot Factory*'s protein foods module with visuals removed, relying on audio alone. The results of the study were overall positive, all players were able to complete the module in the allotted time and spoke positively about their experience. Feedback from participants highlighted a need for better tutorialization and higher-quality voiceovers. Future work with *Foodbot Factory* should include playtests with players who experience low vision and blindness to fully understand the nuances of their needs and ensure that *Foodbot Factory*'s accessibility features are as effective as possible.

KEYWORDS: serious games, accessibility, audio games, nutrition education, foodbot factory

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ontario Tech University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize Ontario Tech University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

The research work in this thesis was performed in compliance with the regulations of Ontario Tech's Research Ethics Board/Animal Care Committee under reference number 14879.

Rob Savaglio

STATEMENT OF CONTRIBUTION

Some ideas presented in this thesis have appeared previously in the following publications:

- Jacqueline Marie Brown, Robert Savaglio, Graham Watson, Allison Kaplansky, Ann LeSage, Janette Hughes, Bill Kapralos, and JoAnne Arcand. Optimizing child nutrition education with the Foodbot factory mobile health app: Formative evaluation and analysis. Journal of Medical Internet Formative Research, 22(4), 2020. ISSN 14388871. doi: 10.2196/15534.
- Hannah M. Froome, Carly Townson, Sheila Rhodes, Beatriz Franco-Arellano, Ann LeSage, Rob Savaglio, Jacqueline Marie Brown, Janette Hughes, Bill Kapralos, and Jo Anne Arcand. The Effectiveness of the Foodbot Factory Mobile Serious Game on Increasing Nutrition Knowledge in Children. Nutrients, 12(11):2-15, 2020. ISSN 20726643. doi: 10.3390/nu12113413.
- Robert Savaglio, Jacqueline M Brown, Bill Kapralos, Ann LeSage, Beatriz Franco Arellano, Janette Hughes, and JoAnne Arcand. Enhancing the accessibility of serious games: A case study with foodbot factory. In 12th International Conference on Information, Intelligence, Systems & Applications (IISA), pages 1-4. IEEE, 2021.

The *Foodbot Factory* serious game discussed in this thesis was developed by an interdisciplinary team of experts at Ontario Tech University in Oshawa, Canada. The author was a consistent member and leader of the development team, from the initial meetings in September 2017 to April 2021. The majority of the gameplay, code, dialogue scripts, and audio was designed and implemented by the author.

The updated audio game version of *Foodbot Factory* discussed in Chapters 5 and 6 was developed by the author.

ACKNOWLEDGMENTS

Thank you to Dr. Bill Kapralos, Dr. Peter Lewis, Dr. Alvaro Joffre Uribe Quevedo, Dr. Loutfouz Zaman, Dr. Meg Freedman, The Bodyzone Team, Bird Brain Games, and anyone else who supported this work and inspired me to become a game developer.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 OVERVIEW

This thesis provides a game development perspective for the development of *Foodbot Factory*, a serious game for nutrition education, and details the development of accessibility features for players who experience low vision or blindness. Serious games are popular educational tools that leverage engagement and interactivity to promote learning [76]. Thus, *Foodbot Factory* is a serious game used to promote nutrition knowledge for children in grades 4-6, developed by an interdisciplinary team of experts at Ontario Tech University in Oshawa, Canada. The author was a consistent member and leader of the development team, from the initial meetings in September 2017 to April 2021. The majority of the gameplay, code, dialogue scripts, and audio was designed and implemented by the author. In this chapter, the motivations for this thesis are described, including a lack of research detailing the development processes behind serious games, the need for accessible games, and the need for further audio games work. A summary of each chapter is also provided below.

1.2 MOTIVATION

First, while serious games have been widely discussed in recent literature, the development methodologies and implementations of these games are often overlooked in lieu of discussing effectiveness [66, 114, 32], categorization of serious games [99], the adoption of serious game approaches in other research [33], and gamification theory [78]. Game development methodologies including project management methods, communication processes, tool recommendations, development pitfalls, and other workflow-related advice are often absent in serious game work. For example, in a recent literature review of serious games for medical education, of 65 included articles only seven described an iterative development process [87]. Furthermore, despite the complexities associated with developing serious games, the literature reveals little with respect to courses regarding serious games development at the post-secondary level and no existing courses or lesson plans focusing on serious game design and development at the K-12 level [75]. Recent work for the serious game *Foodbot Factory* has focused on the educational and health aspects of the project. Brown et al. detail the iterative development of *Foodbot Factory* using the Obesity-Related Behavioural Intervention Trials (ORBIT) model [41]. Motivated by the risks of poor diet quality in Canadian children, Brown et al. position *Foodbot Factory* as a mHealth intervention (mHealth, short for mobile health, is the practice of medicine and public health supported by mobile devices) [41]. Froome et al., detail a five-day pilot study of *Foodbot Factory* focused on evaluating its impact on nutrition knowledge. Although, these works have explained much of the iterative user-based development of the *Foodbot Factory* project, they were written from the perspective of mHealth research. Thus far, the game development approaches used to develop *Foodbot Factory* have gone undocumented. Since *Foodbot Factory* was developed primarily by students, much of the development was challenging and the process contained useful learning experiences that should be shared for other student developers and researchers to learn from. This work aims to address this gap by providing a detailed discussion of the development of *Foodbot Factory* from a game development perspective and provide specific practical recommendations to developers.

Second, most video games and serious games, including *Foodbot Factory*, are not completely accessible and often cannot be played by individuals with disabilities. Greater work is needed to encourage and facilitate the development of accessible games. Heron describes the game industry as a two-tier system; for players that can enjoy and use games freely, and players that are excluded from full participation [70]. For example, an individual with low vision may not be able to play a video game designed for interaction through visual cues. This implies that people with disabilities cannot enjoy the full social, recreational, and educational benefits that games offer [74]. Awareness of game accessibility amongst game developers and researchers has increased, for example, a popular set of guidelines, the Game Accessibility Guidelines (GAG) [1], have been created by a collaboration between a group of game studios, specialists, and academics to provide a heuristics based tool for assessing the accessibility of a game. However, greater work is required, particularly work that details the development of accessible games using the GAG and other similar tools. In Canada, the Web Content Accessibility Guidelines (WCAG) are used to ensure that web content is inclusive in design [10]. In the province of Ontario, under the Ontario Education Act [9], it is each school board's responsibility to provide an inclusive school climate, which includes providing special education programs and services to students with disabilities. Since *Foodbot Factory* is intended to be played in Canadian classrooms, it must comply with both the WCAG and the GAG and is a candidate to address this need for literature focused on developing accessible games.

Third, the audio game community is eager for quality audio games and there is a lack of research detailing audio game development. Audio games are digital games akin to video games, however, their primary output is audio rather than video and can be played with audio alone. Thus, the audio game community is largely populated by individuals with low vision and blindness. In a blog post titled "Audio Games and The Never Ending Cycle of Mediocracy" [73] the website creator and manager of BlackScreenGaming.com, known online as Smoke J, argues that a lack of thoughtful game design by developers is contributing to the release of many poorly designed audio games, as many recent audio game releases are just copies of old games. This sentiment is echoed in academic research. For example, with the rise in popularity and complexity of three-dimensional video games in the 1990s, the younger generation of players with low vision or blindness are having an increasingly difficult time participating, and a lack of guidelines and industry conventions for audio games has led to simplistic designs that are not very engaging, often due to audio games being developed solely by small teams of developers or researchers [58]. This need for quality audio games and design standards has sparked interest in academic research and work creating and summarizing audio game design recommendations does exist, however, research describing their use and testing is limited [38]. Until audio games and accessible features in video games become widely available, audio game research must continue and should be utilized to inspire accessibility features in serious games and video games in general. This thesis helps to address this need by using audio game research as the basis for updating *Foodbot Factory* with accessibility features.

1.3 THESIS CONTRIBUTIONS

This thesis presents two main contributions.

First, this thesis presents a developer perspective on the development of the *Foodbot Factory*, a serious game focusing on nutrition education for children in grades 4-6, developed by an interdisciplinary team of experts at Ontario Tech University in Oshawa, Canada, of which the author was a consistent member and leader of the development team. This development description is presented in Chapter 3 as a sequence of iterations and describes the project management tools and feature implementations developed throughout the project. Recommendations for serious game developers based on this experience are discussed in Chapter 4.

Second, this thesis aims to answer the following:

> Can audio game research be used to update *Foodbot Factory* with accessibility features to support players with low vision and blindness?

The development and testing of the *Foodbot Factory* audio game is described in Chapters 5 and 6. The goal of this work is to determine if recommendations presented in other audio game research can be used as a basis for updating serious games with accessibility features for players with low vision and blindness.

1.4 ORGANIZATION

The following chapters of this thesis are presented as follows:

- Chapter 2: First, this chapter discusses an overview of the game development methodologies used for developing *Foodbot Factory*, including agile and scrum. Next, current serious game research is discussed, including serious game research in early education. Finally, game accessibility and audio game research is discussed.
- Chapter 3: This chapter discusses the iterative development of *Foodbot Factory* from a developer perspective. First, a summary of *Foodbot Factory* and the project's motivation is provided. Next, the development approach and iterations are summarized.
- Chapter 4 This chapter expands on the lessons learned during the development of *Foodbot Factory* and presents them as discrete recommendations for serious game developers.
- Chapter 5: This chapter provides an overview of the accessibility updates developed to create the *Foodbot Factory* audio game. The chapter provides an accessibility analysis of *Foodbot Factory*, then provides a description of each development iteration used for updating *Foodbot Factory*.
- Chapter 6: This chapter describes a usability study conducted to examine the *Foodbot Factory* audio game. The study design and methods are described, then the results are presented and discussed.
- Chapter 7: This chapter describes the main contributions of this thesis and its corresponding work. The thesis statement is presented once again and is discussed using the full context of the work described in this thesis.

2

RELATED WORK

2.1 OVERVIEW

This chapter provides an overview of concepts and previous work relevant to this thesis. The concepts discussed below include game development methodologies, serious games, game accessibility, and audio games.

2.2 GAME DEVELOPMENT METHODOLOGIES

Foodbot Factory was developed following an iterative approach, using ideas and tools related to game development methodologies, such as Agile and Scrum. Brief overviews of these methodologies will provide context for the project management processes and design decisions made throughout the development of *Foodbot Factory*, discussed in Chapter 3.

2.2.1 *Agile*

Agile development is a form of lightweight, iterative, user-based project management meant to counter long-form planning methods such as the Waterfall method. Agile is generally favorable for small teams of developers [44] and takes a full development cycle of the Waterfall method (conception, design, implementation, testing, release), and repeats it in shorter iterative cycles. Instead of spending many months or years on one plan and development cycle, each iteration in an Agile framework can take only weeks or a few months at most. This allows developers to design, implement, and test features in bursts, enabling developers to be flexible by implementing stakeholder feedback after each cycle. The key here is that stakeholders are involved in the development process as early as possible, and throughout the process, so they can help shape the product to best fit their needs and expectations before designs are locked into place.

Agile development contains two main categorizations used for project management: i) User Stories, and ii) Epics. These categorizations allow project managers to organize tasks congruently with user-based iterations [113].

User stories are glorified descriptions of features to be developed and are written in a user-based phrase structure. These phrases describe a feature from a user's perspective. For example, a user story for interacting with in-game dialogue could be "As a player, I'd like to tap anywhere on the screen to advance the dialogue". The "user" may differ in some circumstances; user stories related to development tool creation may be written from another developer's perspective. For example, a user story for a dialogue tool may be, "As a scriptwriter, I'd like to add emojis to the script to cause the characters to show different emotions as they say their lines". Generally, user stories are to be completed within one iteration cycle but can be broken down into sub-tasks if needed [113].

Epics are a collection of user stories that combine to create one larger feature. Epics generally take a lot of time to complete over multiple iterations, allowing them to be shaped by user feedback at the user story level. Along with other user stories, the two dialogue-based user story examples in the quotations above could be organized into two separate Epics: "Front-end Dialogue" for the player interaction and "Back-end Dialogue and Tools" for the developer script writing. However, the scope of epics can vary based on team size and organizational methods. These example epics make sense if different development teams are focusing on the back-end and front-end development separately or at different times. However, for *Foodbot Factory* since only one to four game developers were working on the project at one time, a single "Dialogue System" epic is sufficient.

2.2.2 *Scrum*

While Agile provides developers with a lightweight methodology for organizing the goals and tasks of a project, it does not clearly dictate how the methods should be put into practice and tracked over the course of a project [93]. Thus, many Agile-based project management frameworks, such as Scrum, have emerged over the last few decades that take the ideas of Agile and enhance them with practical techniques.

Scrum is an Agile-based framework for facilitating collaboration, organizing tasks for each iteration, and tracking the team's efficiency. Scrum divides work into roughly two to four-week iterations, called sprints. Sprints start with a short planning meeting where the team selects user stories or features from a prioritized list called the project backlog. The team then decides which tasks are required for each feature and adds them to a sprint backlog with an estimated time of completion. Features and tasks should only be added to the sprint backlog if the team decides they are achievable during the sprint. The goal of each sprint is to create a usable iteration of the product and allow the team to easily pivot towards emergent ideas as they discover what makes the product fun or effective [93].

During sprints, daily Scrums are used to help the team track their progress. A daily Scrum is a short meeting held each morning where team members take turns explaining their progress, their current plan for the day, and any problems they are currently facing. The daily Scrum usually takes about 15 minutes and is often referred to as a stand-up meeting or simply a stand-up. Stand-ups were created to encourage short, efficient meetings by having all team members physically stand-up while providing their updates, to avoid getting comfortably off-topic. These days, not all stand-up meetings are performed standing, especially when online, but the succinct format remains and teams are encouraged to find a format and frequency that works best for their team [100].

To ensure these sprints and stand-up meetings run smoothly, a team member is appointed the duty of Scrum master. A Scrum master is an individual in charge of the Scrum-related project management. Scrum masters should have the knowledge needed to coach teams to better understand and use the Agile and Scrum methodologies in a way that fits their needs. Other responsibilities of the Scrum master include: i) monitoring progress, ii) facilitating reviews and retrospectives, iii) helping stakeholders and teams communicate, and iv) managing sprint organization and logistics. Classically, a full-time Scrum master is not a member of a development team who works on tasks during sprints, they instead manage the sprints of two to four teams at once. However, a developer on the team may be given the Scrum master responsibilities or some teams may not have an official Scrum master and will share the responsibilities [57].

2.2.3 *Kanban Boards*

Kanban boards are an Agile-based project management tool that helps developers organize tasks and visualize their work-in-progress [50]. Kanban boards generally consist of customizable columns where team members can add cards. These cards then contain descriptions of tasks, can be tagged or colour coated, and are moved between columns based on the state of the task. For example, a basic kanban board may contain the columns: backlog, in-progress, review, complete. As new tasks are created they are added to the backlog column. When a task is assigned to a developer it gets moved into the in-progress column. Once, the developer has completed the task, it gets moved into the review column, where it waits to be reviewed by another developer. Once the tasks implementation is deemed sufficient by the team, it gets moved into the complete column.

In physical collaborative spaces, kanban boards can simply be created on a wall or whiteboard using sticky notes, tape, and markers. Physical kanban boards encourage collaboration and are quite malleable. There are also many software applications or websites that have integrated kanban boards, such as Trello [18] or Jira [16]. These applications often have built-in tools that allow users to customize cards with extra information such as checklists, related files, images, and more.

Software tools also provide the ability to integrate with other tools in a project's pipeline. For example, Jira can connect with version-control software to update task progress in tandem with code submissions to a repository. Other benefits of software-based kanban boards include remote use from multiple devices, built-in efficiency tracking, and notifications.

2.3 serious games

The video game industry has been growing rapidly over the last few decades [80], in 2022, the video game market size in the United States was estimated to be 97.67 billion U.S. dollars [21]. Video games are ubiquitous and have become a primary source of entertainment and social interaction for many people. For example, the popular online multiplayer game Fortnite, has over 400 million registered users [15], and allows friends to connect and play together remotely.

Serious games, popular educational tools that leverage engagement and interactivity to promote learning [76], have paralleled the video game industry and are increasing in popularity. They have become popular in post-secondary institutions, particularly in the nursing and medical disciplines. For example, in an attempt to leverage the popularity and use of online games, Fleming et al. explored integrating mental health resources into online games [59]. The popular gaming company Ubisoft has recently explored serious games and created *Dig Rush*, which focuses on treating amblyopia (commonly known as a lazy eye) [4]. *Dig Rush* is the first serious game approved by the Food and Drug Administration (FDA) and requires a prescription [37].

Many studies have highlighted a gap between the theory learned in academic education and the practical skills required for their application in the real world, often referred to as the theory-practice gap [56]. For example, when applied to medical education serious games are a promising solution to this problem as they can introduce students to the intricacy of clinical situations without harming real patients [97]. They provide nursing and medical students with the opportunity to experience, learn, and practice important procedures and skills in a safe space before ever meeting their first patient [82, 36, 46, 56].

In addition to post-secondary and medical education, serious game research is being used in many other domains. Advergames, a type of serious game, are advertisement-focused games created to enhance brand awareness among consumers [98]. Marketers can embed brand messaging into their games. Game characters can wear logos or represent brand mascots, game mechanics can involve using the brand's products and services, and in-game messaging can simply include advertisements [45, 98]. Pro-social narratives can be used in advergames to improve the player's attitude of a company and the subtlety of brand messaging can play a role in the narrative's effectiveness [101]. Researchers are studying the impact that advergames have on players' recognition and attitude towards companies and their brands [106]. Often this research focuses on children and adolescents, who are frequently targeted by marketers, and may be more susceptible to the persuasive nature of advergames [105, 54, 60].

While serious games have been widely discussed in recent literature, the development methodologies and implementations of these games are often overlooked in lieu of discussing effectiveness [66, 114, 32], categorization of serious games [99], the adoption of serious game approaches in other research [33], and gamification theory [78]. Game development methodologies including project management methods, communication processes, tool recommendations, development pitfalls, and other workflow-related advice are often absent in serious game work. For example, despite the complexities associated with developing serious games, the literature reveals little with respect to courses regarding serious games development at the post-secondary level and no existing courses or lesson plans focusing on serious game design and development at the K-12 level [75]. Additionally, in a recent literature review of serious games for medical education, of 65 included articles only seven described an iterative development process [87]. To begin addressing this gap, Olszewski et al. propose a structured iterative framework for serious game development [87], but greater work remains for discussing and validating serious game development workflow advice.

2.3.1 *Serious Games in Early Education*

Pedagogy enhanced by serious games provides a practical means of encouraging active participation by students through play. Studies related to the use of serious games in the classroom have observed enhancements in learning [6, 111], or identify advancement in the cognitive capacity of students [79]. Often incorporating scaffolding approaches, with emphasis on interactive learning, serious games can also be used effectively for achieving behavioral and attitudinal changes in the classroom [88]. As Papanastasiou et al. point out,

> Gamers learn as they play, solving puzzles, learning strategies becoming immersed in their writing while staying within the constraints of the game world, transforming the writing classroom from workspace to gamespace. [88]

Serious games can encourage students to learn about certain topics [88]. For example, Carvalho et al. employed a city-simulator serious game to encourage younger students to pursue an engineering career path [53]. Serious games are also being used to delve into important life-related topics for children and adolescents such as bullying [43], or substance abuse [90]. However, although some studies report an increase in engagement, they do not provide conclusive results on meeting learning objectives [47, 72]. Also, a generational barrier may exist, indicating that teachers above 35 years of age lack previous positive exposure to serious games and their potential in education [77].

Searching Google Scholar, as it connects with a wide range of journals, for serious games articles, focused on preschool and elementary school education, three main themes are prevalent: i) Serious games focused on a specific school subject, ii) serious games for students with disabilities, iii) articles summarizing work in the field and the overall learning impacts of serious games.

Serious games articles focused on schools subjects include topics one may expect: Reading and writing [95, 81], mathematics [61, 62], programming [112], health [63, 41], and science [2, 48]. Some commercial games have also been used in the classroom. For example, the geography game *GeoGuesser*, which places the player in a randomly selected location in the world, and challenges them to decipher their location on a map using clues they find while exploring with *google street view* [67]. The widely popular game *Minecraft* has far-reaching potential in the classroom and has been used to teach topics such as language, storytelling, social skills, informatics, programming, project management, chemistry, and more [84].

Articles detailing serious games work for students with disabilities often focus on how games can be used to aid students with specific disabilities. For example, work by Drigas et al., explains that recent developments in serious games have been shown to improve the quality of life and functional independence of young learners with attention deficit hyperactivity disorder [55]. Work regarding developmental dyslexia allows for early identification in preschool before children are of age to be formally diagnosed [64]. Serious games focusing on autism spectrum disorder allow teachers to better prepare and adjust teaching strategies by analyzing which concepts and skills students are struggling to understand, in an environment that is motivating for both teacher and student [83]. Other recent studies have focused on disabilities such as diabetes [85], cerebral visual impairment [49], asthma [68], cerebral palsy [89, 115, 108], and coordination in survivors of childhood brain tumors [91].

2.4 GAME ACCESSIBILITY

Most video games and serious games are not completely accessible and often cannot be played by individuals with disabilities [74]. Heron describes the game industry as a two-tier system; for players that can enjoy and use games freely, and players that are excluded from full participation [70]. For example, an individual who experiences mobility issues may not be able to play a game unless it allows them to reconfigure the control scheme to a layout that fits their needs. This implies that people with disabilities cannot enjoy the full social, recreational, and educational benefits that games offer [74].

In the last few years, awareness of game accessibility amongst game developers has increased. Many have developed and updated their guidelines to encourage quick, efficient, and inclusive design decisions in the early stages of game development [52, 39]. A popular set of guidelines is the Game Accessibility Guidelines (GAG) [1], created by a collaboration between a group of game studios, specialists, and academics. The GAG is divided into three categories: i) basic, ii) intermediate, and iii) advanced. Each is based on a balance of reach (the number of people who benefit), impact (the difference made to those people), and value (the cost to implement). The three categories are further subdivided into sub-categories that relate to types of disabilities. These include: i) motor, ii) cognitive, iii) vision, iv) hearing, v) speech, and vi) general. The GAG is continuously evolving and the authors encourage open communication and feedback.

In Canada, the Web Content Accessibility Guidelines (WCAG) are used to ensure that web content is inclusive in design [10]. For example, content posted on government websites must be compatible with screen readers to ensure that people with low vision can easily access important information. In a 2018 gap analysis, Westin et al. analyzed the differences between the WCAG and the GAG, as the WCAG was not originally intended to be used for games [109]. While version 2.1 of the WCAG includes several improvements relating to games, they recommend that the WCAG are used in conjunction with the GAG to ensure games are as inclusive as possible [109]. In the province of Ontario, under the Ontario Education Act [9], it is each school board's responsibility to provide an inclusive school climate, which includes providing special education programs and services to students with disabilities. For example, since *Foodbot Factory* is intended to be played in Canadian classrooms, it must comply with both the WCAG and the GAG.

2.4.1 *Audio Games*

The lack of accessible games has caused some communities with similar lived experiences to form. For example, audio games have gained popularity among some individuals with low vision and blindness. Online communities such as AudioGames.net have also emerged to help players find suitable games, connect with other players, and make friends [3].

Audio games are digital games akin to video games, however, their primary output is audio rather than video, and can usually be played with audio alone. Not to be confused with audio-based video games such as *Guitar Hero* or *Osu*, which are themed around sound and music, audio games require no visual graphics to play. In fact, most audio games found on AudioGames.net can be played completely with audio alone [13]. Some audio games do contain visuals, but they are usually text or images that add additional usability to the game, rather than change the overall experience.

Audio games such as *A Hero's Call* allow players to freely explore full fantasy worlds and take part in turn-based battles, similar to popular video games such as *Final Fantasy* or *The Elder Scrolls V: Skyrim*. *A Hero's Call* uses an audio-radar system that allows players to listen to specific spatial audio cues to help orient themselves in the word. Other popular audio games, including *Swamp* and *Crazy Party*, allow players to connect and play together online.

> Journey to the city of Farhaven, a town under attack from a ruthless enemy. help the citizens by pursuing the main quest, or explore and see what other tasks you can find, since you will not be alone, four allies will join you in your mission, each with their own history, backstory and even their own options for quests[7].

Audio games have gained significant popularity among some individuals with sight loss. Online communities, such as Black Screen Gaming [5], have formed to help players find and play these games, as well as to connect players together and make friends. AudioGames.net serves as a main hub for finding audio games. It contains a large list of audio games in a format easily parsed with screen readers and text-to-speech (TTS). The list contains a description for each game and provides a link to the game's website where users can find and download it. AudioGames.net also maintains a forum [3] where the community can participate in discussions about games, find information about developing audio games, or just simply chat with others about topics unrelated to games. In a previous study [34], when asked about their thoughts on audio games and the community, one participant explained,

> There's three or four games that blind people really tend to gravitate towards right now. Swamp, Redspot, [Survive the Wild], those three. You're almost guaranteed to find people that you find on AudioGames.net and that you email or Skype with. It's a close-knit community. Everybody pretty much knows everybody or can remember them if they talk to them for a little bit [34].

While the audio game community seems to be a haven for certain gamers, the community has its share of struggles. In a blog post titled "Audio Games and The Never Ending Cycle of Mediocracy" [73] the website creator and manager of BlackScreenGaming.com, known online as Smoke J, presents two main problems contributing to the release of many poorly designed audio games.

First, he criticizes audio game developers for not putting in the "needed work before they write the first line of code" [73]. He explains that many new audio game releases are just copies of old games and that developers are partially to blame for creating poorly designed games. "[Developers] don't plan, [they] don't pre-balance, [they] don't conceptualize, and [they] do the bare minimum forethought" [73].

This idea is echoed in the developer's room on the AudioGames.net forum [3]. The room contains a pinned post titled "List of resources for programmers, developers, and more", which contains accessible programming resources for implementing audio games. However, the post lacks any resources related to proper game design techniques or any reference to audio game design guidelines. Posts in the developer's room are also overwhelmingly related to programming; users seldom post threads asking about tips for game design. In fact, not a single post about audio game design is found looking within 200 of the most recently posted threads in the developer's room.

Second, Smoke J argues that the players and the community are partially to blame for the continued release of poorly designed games. "Dissenting opinions against a game are almost always met with hostility from the other players. This is some form of petty tribalism I don't fully understand, but one which runs unchecked in the audio gaming community [73]." Smoke J, argues that this community reaction to new games encourages a cycle of unambitious audio games to be developed, praised, and released. "Before you know it there's a flood of the same thing being released where the only difference between any of them are sounds and names [73]."

The audio game community has also sparked interest in academic research, helping developers find new and intuitive solutions for equivalently accessible gaming [51]. The goal of equivalent accessibility in gaming is to ensure that players with disabilities can experience a game in a similar manner as other users [96]. Cairns et al. explain that games, unlike other media, are meant to be challenging, and therefore, developers should strive to reach the largest audience possible without diminishing their intended experience [42]. Similarly, Smith and Nayar explain that accessible games should strive to preserve both player intention and the efficiency of gameplay [96]. Unfortunately, developers often sacrifice one of these concepts to make a game accessible, prioritizing usability over user experience. For example, the audio game *Drive* [8], preserves fast-paced audio-guided racing at the cost of only allowing the player to press the up-arrow on the keyboard to drive forward, removing any meaningful sense of intention. Nonetheless, the audio game community is passionate and excited for more opportunities to play games [34] and online spaces such as Audiogames.net [3] and audio games research are excellent sources for inspiring accessible designs for games.

2.5 SUMMARY

While serious games have been widely discussed in recent literature, the development methodologies and implementations of these games are often overlooked. Specifically, there is a lack or work discussing serious game development [75, 87]. Addtionally, serious games are not completely accessible and cannot be played by everyone, including people who experience low vision and blindness [74, 70].

With greater discussion of serious game development, opportunities for injecting accessiblility design and features into serious games may become more apparent. Work detailing serious game development could be scrutinized and frameworks for development could be adapted to include processes for analysing accessibility, such as utilizing the WCAG [10] and the GAG [1] early in development. Additionally, greater work is needed to utilize accessibility and audio game research to better understand their effectiveness for serious games [109, 52, 39].

Therefore, to address these gaps the development of *Foodbot Factory* should be discussed and game accessibility guidelines such as the WCAG [10] and the GAG [1] should be used to explore specific areas of improvement for *Foodbot Factory*. Finally, audio games and audio game research can be used as inspiration for updating *Foodbot Factory* with accessibility features for people who experience low vision and blindness.

3 THE DEVELOPMENT OF FOODBOT FACTORY

3.1 OVERVIEW

Foodbot Factory is a serious game focusing on nutrition education for children in grades 4-6, developed by an interdisciplinary team of experts at Ontario Tech University in Oshawa, Canada. The author was a consistent member and leader of the development team, from the initial meetings in September 2017 to April 2021.

For clarity, when referring to the full research and development team, including students and professors from Health Sciences, Education, and Information Technology/Computer Science faculties, the "Bodyzone team" will be used. When referring to the three to four Game Development students working on the code, art, and implementation for the application the "development team" will be used.

Foodbot Factory consists of five modules intended to introduce topics related to the 2019 Canada's Food Guide [14]: i) Drinks, ii) Vegetables and Fruit, iii) Whole Grain Foods, iv) Animal Protein Foods, and v) Plant Protein Foods. Each module is independent from the other and thus a module can be played in any order. Each module uses engaging dialogue between nutrition scientists (Robbie and Rebecca), robots (Foodbots) and villagers to provide educational nutrition information to users based on Canada's Food Guide. In the Drinks module users learn why water should be a drink of choice, the health impacts of different sugary drinks and catch different types of drinks (e.g., water, milk). The Whole Grain Foods module exposes the user to how whole grains become refined and the nutritional differences between types of grain foods by having users interact with a whole grain kernel and sort whole grain foods from refined grain foods (e.g., whole grain and white bread). The Vegetables and Fruit module emphasizes the importance of eating a variety of vegetables and fruit and that juice is a sugary drink by requiring users to catch a variety of different vegetables and fruit (e.g., apple, zucchini) and avoid catching juice. The Plant Protein Foods module provides information on the health benefits of choosing plant protein foods more often and has the user prepare a meal using a plant protein food (e.g., beans). Finally, the Animal Protein Foods module educates the user on key nutrients (sodium, saturated and unsaturated fat) found in animal protein foods, their impacts on health and sort different animal protein foods (e.g., fish, processed meats). Each module is comprised of four types of interaction: i) dialogue sections where players learn about the topic through humorous conversation shown in Figure 1, ii) conversational quizzes that access understanding of the current topic shown in Figure 2, iii) short dialogue-based interactions that add variation to dialogue sections shown in Figure 5, and iv) mini-games that generally serve as a break from dialogue shown in Figure 3. The game is considered complete once each module has been completed and each modules takes 10 to 15 minutes to complete.

Figure 1: Example dialogue in *Foodbot Factory*.

As this project has been a collaboration between the Education, Health, and Information Technology/Computer Science faculties at Ontario Tech University, development approaches for the project have varied between each discipline. The approaches for developing the nutrition and education content of *Foodbot Factory*, using mobile health (mHealth) research, have been explained in detail.

Figure 2: A dialogue quiz in *Foodbot Factory*.

Figure 3: The *Food Drop* mini-game in *Foodbot Factory*.

Specifically, Brown et al. detail the development of *Foodbot Factory* using the Obesity-Related Behavioural Intervention Trials (ORBIT) model [41]. Motivated by the risks of poor diet quality in Canadian children, Brown et al. position *Foodbot Factory* as an mHealth intervention. Development and testing of each game module are described from the perspective of engagement and usability, with positive findings [41]. Froome et al. detail a five-day pilot study of *Foodbot Factory* focused on evaluating its impact on nutrition knowledge. The study was single-blinded, parallel, randomized controlled, conducted among children ages 8-10 years and children who used *Foodbot Factory* had significant increases in overall nutrition knowledge, compared to the control group [63].

Figure 4: The Food Log in *Foodbot Factory*.

Although these works have explained much of the iterative user-based development of the *Foodbot Factory* project in detail, they were written from the perspective of mHealth research. Thus far, most of the game development style approaches used to develop the technical aspects of *Foodbot Factory* have gone undocumented. Brown et al. describe five user testing sessions held at the end of each development iteration [41], which includes semi-structured observations recorded by members of the development team. While the results from these sessions are well described in relation to mHealth research and nutritional content, the developer perspective, including project management methods and specific feature implementations, has yet to be described. Therefore, the following is an account of the game development tools, implementations, and iterations used to develop the serious game, *Foodbot Factory* from the development team's perspective.

3.2 DEVELOPMENT APPROACH

Over the course of the project, the development team adopted an iterative approach using ideas from Agile methodologies. Scrum-style sprints were held every two weeks. However, since the development team consisted primarily of undergraduate

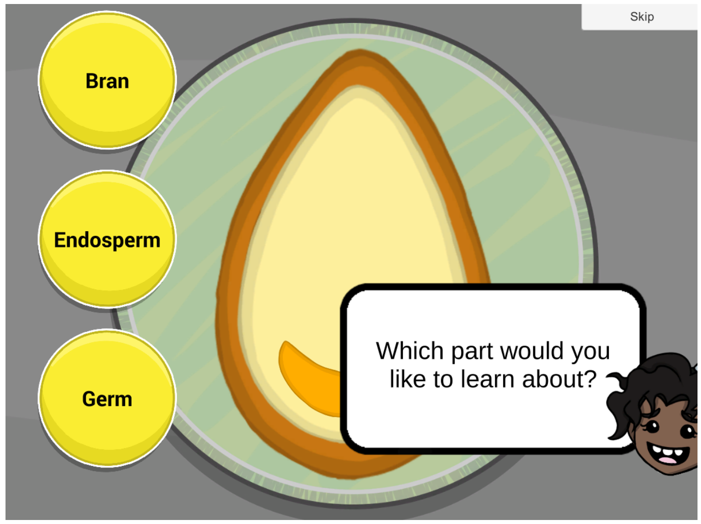

Figure 5: A dialogue interaction in *Foodbot Factory* that has players looking through a microscope to learn about a grain.

students working on a part-time basis, quick iteration cycles were not always feasible and the cycle was occasionally extended as needed. Tasks were organized using Trello's Kanban boards, structured with columns: Backlog, In Progress, In Review, and Complete. New tasks were added to the Backlog column after each sprint, keeping higher-priority tasks towards the top of the column. At the start of each sprint, tasks were assigned to specific developers and moved into the In Progress column. As a task was completed, the assigned developer moved it into the In Review column. Then the task was discussed at the next development team meeting, generally held a few days or each week, where the task was sent back to the In Progress column with revisions or finally moved into the Complete column. The author took on a Scrum master-type role, ensuring the Trello boards were kept updated and organized, and tasks were delegated to the correct developer.

Tasks were not formatted as epics or user stories. Instead, the development team maintained a list of requirements used for making design decisions and generating tasks. These requirements were over-arching goals that the final iteration of the app was required to meet. They were thought of as a list of goals and as tools for making focused design decisions. As designs were discussed, the development team tested ideas against the requirements list before making a concrete decision.

Any game mechanic, art asset, sound effect, or other addition to the app was required to enforce at least one of the requirements, otherwise, it wasn't created or implemented. The requirements helped ensure that design decisions made throughout the project were cohesive and encouraged a positive player experience. The requirements were not static, they evolved with the project. The requirements had to be updated as the development team learned more about the game, what they thought made the game fun, and more importantly what players during playtests found fun. After each playtest, the requirements were revisited and updated to match the findings. Each iteration of the project began with an updated requirements list and ended with a playtest. For the full playtest details, data, and results refer to Brown et al. and Froome et al. [41, 63].

The first iteration of *Foodbot Factory* was mostly spent brainstorming the overall design of the game with the Bodyzone team and developing a small prototype. A dialogue-based visual novel style was chosen as nutritional messaging could easily be incorporated into the gameplay. A playtest with elementary school students from the Durham Catholic District School Board held positive results [41]. Students enjoyed playing the early version of *Foodbot Factory*, although some students found the dialogue sections tedious. The second iteration improved on *Foodbot Factory* by including mini-games to break up the long dialogue sections and an improved dialogue system using the *Yarn Spinner* plugin [31] which allowed for branching dialogue and multiple choice quizzes. The second playtest was used to test *Foodbot Factory*'s first module, based on drinks, which held positive results [41]. The third iteration further improved *Foodbot Factory* with finalized character and background art, more branching dialogue options, and an additional module based on grain foods. Again, the playtest for this iteration held positive results. Overall, students enjoyed the dialogue and mini-game format and further development iterations were spent developing more modules, with no major changes to the features and format. For the full iteration details for the development of *Foodbot Factory*, including a discussion of anecdotal observations made by the author while observing players during playtests, please refer to Appendix A.

3.3 THE FUTURE OF FOODBOT FACTORY

After multiple iterations, over a year of part-time development, and playtests in classrooms with children, *Foodbot Factory* was complete. Players found its five modules engaging and often laughed and smiled along with the dialogue. Each module was created to be easily integrated into teachers' lesson plans and playtests with students showed that *Foodbot Facotry* is an engaging app that has the potential to improve the nutrition knowledge of Canadian children [41, 63]. Ultimately, the Bodyzone team is proud of *Foodbot Factory* and feels that it contains the level of quality necessary to be released to the public and used in Ontario classrooms.

4

RECOMMENDATIONS FOR SERIOUS GAME DEVELOPERS

Chapter 3 and Appendix A discussed the development process that produced *Foodbot Factory*. For most of the developers on the team, this was their first large-scale development project that required collaboration with an interdisciplinary team that included various non-developers, including educators and nutritionists, with the goal of delivering a usable product. As such, much of the development team's culture and workflow was developed in conjunction with *Foodbot Factory*, often needing adjustment. The following chapter describes the lessons learned during the development of *Foodbot Factory* presented as recommendations for serious game developers.

4.1 KEEP IT SIMPLE

From project scope, to organizational methods, to project management, to the art style, to game mechanics, to documentation, to almost every aspect of development you can think of, the most important recommendation is to keep it simple!

Game development and computer science students are often overwhelmed with topics and ideals based on the experiences of full-time developers or other similar professionals. In class, students are shown examples of cutting-edge technology, project management processes that track every detail of development, examples of fully realized programming systems and documentation, and more. In their free time, students play games with beautifully rendered art styles, avant-garde gameplay mechanics, tight responsive controls, and more. For some developers, it's easy to feel that they need to fully integrate every concept they've learned into their own projects, often confusing what they want to accomplish with what they can and need to accomplish. For example, the early town explorer idea for *Foodbot Factory* contained multiple locations to explore, a full player avatar system with clothing rewards, an overworld map of the town, and a story that evolved over the course of many play sessions. While this idea sounded great, it wasn't realistic and didn't directly address the goal of the project, an engaging nutrition application based on Canada's food guide. In the end, the simple dialogue-based approach was achievable and successful.

This theme of overthinking occurred continuously over development and was often addressed by simplifying the approach. Many of the recommendations below can be summarized to "keep it simple", and for student projects with few developers who are likely constantly learning as they go, this is a reliable approach.

4.2 START ORGANIZED, STAY ORGANIZED

As early as a team can, they should decide on standards for staying organized during development. As mentioned, the *Foodbot Factory* development team had to re-create their Unity project between the first and second iterations, largely because these standards were not discussed early enough. Additionally, it's important to revisit this discussion at regular intervals, perhaps at each project milestone. As the project progresses, staying organized will become second nature and will ease development.

Here are a few areas to discuss:

4.2.1 *Data Repository*

Using a repository is critical for collaboration, backing-up work, and keeping a history of changes. GitHub [22] and GitLab [23] are common choices and provide good tools and support for development.

Keep the repository structure simple and clean. Early structures for the *Foodbot Factory* repository involved developers creating multiple branches for each feature they worked on, then merging their branches together upon feature completion.

The team also separated the project into two repositories, one for the Unity project, and another nested repository for art assets. Unfortunately, this led to too many branches, a convoluted repository structure, and large conflicts when merging. Also, while a second repository for art assets is a common approach for projects with large data size requirements, it was unnecessary for *Foodbot Factory* as the project's size no larger than a few gigabytes.

Eventually, the team adopted a single repository consisting of a simple linear structure and two branches: Master and Development. The Master branch was pushed to at each iteration or milestone so that a working, presentable version of the game was always accessible. The Development branch was used by developers to constantly push their local changes as they completed work. Their tasks did not have to be complete, but the repository stayed clean, and merging more often helped avoid conflicts.

4.2.2 *File and Folder Structure*

Keeping a clean folder structure and consistent file names ensures that developers can find what they need when they need it. This will shorten the time it takes developers to find new assets and should allow them to take full advantage of the search feature in whatever game engine they use. This is especially important in an asynchronous working environment, which is common for student teams. If developers constantly have to message each other (and wait for a reply) to find the files they need, then the team should immediately revisit their file structure.

A common effective strategy is to separate asset types and code, then create project-specific subsections based on deliverable type. For example, *Foodbot Factory*'s structure started with main folders for art, audio, dialogue, and code. Art was subdivided into UI, characters, foods, and background. Audio was subdivided into sound effects, music, and ambience. Dialogue was subdivided by module. Finally, code was subdivided by system type, such as dialogue, Food Drop, menus, and scene management. For searchability, some art asset file names were prefixed with labels such as "background_", "character_", "food_", to take advantage of the search functions in Unity when implementing assets.

A structure to avoid is creating folders dedicated to each module/scene of the project. For *Foodbot Factory*, the team had considered using the module names as the main folders, with subsections for asset types. For example, all the art, audio, dialogue, and code used in the drinks module would exist in a "Drinks" folder. This seems like a promising solution, as developers generally only work within one module at a time. However, in practice, finding files quickly becomes a guessing game as assets are reused in multiple modules/scenes.

4.2.3 *Coding Standards*

The key to coding standards is consistency. Enforcing code standards helps with code readability, searchability, and collaboration. Similar to the file and folder structure, programmers should be able to easily find the code they need when they need it, with the added requirement that they can easily understand it.

There are multiple coding standards available online such as Google's style guides [24]. They commonly include: (1) consistent naming conventions for classes, functions, and variables, (2) consistent spacing and tab style, (3) comment frequency and requirements, and (4) file organization.

Code cleanliness and structure is a topic well beyond the scope of this thesis. However, keeping functions as short as possible is a reliable approach, using many short functions that each perform a single. If time permits on the project, have developers review each other's code before merging. This will help the cleanliness of the code and facilitate knowledge sharing among developers, strengthening their programming and their knowledge of the codebase. This is also good experience for programmers planning to work in the industry, as participating in code reviews is generally a requirement.

4.3 MAINTAIN SIMPLE DOCUMENTATION

Documentation adds another layer to development that could easily be ignored and often is. Explaining code and systems can be challenging for any developer, especially for students learning to code. In class, students are shown UML diagrams, flowcharts, state diagrams, sequence diagrams, and many other visualizations of code structures. When using tools, students see detailed documentation with pages upon pages of detail and plenty of code samples. Similar to programming, technical writing is a skill that requires years to develop. Companies often hire developers specifically as technical writers to create and maintain documentation. Fully realized documentation takes a lot of work, definitely too much work for a student project. But student projects don't need fully realized documentation.

So, keep it simple.

- Point form documentation is valid and easy to create.
- Use a shared document online for project organization details.
- Code-specific documentation can be shared online or directly in the code as comments.
  - Provide a short description of each class.
  - Mention the key functions of each class and what they do.
  - Add a name so other team members know who to speak to if they have questions.

Simple documentation can be created quickly and can ease collaboration, helping share code between team members and helping teach code to new members who join the project. It can also help a developer recall code that they wrote months prior, a surprisingly useful benefit. All developers should become accustomed to taking notes as they develop and keeping the notes clear and simple.

4.4 YOU NEED A PROJECT MANAGER

Without some degree of task tracking and prioritization, team-based development can quickly become disorganized. Some developers may end up wasting time on the same tasks and others may create assets or code that doesn't fit with what the others are creating. Estimating timelines and successfully completing work before deadlines will also become much more challenging than necessary.

For the early stages of the development of *Foodbot Factory*, ignoring project management was perhaps the biggest mistake the development team made. Progress was slow, unfocused, and developers seldom communicated their goals and progress. Once the team adopted agile-based project management strategies, development became much less exhausting.

The key to project management is having tasks and goals tracked and prioritized in some shareable capacity. Tools such as Trello [18] and Jira [16] are good choices. Teams should decide on some simple standards for the tool they choose, and keep the tool updated as they work. Having short weekly meetings to discuss progress and update the tool can also help keep your team organized. The project manager role does not need to take up all of a single developer's time nor does it need to be given to a single developer. Project management responsibilities can be shared among the team.

4.5 TEST EARLY, TEST OFTEN

Developers should create a prototype that players can interact with, as early as possible. These prototypes should not be widely released to players but can still earn actionable feedback. Use Unity to create a simple interface or a basic mini-game. Use cue cards or paper to create a fake interface for players to discuss. Anything that developers can create in a day or two that can facilitate interaction and discussion with real players.

For *Foodbot Factory*, the earliest playtest was extremely helpful for learning about how students interact with tablets in a chaotic classroom setting, showing the development team that modules needed to be short and easy to use, without external instruction. The team also learned that the novelty of using tablets in class was enough to keep most students engaged in learning, even if the game was reading-focused.

Continue these playtests throughout development. Receiving constant feedback from players is vital for the health of the project. Usability issues can be easily spotted when players interact with the game and player feedback can be used to adopt more successful designs. For student projects, these playtests are also a good time to spot bugs they may have missed during development.

Influential studies have revealed that the majority of usability problems are detected with the first three to five participants in a usability study. More specifically, four or five participants allow developers to discover 80% of a product's usability issues [107]. So developers don't need to spend a lot of time testing, just a few sessions are enough to be confident that their project stays on track.

Finally, if finding players in the target demographic is a challenge, developers should simply ask friends or family to play. Just remember their feedback may be biased, as they may avoid providing negative feedback since they know the developers personally.

4.6 take advantage of plugins and premade assets

Game development students are often encouraged to create all of the code and art for their games themselves based on class-specific requirements. Art and animation classes will require students to implement various custom assets. Programming classes will require students to implement game-specific programming patterns to support game mechanics. Modeling and graphics classes will require students to implement 3D models with fancy shading and lighting techniques. With all of these requirements, it often becomes easy for student developers to forget that there are plenty of royalty-free pre-made assets available online that they should absolutely use in their projects.

Developing games takes a significant amount of time and effort. Much of that effort often goes into developing art assets and variations of those assets needed to create a polished game. Using pre-made assets can cut development time significantly, especially for teams who may only have one or two artists. Have artists spend their time working on game-specific assets, such as original characters and their personalized weapons. If your project allows, use pre-made assets for generic content, such as trees, buildings, weapons, walking animations, and anything else your game might require. A small budget for assets can help your team find assets that really fit your game's needs, but many assets can be found online for free as well. Just be sure to read the copyright agreements that accompany the assets and ensure they fit with the goals of your project.

Plugins are premade tools or code that provide specific features to a development environment. For example, the Yarn Spinner plugin provided dialogue system tools for Unity that were used to create the dialogue in *Foodbot Factory*. Plugins are an excellent way to integrate specific mechanics into a game that may have been too time-consuming to create from scratch. Using plugins is also great practice for students, as learning to understand and use code written by other developers is an important skill. Students using plugins should also spend some time thinking about what they liked and didn't like about the code, and what they might have done differently had they written it themselves.

For reference here is a list of all the pre-made assets and plugins used in *Foodbot Factory*:

- Most background art from Vector Toons [30].
- Various food items from Vector Toons [30].
- Menu assets (buttons, icons, etc.) from the Unity Asset Store [19].
- Various sound effects from the Unity Asset Store [19].

- Menu and grocery store music from Storyblocks [27].
- Ambience sounds for deli, pier, kitchen, and lab from Storyblocks [27].
- Yarn Spinner plugin for the dialogue system [31].
- RTVoice plugin for dialogue voice actors [17].
- DoozyUI plugin for developing the main menu interface [20].

4.7 REASONS TO USE UNITY

The first large decision a development team must make is to determine what game engine they will use to develop their game. Unity is an excellent option for student developers.

Some benefits of using Unity are as follows:

- Free to use.
- Detailed documentation [28].
- Supports tools for many common game mechanics.
- Simple interface and controls.
- Common scripting language (C).
- Integrated marketplace to download plugins and assets [19].
- Build support for most gaming platforms (PC, mobile, consoles, etc.).
- Supports 2D and 3D games.
- Lots of community-made tutorials online.

There are other game engine options available to students, most notably Unreal Engine [29]. Unreal supports many of the same features as Unity, however, its blueprint scripts use a node-based visual programming style that may be too advanced for some student developers. Unreal was also created specifically for 3D games, so Unity has better support for 2D.

Unity is a reliable option, likely to have all the features student developers require but ultimately developers should choose the engine that contains the features they need and has a workflow they find motivating to use.

4.8 EMBRACE CHALLENGES

Development is challenging. Every detail of a game needs to be carefully thought out and implemented. The vast knowledge required to fully implement a game can be overwhelming, especially when game technology is constantly advancing into new territory. Students and professional developers alike must be able to continuously implement new mechanics, use new tools, and design in new styles. Developers transition between projects, companies transition to new software, and designs transition to meet unexpected player expectations. The learning never ends.

This fluidity, while challenging, is part of the reason why many developers are so passionate about game development. Challenges are the source of new and exciting game experiences that are motivating to develop and play. Cool gameplay mechanics, beautiful graphics, brilliant stories, interesting characters, captivating music, clever puzzles, and much more, are all born out of new challenges in game development. Developers that learn to embrace these challenges and adopt a mindset for continual growth will find success.

When faced with challenges that seem insurmountable, which will happen, developers should take a step back, reduce the scope, and simplify the approach. Then, iterate on the solution until it meets the needs of the project. Changing their mindset from "I won't be able to figure this out" to "these are the steps I can take to figure this out".

5 ACCESSIBILITY UPDATES

5.1 OVERVIEW

As described in Chapter 3 and Appendix A, *Foodbot Factory* was developed following an iterative approach, with each iteration informed by the results of user tests conducted in elementary school classrooms. Following the initial evaluations of *Foodbot Factory* [41], a second study found that *Foodbot Factory* significantly increased nutrition knowledge in children compared to a control app [63]. However, *Foodbot Factory* was not developed with full accessibility in mind and thus students who experience low vision and blindness may not be able to benefit from the nutrition education it provides. While the app does include voice-overs for most in-game dialogue, some aspects of the game are presented only visually. This limitation may cause unnecessary challenges for students with low vision or blindness. Furthermore, users interact with *Foodbot Factory* using a touch screen or mouse which is a barrier for students with varying degrees of low vision. *Foodbot Factory* also lacks remappable keyboard controls that can assist people with limited motor control [1].

5.2 ACCESSIBILITY ANALYSIS OF *FOODBOT FACTORY*

As a starting point for updating *Foodbot Factory* for full accessibility, both the WCAG and the GAG were used to drive proposed updates. However, the Speech category from the GAG was not considered as it is not applicable to *Foodbot Factory*, as players are not required to speak while playing. Furthermore, many suggestions in the GAG focus on 3D and multiplayer games which are also not applicable to *Foodbot Factory*. Each proposed update was chosen based on its feasibility to be implemented, and its relevancy to *Foodbot Factory*. The following is a list of accessibility features that *Foodbot Factory* should integrate, organized based on the type of disability, similar to the structure of the GAG [1].

5.2.0.1 *Proposed Motor Updates*

- Allow players to use the keyboard to play.
- Allow players to customize their controls.
- Support portrait and landscape modes on mobile devices.
- Re-sizable interfaces.

The Motor updates are intended to assist users with mobility limitations. Currently, *Foodbot Factory* can only be played in landscaped mode with either the touch screen or a mouse.

5.2.0.2 *Proposed Cognitive Updates*

- Include interactive tutorials.
- Mini-game difficulty adjustment.
- No important information conveyed with text alone.
- Highlight important words.
- Voice-overs for all text.
- Re-playable dialogue.

The Cognitive updates are intended to assist users with cognitive disabilities. Many of the suggestions from the GAG in this category have been implemented since *Foodbot Factory* was designed for young students of varying ages and reading skill levels. However, only the main dialogue in *Foodbot Factory* has the voice-over option, and menu text and some other user interface text do not. The tutorials for each mini-game are also only presented in text. Adjustable game speed was considered for this list, however during most sections of the game users are able to progress at their own pace, therefore game speed is not relevant.

5.2.0.3 *Proposed Vision Updates*

- No information is conveyed with visuals alone.
- No information is conveyed with colour alone.
- Full text-to-speech voice-over support.
- Adjustable contrast.
- Adjustable font size.

The Vision updates are intended to assist users who have low vision or are blind. These updates also increase compatibility with different screen sizes. A universally appropriate font size does not exist due to differences in blindness, screen sizes, and distances from the screens. Therefore, allowing the player to select and adjust the text size and visuals based on their preferences and environment is ideal. *Foodbot Factory* also contains multiple-choice quizzes that display small images of foods. Some of the images, such as the icons for whole grain bread and refined grain bread, use the same base image, only varying in colour, as seen in Figure 6. The subtle difference may be problematic for some users.

5.2.0.4 *Proposed Hearing Updates*

- Provide separate volume controls for different types of sounds.
- Stereo/mono toggle.
- Volume sliders for better customization.

The Hearing updates are intended to assist users with hearing disabilities. *Foodbot Factory* does not rely on audio cues for important information, other

Figure 6: The bread quiz uses bread images that only vary in colour.

than voice-overs. Currently, background music, sound effects, and voice-overs are controllable with a toggle on/off option, only allowing them to mute each type of audio. The audio settings menu should be updated to use volume sliders, to allow players to customize the volumes of each type of sound (background music, sound effects, and voice-overs).

5.2.0.5 *Proposed General Updates*

- Provide accessibility details in-game and externally.
- Savable settings.
- Inform updates through user studies.

The General updates detail general suggestions that support the overall accessibility of the game. Currently, *Foodbot Factory* completely resets when the game is closed, and settings are not saved. Other suggestions for this category include external information that may help players access and use the game. The GAG also suggests that user studies should include users with disabilities from each of the specified categories.

5.3 FOODBOT FACTORY AS AN AUDIO GAME

Updating *Foodbot Factory* into an audio game will provide a great starting point for updating its accessibility. Since audio games are designed for people with low vision and blindness and are playable without any visuals, they tend to meet many of the GAG accessibility requirements by default, especially from the vision category. *Foodbot Factory*, being a visual novel game, also lends itself well to being updated into an audio game and already supports voiceovers for most dialogue.

An iterative, vertical slice approach was used to implement and test audio game-inspired features into the main menu and one module of *Foodbot Factory*. Two iterations were used to focus on specific features of the update. First, a keyboard control update, and second an audio update. Both iterations are explained in detail below. The goal of these iterations was to create a testable vertical slice of *Foodbot Factory* that allows players to boot the game, select a module, and play it to completion by interacting through the keyboard and audio alone, with no visuals or screen required (though still available). The protein foods module was chosen for this update, as it takes the longest time to complete and contains a variety of interaction types, making it a good candidate for a user study.

5.3.1 *Iteration 1: Keyboard Controls*

Keyboard controls are the primary means of interaction for most audio games. People with low vision and blindness often control computers using a keyboard paired with screen reader software, such as JAWS [25] or NonVisual Desktop Access (NVDA) [26]. This software provides users with a means of navigating interfaces with spoken text and descriptions of onscreen items (text, titles, icons, image descriptions, etc.). Thus, adding keyboard controls to *Foodbot Factory* will mimic this screen reader interaction and will help support further audio game updates. According to the GAG, keyboard controls can also help support users who simply have trouble using touch screens or a mouse for various reasons.

Foodbot Factory was designed to be played on a touch screen or with a mouse/cursor, thus keyboard controls had to be added retroactively to support current interface designs. For the vertical slice, the following interfaces needed to be updated with keyboard controls:

- Advancing dialogue.
- Sorting mini-game.
- Title screen "Go!" button.
- Main menu module selection button.
- Protein foods module selection buttons.
- "Where should Superbot travel?" buttons.
- Quiz multiple choice buttons.
- Final continue button.

5.3.1.1 *Design and Implementation*

The interfaces listed above can be divided into two categories: (1) gameplay, and (2) user interface buttons.

First, the gameplay category includes the dialogue and sorting mini-game, which both require simple custom solutions. For dialogue, all keystrokes from the keyboard are detected and advance the dialogue, using the same code that advances the dialogue when the user clicks or taps the screen. In the sorting mini-game, foods drop from the top of the screen onto moving conveyor belts. Users must click or tap the conveyor belts to change their direction, indicated by left or right arrows, to sort food into matching buckets. Keyboard controls were added here to allow users to use the left and right arrows on the keyboard to change the direction of all conveyor belts, again hooking into the same code used to change direction on a click or tap.

Second, users should be able to navigate through a set of user interface buttons using the arrow keys, then press the space bar to use the button, causing the same behaviour as if the on-screen button was clicked or tapped. All button items from the list above can use this approach, which can be accomplished by taking advantage of Unity's hierarchy and component scripts. The hierarchy allows developers to specify groups of objects (such as menu buttons), by placing them in a list as children of a single parent object. This pattern can be repeated, so children can be parents with their own list of child objects, creating a sort of hierarchy-family tree. In fact, all objects in a Unity scene exist somewhere in the hierarchy as either parents or children of other objects. Component scripts allow developers to attach C# code to objects to give them custom behaviour, such as detecting the arrow key inputs to change the direction of conveyor belts in the sorting mini-game.

Selection and Selector components were created to organize buttons into a list that users can navigate to make a selection. Every button object receives a Selection component which allows it to be selected and pressed. An object with a Selector component holds a list of all its children that have Selection components and keeps track of which button is currently selected. This list is gathered recursively through children of the Selector component, so any Selection components on children of children are gathered as well.

In practice, this system allows a developer to add Selection and Selector components to objects that already exist within the Foodbot Factory Unity scenes. For example, on the module select screen, each of the module buttons exists as grandchildren of the Plate object (which represents the dinner plate in the scene), as seen in Figure 7. Each button receives a Selection component, and the Plate object receives the Selector component. When the scene becomes active the Plate gathers all selection components from the objects below it in the hierarchy, in this case, the buttons, then sets the first button in the list as active. The active button, in this case, the Drinks button, begins to animate so the player knows it is currently selected. Now the system waits for the player's input. They can either press an arrow key to change their selection to another button in the list or press the space bar to use the button as if they had clicked or tapped it.

Figure 7: The module selection screen in *Foodbot Factory*

After updating the dialogue system and the sorting mini-game to support keyboard controls, and adding Selector and Selection components to all buttons listed above, players are capable of selecting and playing the Protein Foods module with the keyboard, completing the iteration.

5.3.2 *Iteration 2: Audio Update*

As the GAG specifies, "Ensure no essential information is conveyed by a colour alone" [1]. Currently, many aspects of *Foodbot Factory* are presented only visually. For example, quiz options during dialogue sections are displayed to the player with only food icons, often varying only by colour.

For guidance on how to implement this heuristic, audio games and their research can be used for inspiration. Audio games must be completely playable by audio, therefore their audio implementations can be used to bolster the aspects of *Foodbot*

Factory that do not convey information through audio. We can thus upgrade the GAG heuristic above to "all information required to play *Foodbot Factory* efficiently should be presented by both audio and visuals".

Three aspects of *Foodbot Factory* and their audio upgrades are discussed below.

5.3.2.1 *Audio Menus*

Voiceovers had to be extended to support reading all important text, buttons, and icons using audio menus. Audio menus, found in most audio games, are interfaces that allow the user to make a selection by cycling through a list of spoken options. Automated phone menus, commonly encountered when using a voicemail system, are a type of audio menu, though it's generally recommended to allow users to cycle through options of a list themselves using the keyboard rather than forcing them to listen to all options before making a selection [65].

Leveraging the Selector and Selection system used for keyboard controls, audio menus can be added to button screens, using RT Voice. A text field is added to each Selection component where a developer can set text for RT Voice to read when the Selection is active. Additionally, when a Selector becomes active and gathers all the Selections RT Voice reads out a title or instruction for the menu, specified by a developer.

For example, once the module select screen becomes active (as seen in figure 7) and the Selector has gathered all the Selections attached to the module buttons, RT Voice will read out "Please select a module using the arrow keys and make a selection with the space bar". Then as the user presses the left or right arrows to cycle through the selections, RT Voice will read out the text on each button as "Drinks", "Protein Foods", "Grain Foods", "Veggies and Fruits". Finally, the player can press the space bar to activate the button once they are happy with their selection.

According to Garcia and Neris [65], users found it difficult to track how many items were in a spoken list and did not know when the list cycled back to the beginning. They recommend using numbered items in a spoken list, so users have an easier time navigating to their desired item. Using this idea, a number system was added to the Selector component. To implement this idea, Selector components assign a number to each of its gathered Selections and will read out the number before reading the Selection's text. Thus, the buttons are now spoken as, "1. Drinks", "2. Protein Foods", "3. Grain Foods", and "4. Veggies and Fruits".

5.3.2.2 *Audio Mini-Game*

The current sorting mini-game design cannot be played without visuals. Specifically, three aspects of the game require visuals to play: (1) object icons on falling items must be paired with the matching icons on buckets, (2) players must watch the positions of items on the screen, (3) conveyor belt directions are shown with arrow icons on conveyor belts. Two approaches were considered for updating this mini-game into an audio game.

First, many audio games that contain real-time moving objects use spatial audio. For example, games such as *A Hero's Call* [7] which can be found on AudioGames.net [13]. For the sorting mini-game, spatial audio would allow players to listen to the position of the falling items. However, it would require the falling objects in the game to constantly produce a clear sound. Text describing the item's icon could be spoken on a loop, though this would sound unnatural, and could become too overwhelming or annoying when more than two items are falling, which is not a recommended approach [103]. Spatial audio also requires the player to wear headphones and have some experience using spatial sound cues, which may not be suitable for students in an elementary school classroom [103, 35]. Thus, this approach was not used.

Second, creating a re-imagining of the sorting mini-game as an audio-sorting game. Keeping with the sorting theme is important, as the dialogue before the sorting mini-games involves asking Superbot to help deli or dock workers to help sort a new shipment of food. Rather than having to spatially sort objects based on their position, players are asked to press the left arrow if a played sound could be found in a kitchen and press the right arrow if not. Four sounds, chosen randomly from a list, are presented to the player in sequence and the player must decide if each sound could generally be found in a kitchen. The sounds were selected carefully to ensure they clearly convey their source and context [103, 104]. When the player chooses correctly, a cheerful ring sound is made, and when the player chooses incorrectly a dissonant buzz sound is made. Sounds used for sorting in the game include:

- Chopping an apple.
- Chopping an onion.
- Grill sizzle.
- Starting a microwave.
- Opening a can of pop.
- Grinding pepper.
- Pouring water an ice.
- Utensils in a drawer.
- Cheering from a crowd.
- A crowd saying "aw".
- Chicken clucking.
- Drums.
- A heartbeat.
- Slide whistle.
- Xylophone.
- A voice saying "yehaw!".

This sorting audio approach was implemented to replace the standard sorting mini-game. While not necessary for playing the game, visuals were also added to accompany the audio for players that do play with visuals. These visuals include text to accompany any spoken text, check mark and "OOPS" icons to accompany correct and incorrect answer sounds, a large question mark icon to be displayed while a sound is being played, and miscellaneous icons to accompany the sorted sounds once their source is revealed (such as an apple icon for the "Chopping an apple" sound).

5.3.2.3 *Dialogue Updates*

As mentioned, *Foodbot Factory* contains voiceovers for all dialogue. However, these voiceovers were added to support young readers and were not intended for full audio support. Some important information found in the dialogue sections is only presented through visuals and needs to be updated with audio.

First, contextual information, such as the setting and who is speaking, are not always shared through dialogue and voiceovers. Background images show the setting, and character portraits accompanied by directed speech bubbles show who is speaking. Audio information could be added by including a narrator voice that describes the setting and who is speaking. However, characters often pop in and out during the dialogue, and RT Voice is limited to only two types of voices, so this could add more confusion for players and too much complexity to the implementation. The simplest solution is to ensure the dialogue is diegetically descriptive. As in, when characters enter a new space and speak with a new person, they can describe the context in their speech. For example, when Superbot enters the deli, the deli worker can say "Hi Superbot, welcome to my deli!", and Superbot can respond, "Thank you. Can you help me learn about your protein foods?". This approach requires minor updates to dialogue at all transitions and keeps the flow of dialogue focused on the game and story.

Next, RT Voice presents another challenge, it only allows developers to use two voice types, therefore characters speaking to each other can often have the same voice, making dialogue confusing without visuals. Ideally, custom voiceovers would be recorded and played with dialogue, so players could decipher each character by their voice. However, custom voiceovers are expensive and RT Voice is already tightly implemented into the dialogue system, so it must be utilized. RT Voice does provide some control for voiceovers, including a choice between low and high voice options (described in the RT Voice system as male or female), and pitch control. These options can be set per character, and combined with careful planning of which characters speak to each other during dialogue sections, can provide enough variability in the voices to make conversations understandable.

Finally, players must tap, click, or press a key to advance dialogue. With visuals, players can clearly see when they have finished reading a line of dialogue and will tap as needed. However, with spoken dialogue, it's not clear when a line has been completed, as spoken dialogue often contains natural breaks that could be confused for a completed line. To address this issue, the spoken dialogue could advance automatically, so players would not need to tap at all. However, tapping to advance dialogue is an intentional choice, as it encourages players to stay engaged through interaction and allows players to advance at their own pace so they have time to think about what they're reading. A small click sound can be played to provide feedback to the player to signal a completed line of dialogue [65]. In Foodbot Factory, a small click sound already plays when dialogue is advanced; this is a common feature of video games. A lower variation of this clicking sound can be added to play at the end of each dialogue line to signal the end of a line so that the player can advance the dialogue when desired.

After updating the dialogue system and sorting the mini-game to support keyboard controls, and adding Selector and Selection components to all buttons listed above, players are capable of selecting and playing the Protein Foods module with the keyboard, completing the iteration.

5.4 SUMMARY

Now that *Foodbot Factory* contains keyboard controls, audio menus, a new audio mini-game, and an audio-friendly dialogue system, all information required to play the Protein Foods module is presented by both audio and visuals. In fact, players are now capable of playing the vertical slice through audio alone, effectively turning the vertical slice of *Foodbot Factory* into a testable audio game. The next chapter describes a usability study for testing this *Foodbot Factory* audio game with players.

6 USABILITY STUDY

6.1 OVERVIEW

This chapter discusses the usability study that was conducted on the audio game vertical slice of *Foodbot Factory*. The purpose of the study was to examine the usability of the vertical slice and gather player feedback to help inform the full implementation of accessibility and audio game features into all modules of *Foodbot Factory*. Specifically the usability of the audio interfaces and the understandability of the protein foods module content, of users who played without visuals.

This study was held in December 2021 amidst Covid-19 lockdowns and was therefore held entirely online.

6.2 STUDY DESIGN

The usability study had participants play through the Protein Foods module of *Foodbot Factory* while visuals were hidden, forcing them to play using the keyboard and audio as a means of interaction. A semi-structured observational approach was used throughout the play session. Participants were observed playing the module and were encouraged to stop to provide feedback while they played and were asked questions at points of interest. Once the play session was complete, participants completed a System Usability Scale (SUS) questionnaire [92], then time was provided for an open-ended discussion reagrding the play session. Participants played *Foodbot Factory* online by connecting to the author's desktop computer from their own computer using Parsec and spoke to the author through Google Meets.

The SUS questionnaire was specifically chosen as it provides a quick and accurate method for accessing usability, which is closely related to accessibility [71] and has been used in other published serious game work [110, 94]. Other questionnaires including the NASA task load index [69] and the Game Engagement Questionnaire [40] were not chosen at this stage of testing since the purpose here was to examine usability specifically and including other questionnaires would increase the overall study time. Playing the *Foodbot Facotry* audio game can take about 30 minutes due to the speed of the run-time generated voiceovers, thus the SUS provided a quick assessment of usability without taxing participants for a long period of time [86] and still allowed time for discussion.

Ideally, a comparative study that compared the original and audio game version of *Foodbot Factory* would be conducted to better gauge the accesisble version. However, given the lockdowns and work-from-home orders arising from the Covid-19 pandemic, it was anticipated that recruiting participants would be difficult. The author was aiming for five participants only, thus wanted to avoid any delays and focus on collecting data about improving the *Foodbot Factory* audio game.

6.3 PARTICIPANTS

A total of 20 adult participants participated in the usability study. Participants were known either personally or professionally by the author and were specifically recruited from three groups. Nine participants were game developers and nine other participants were known to play many video games. Both of these groups were chosen as they can provide thoughtful feedback based on their experience with video games. The other two participants were elementary school teachers. Since *Foodbot Factory* is targeted at elementary school students, teachers were recruited to provide thoughtful feedback based on their experience in the classroom and how they might use *Foodbot Factory* with their students.

In addition to the participants described above, there are two demographics of players that would have ideally been recruited for this study. First, as with the other *Foodbot Factory* studies [41, 63], children in grades 4-6 as they are *Foodbot Factory*'s target demographic. Second, participants with disabilities, especially with low vision or blindness, as the accessibility and audio game features of *Foodbot Factory* were being examined. Unfortunately, recruiting from these groups was not feasible at the time of the study, as it was held online amidst Covid-19 lockdowns in December 2021.

Although this study did not include participants who experience low vision or blindness, data collected by sighted participants is still helpful for accessing the usability and accessibility of the *Foodbot Factory* audio game. In "Celebrating 20 years of Computer-based Audio Gaming", Urbanek and Guldenpfennig suggest that while sighted participants cannot experience the nuances that blind participants experience, blindfolding sighted participants while they play can bring them one step closer to a genuine audio-only experience [104] and other audio game work has used this blindfolding approach [96, 102]. To replicate this design during an online study the author removed all visuals from the *Foodbot Factory* audio game, leaving only a black screen. Therefore, participants needed to rely on only audio to play.

Participants who participated in the study did so voluntarily and were not compensated for their participation. The experiment was approved by the Research Ethics Board of Ontario Tech University with reference number 14879.

6.4 experimental procedure

The usability study was held online and participants were asked to sit at their computer with headphones on, in a quiet space of their home if possible. At the time of recruiting, participants were asked to create a Parsec account and download the Parsec Windows application in the preparation of a scheduled meeting time. At the scheduled time, participants were asked to join a Google Meets voice chat, where they were greeted and provided with a consent form to complete and sign, found in Appendix A. Participants were then provided with a link to a Parsec session, which gave them remote control of the author's computer. Once connected, the *Foodbot Factory* audio game was described. The game window in Unity was minimized so participants could not see any visuals, they could only hear the game's audio and were instructed to use the keyboard to control the game. Next, participants were instructed to start the game through Unity and to navigate to the Protein Foods module once they reached the module selection menu. No information about how to play or control the game was provided at this stage. Participants then played through the module while the author observed them, and were free to pause at any point to provide feedback or ask questions. If the player was struggling to understand or control the game, the author took note and provided some direction. Once participants completed the module, they were provided with a link to a Google Form that included the SUS questionnaire and an optional section for written feedback. When participants completed the SUS questionnaire, the author facilitated an open discussion with them, asking questions about the play session and giving participants another chance to provide feedback.

6.5 methods

Aside from the SUS questionnaire that participants completed, and the open discussion, the author took semi-structured point-form notes while watching and listening to participants play and took notes for all feedback provided by participants throughout the study. These notes were focused on usability issues observed during the playtests and feedback offered by participants when they experienced an issue. After the play session participants were sent a link to the Google Form questionnaire. The SUS was designed to provide a subjective measure of a system's usability. In other words, the SUS provides opinions on how easy to learn and use a system is. The SUS contains the following 10 statements, each scored by participants on a 5-point Likert scale from *Strongly Disagree* to *Strongly Agree*.

1. I think I would like to play this game frequently.

2. I found the game unnecessarily complex.
3. I thought the game was easy to use.
4. I think that I would need the support of a technical person to be able to play this game.
5. I found the various functions in this game were well integrated.
6. I thought there was too much inconsistency in this game.
7. I would imagine that most people would learn to play this game very quickly.
8. I found the game very cumbersome to use.
9. I felt very confident playing the game.
10. I needed to learn a lot of things before I could get going with this game.

A final score between 1-100 for the SUS is obtained as follows: 1 is subtracted from the result of each odd-numbered question, and for the even-numbered questions, the result is subtracted from 5. The individual scores are summed and then multiplied by 2.5. A score below 68 is considered below average and a higher score shows better overall usability [92]. The SUS questionnaire has been discussed, verified, and accepted as valid measures for usability testing for a variety of technology-based applications [92].

6.6 RESULTS

6.6.1 *System Usability Scale*

Responses to the SUS have been organized into a diverging stacked bar chart, as seen in Figure 8. Favourable responses for an SUS alternate each question between positive and negative responses. In other words, responding "Strongly Agree" to odd-numbered questions indicates better usability, and responding

"Strongly Disagree" to even-numbered questions also indicates better usability. Thus for SUS responses, the ideal state of a diverging stacked bar chart would show bars alternating right then left repeatedly descending down the chart. As seen in Figure 8, other than Question 1, the responses are generally favourable and alternate as expected. For all 20 participants, the mean score was 81, which indicates above-average usability.

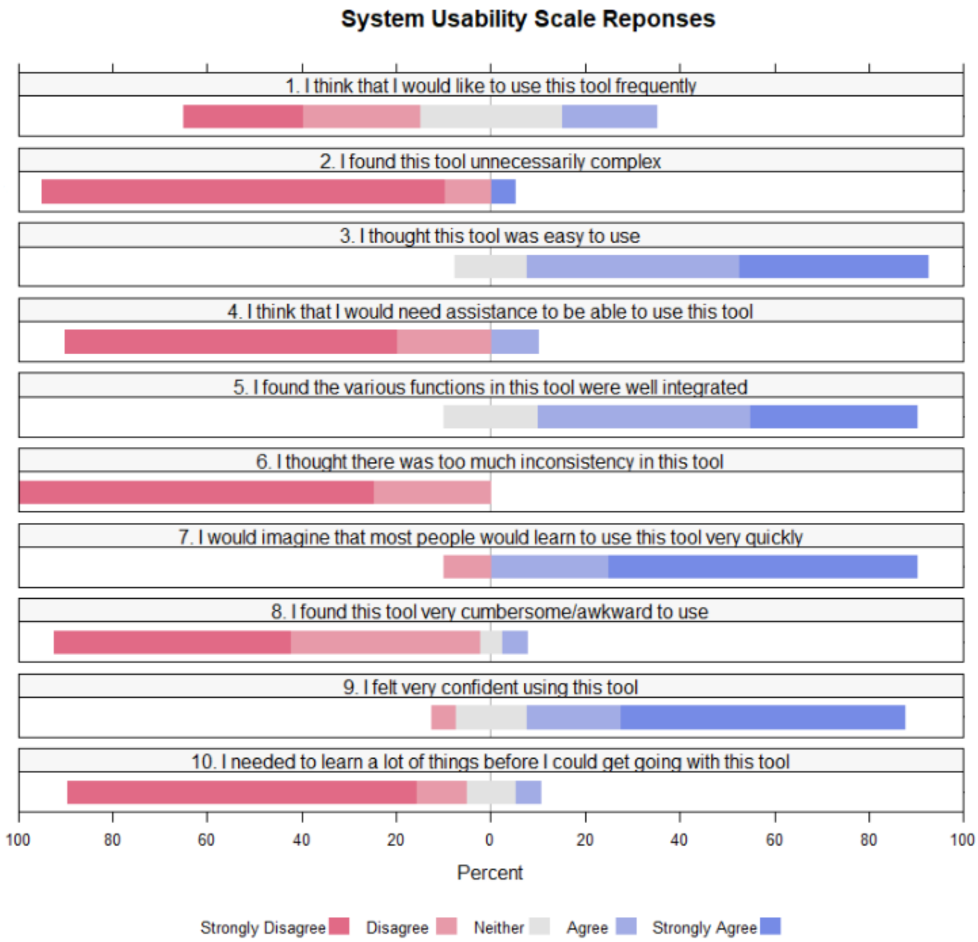

Figure 8: Results for the System Usability Scale.

The mixed responses to Question 1, "I think I would like to use this tool frequently", can be explained by the age difference between the intended audience of *Foodbot Factory* and the participants of this study. *Foodbot Factory* was designed for children in grades 4-6, therefore it is not unexpected that most of the adult participants did not feel that would want to use the tool frequently. Even though the question was not applicable, it was still included as the final SUS score calculation relies on having ten responses. All participants, including those who asked for clarification about the question, were told to answer honestly. This age difference may also explain the above-average mean SUS score. Adult participants, especially those with video game experience, may have an easier time learning to use the *Foodbot Factory* audio game than children.

6.6.2 *Feedback and Observations*

Overall, participants enjoyed their time with the *Foodbot Factory* audio game. Some participants required extra guidance, though all participants were able to successfully complete the play session in the allotted time. Once participants were aware of the controls and the audio cues for the audio menus and dialogue sections, they were able to smoothly and consistently progress to the end of the module, barring the occasional confusion from unclear dialogue, as will be discussed below. Participants laughed along with the characters and jokes in the dialogue and most were able to complete all quizzes without mistakes, showing they were able to follow along and even enjoy the audio-based dialogue. Participants also directly expressed that they enjoyed the game and felt the audio-based mechanics were unique and fun.

Although the overall sentiment was positive, several usability issues were observed during the sessions, many reoccurring for multiple participants. Participants commonly vocalized thoughts about usability issues they were facing, communicating their confusion and providing feedback for solutions.

The semi-structured notes recorded by the author during the play sessions focused on usability issues observed during the playtests and feedback offered by participants, have been organized into Figures 9 and 10. The left column in Figure 9 lists all observed usability issues, and the right column indicates the number of participants who encountered the corresponding issue. Likewise, the left column in Figure 10 lists all feedback offered by participants, and the right column indicates the number of participants who offered that feedback. Both Figures have been organized descending from the most to least frequent by participant count.

Obersvations	Count
The player did not advance the "Go!" dialogue line, before the mini-game.	9
The player needed help understanding how to use the audio menus.	7
The player advanced dialogue lines by accident.	7
The player needed help understanding how to control dialogue sections.	5
The player picked the wrong answer in the final quiz.	5
The player moved between audio menu options before they were read aloud.	3
The player accidentally selected a quiz response, just as the dialogue transitioned into the quiz.	3
The player forgot the audio game controls.	2
The player accidentally selected a location on the "Where should Superbot travel?" screen.	1

Figure 9: Themes among observations during play-sessions.

The following subsections discuss the common themes among observations and feedback presented in Figures 9 and 10.

6.6.2.1 *Tutorials and Help*

The audio menus and dialogue sections do not include a tutorial, and as a result, many players were confused early in their play sessions. When players encounter the first audio menu, on the module selection screen, they are briefly prompted with "Please select a module," then after a brief pause the first item in the audio menu is spoken as "1. Drinks". The majority of participants naturally tried using the arrow keys to move through the options, then selected the protein foods module. However, seven of the participants either waited for more audio or pressed incorrect keys, then needed more instruction from the author. Similarly, during the first dialogue section most participants naturally began pressing keys to

Feedback	Count
Should be able to move back during dialogue sections and/or rehear the current line.	10
Some words and phrases are hard to understand.	10
Some dialogue lines are too short and make the game feel slow.	8
Use a more obvious sound at the end of the dialogue lines.	8
There should be a tutorial for the dialogue sections.	7
The transistion between the Docks and Deli is unclear.	5
Audio menus need a tutorial.	4
Have a narrator describe the scene and transitions.	4
Use real voice actors.	4
Lab background is too loud and distracting.	3
Allow using the number keys to select audio menu options.	3
Say how many options there are at the start of the audio menu.	3
Option to play dialogue automatically, instead of pressing a button to advance.	2
Include a help menu.	2
Use the right arrow key to advance dialogue and the back arrow key to move back.	2
Reduce the amount of required button presses.	1

Figure 10: Themes among feedback from participants.

correctly advance dialogue after each line was spoken. However, five participants waited for additional audio and did not press any buttons until prompted by the author. Additionally, almost half of the participants did not initially realize that the clicking sound at the end of each spoken line indicated that the line was finished. All participants who initially had an issue were able to easily understand and use both the audio menus and dialogue sections after a brief tutorial from the author.

The audio mini-game section does include a brief dialogue tutorial and none of the participants had trouble understanding how to play it, indicating that brief dialogue tutorials may solve the need for external intervention. However, two participants did ask the author to re-explain the controls of the audio mini-game. Brief dialogue tutorials could be included when each section is encountered for the first time, and a help button could re-speak the tutorial for the current section when players need a refresher.

6.6.2.2 *Dialogue Sections*

The dialogue sections presented a few usability issues for players. Most notably, participants often felt that dialogue lines were too short, resulting in slow and unnaturally segmented dialogue. The RT Voice text-to-speech plugin used for the dialogue adds a slight delay in processing each line, which likely exemplified this unnatural feeling. For example, short lines such as “Yes!” add a much larger delay to dialogue when in an audio-only form, as RT Voice needs to process and play the line then the player must wait for the end-of-line queue sound so they know they can move to the next line. There are cases in *Foodbot Factory* where the participants encounter groups of these short dialogue lines, such as right before the audio mini-games with “Ready?”, “Go!”.

This “Go!” line also presented a problem with participants. Nine participants paused for a few seconds on this dialogue line as they believed it signaled that the mini-game had started. This was never an issue with visuals as the “Go!” dialogue is clearly presented in a dialogue bubble which shows players that it needs to be advanced as with all other dialogue in that format. However with audio only, players are not provided the same visual signal and it's fair for them to assume the game has started when they hear "Go!", then wait for another cue so they don't press a button at the wrong moment and accidentally lose the mini-game.

These issues can be easily resolved by editing the *Foodbot Factory* script to better suit the audio format. Short lines should be avoided and can be combined with an adjacent dialogue line where possible. For example, the "Yes!" dialogue line, could become "Yes, unsaturated fat is a healthy fat!". The "Go!" issue could be resolved by simply updating the dialogue line from "Go!" to "Press the space bar to begin!", providing players with direct guidance.

Another issue seven participants encountered was occasionally skipping dialogue lines by mistake. Either players thought the line had completed and advanced too soon or players accidentally pressed twice by mistake when advancing dialogue at the correct time. A slight delay could be added to the input detection to avoid the double button press, though this does not help if the player accidentally advances too soon. As ten participants suggested, a good solution here is to allow players to move back during dialogue. This would allow players to re-listen to the dialogue they missed or did not understand, which was a common complaint to be discussed below. Two players suggested using the right arrow key to advance the dialogue and the left arrow key to move back, providing them with full control over dialogue sections.

6.6.2.3 *Audio Menus*

A few usability issues occurred frequently for participants. First, as discussed, a lack of tutorialization led to confusion for players early in their play sessions. A simple dialogue tutorial when the first audio menu is encountered that explains the controls and audio cues for the menu should help avoid this early confusion.

Next, there were four instances of players accidentally selecting a quiz option just as the dialogue section transitioned into a quiz. In these instances, participants explained that they did not realize that the audio menu had started and they were trying to advance dialogue. Two possible solutions could resolve this issue.

First, the prompts that precede all audio menus are not consistent and should be rewritten to clearly tell players that they must select an option from the list. Second, a change in controls could ensure players don't accidentally select a menu option. Currently, dialogue can be advanced with any button on the keyboard, and audio menu options are selected using the space bar. To solve this, players should be required to use different keys for advancing dialogue and selecting a menu option to ensure that an accidental button press does not lead to missed dialogue or a failed quiz.

Finally, five participants chose the wrong answer during the last quiz question. The final quiz questions in the module are re-used questions from earlier and are intended to re-enforce the player's knowledge of the content presented during the module. However, the final quiz question is slightly altered from its earlier counterpart, changing from "Which of these is a healthy fat?" to "Which of these is *not* a healthy fat?". It's likely that players did know the answer, however, did not realize the quiz question had been altered. In fact, participants that chose the wrong answer expressed that they did not hear or expect the "not" in the question and were not able to listen to the question again to confirm. Including an option to re-listen to an audio menu prompt could help alleviate this issue.

6.6.2.4 *Text-To-Speech*

Ten participants mentioned that the text-to-speech voices were hard to understand and needed clarification for some words and phrases, especially any words created for *Foodbot Factory*, such as "Foodbot". Participants also found that the lack of variation in the text-to-speech voices led to confusion about which character was speaking, especially during scene transitions when new characters joined or exited the dialogue. Participants were also observed switching between audio menu options before the prompts were read out.

These issues can be explained by *Foodbot Factory*'s text-to-speech implementation, specifically the RT Voice Unity plugin. RT Voice uses the device's native text-to-speech voices to turn written lines into speech at run-time. In other words, the text-to-speech voices used for *Foodbot Factory* come from whatever device is being used to run *Foodbot Factory*, and voice audio is generated as the player plays, for this study Microsoft Windows voices were used by RT Voice. This run-time implementation provides flexibility to developers, as they can change dialogue at any stage and simply use the new lines as input when players run the game. However, the disadvantage is that developers cannot directly control how voices will sound and cannot ensure that pronunciations will be correct, especially since voices vary between different devices. The run-time processing also adds a slight delay to each spoken line which can slow down the game by creating unnatural pauses during dialogue and audio menus.

Participants suggested that real voices would be easier to understand and would help with their engagement since play sessions mostly consisted of dialogue sections. Participants also suggested using a different narrator voice to tutorialize and provide context for the characters and settings.

6.6.2.5 *Volume Options*

Eight participants felt that the end-of-dialogue line sound cue should be louder. This cue, which sounds like a soft click, plays at the end of each dialogue line to prompt the player to advance to the next dialogue line. Advancing to the next line causes a similar clicking sound to play. Some participants did not notice the end-of-dialogue line sound until they later discussed dialogue sections with the author. The cue was implemented to be quiet as it is played very frequently and could become intrusive if too loud. However, an options menu with a volume mixer would allow players to adjust the cue's volume as desired. This cue and volume option could be explained in a dialogue section tutorial. Likewise, three participants felt that the lab setting's ambient noise was too loud and distracting, therefore the volume options could also include background ambience, voiceovers, and sound effects so players have full control over their experience.

An options menu could also include various gameplay and button mapping customization. For example, two participants suggested including an option to play dialogue automatically, so they wouldn't need to press a button to advance each dialogue line. Button mapping options could also allow players to choose their own button configuration.

6.7 RECOMMENDATIONS FOR FUTURE UPDATES

The following subsections contain recommendations for future updates of the *Foodbot Factory* audio game, based on the findings of the usability study. These recommendations are organized into three categories based on the effort required for their implementation. All recommendations vary in their estimated impact on the usability of *Foodbot Factory*. Implementing the low-effort recommendations should have a positive impact on usability while only requiring minor changes to current code systems and edits to current dialogue scripts. Medium effort recommendations require updates or additions to current code systems and the addition of new dialogue scripts (new scripts require an additional implementation into the Unity project). Finally, high-effort recommendations require overhauling current code or dialogue systems, which requires significant time for design, implementation, and testing.

6.7.1 *Low-Effort Recommendations*

- Short dialogue tutorials should be included when players first encounter each interaction type [103, 65]. These tutorials can be written into current dialogue scripts, generally at the start of the related audio menus or dialogue sections. For example, the first dialogue section in the Protein Foods Module could include a short tutorial that introduces the dialogue section controls and the end-of-line sound cues. Participants in the study who needed assistance only required brief prompts from the author, therefore these short tutorials should alleviate this early confusion.

- Prevent players from advancing dialogue lines too quickly. This can be accomplished by implementing a timer that resets after each dialogue line is advanced and only accepts the next button input after a short delay. This delay will ensure players do not accidentally skip dialogue lines.
- Use the right arrow key to advance dialogue and use the space bar for selecting audio menu options [38]. Controls are easily adjusted within the Unity C# scripts that implement both of these interactions. This control scheme will ensure that players do not accidentally select audio menu responses when attempting to advance dialogue lines. Other control scheme ideas could be tested with new participants as well.
- Update dialogue scripts to remove short lines and provide clear setting and character transitions [103, 38]. This recommendation requires developers to carefully read through all dialogue scripts and to restructure any problem sections, including areas with short lines and areas where characters transition between settings. For example, the "Go!" dialogue line just before the audio mini-game begins. This will ensure dialogue sections are clear and flow naturally.

6.7.2 *Medium Effort Recommendations*

- Allow players to replay the current line of dialogue, including audio menu prompts. This recommendation requires code to save the current line of dialogue after it has been processed, code to detect a new key input (perhaps the "R" key for "replay"), and code to detect the current state of the game to determine if an audio menu prompt needs to be replayed in addition to the current line of dialogue. This recommendation will allow players to re-hear quiz questions if needed while choosing a response and re-hear dialogue lines they may have missed.

- Provide players with a help button that will replay the tutorial for the current section of the game. This recommendation requires code to detect the current state of the game, the addition of new dialogue scripts that contain the tutorials for each section type, and code to play the tutorial when a new key input is detected (perhaps the "H" key for "help"). This would allow players to the tutorial for each section as needed, instead of having to ask for external assistance.

6.7.3 *High Effort Recommendations*

- Use pre-rendered audio for dialogue voiceovers. Unique voices should be used for each character and developers should ensure that dialogue is easily understood. This recommendation requires one of two implementations. First, developers could create a tool for processing dialogue scripts and generating audio files from text-to-speech voices. Second, developers could record real voices. Both of these solutions would then require a code system to pair each written dialogue line with its audio file counterpart. The benefit of the text-to-speech tool solution is that dialogue audio could be quickly regenerated if the dialogue scripts need to change. However, real voices may provide a more engaging user experience. Both implementations could be used in suggestion, generating text-to-speech voices for placeholders, then using real voices once the dialogue has been finalized.
- Give players complete control over dialogue sections, allowing them to use the left and right arrow keys to move through dialogue in each direction. This recommendation would provide players with complete control over dialogue sections and allow them to move through dialogue at their own pace. This full dialogue control requires code to save a history of encountered dialogue lines, code to process dialogue in reverse order, and code for button input. The *Foodbot Factory* audio game was designed to be played without

visuals, however, it does contain visuals, thus dialogue script commands that move character portraits during dialogue sections would need to be updated to work in reverse order. Additionally, dialogue sections are not linear, as correct and incorrect quiz responses lead to different paths in the dialogue scripts, and therefore developers would have to implement a solution for traversing these paths in reverse.

6.8 limitations and future work

This study was held online during December 2021, amidst Covid-19 lockdowns, making it difficult to recruit participants within the target demographics of this work, and thus, adult participants were recruited instead. These participants were known personally by the author and all participants were aware that the author had assisted in developing *Foodbot Factory*, thus responses may have been biased and skewed positively out of politeness.

As mentioned above, in addition to adults, there are two demographics of players that would have ideally been recruited for this study. First, as with the other *Foodbot Factory* studies [41, 63], children in grades 4-6 as they are *Foodbot Factory*'s target demographic. Second, participants with disabilities, especially adults and children with low vision or blindness, as the accessibility and audio game features of *Foodbot Factory* were being examined. Unfortunately, recruiting from these groups was not feasible at the time of the study, due to Covid-19 lockdowns as testing online with children was not possible and the Parsec tool required to play the *Foodbot Factory* audio game remotely is not accessible to people with low vision or blindness [12].

To address these limitations, future work with *Foodbot Factory* could include the low and medium effort recommendations described above and be tested with children in grades 4-6 including individuals who experience low vision and blindness. Additionally, this work focused specifically on accessibility for people who experience low vision and blindness, future work could include wider areas of accessibility including hearing, and motor control, amongst others.

6.9 CONCLUSION

The purpose of the study was to examine the usability of a vertical slice and gather player feedback to help inform the full implementation of accessibility and audio game features into all modules of *Foodbot Factory*. Specifically, the usability of the audio interfaces and the understandability of the Protein Foods module content, of adult users who played without visuals. Throughout the play session participants discussed usability issues they encountered and observational notes were taken. After the session participants completed a system usability scale questionnaire and were given more time to discuss the game.

Overall, the system usability scale results were positive, with a usability score of 81, indicating above-average usability. Recall that an SUS score above 68 denotes above-average usability and a higher score indicates a higher degree of overall system usability [92]. Participants enjoyed their time with the *Foodbot Factory* audio game, were able to follow along and enjoy the audio-based dialogue, and were all able to complete the play session successfully with little assistance from the author. However, feedback and observations indicated several usability issues. Most notably unnatural text-to-speech dialogue and a lack of tutorials for dialogue sections and audio menus.

Recommendations organized into three categories based on the effort required for their implementation were provided to help eliminate these usability issues in future *Foodbot Factory* and accessibility work. These recommendations include updating the dialogue to flow more naturally, providing tutorials for each interaction type, including a help button to replay tutorials, allowing players to replay dialogue, and more. Future work should include these recommendations and be tested with children from *Foodbot Factory*'s target demographic and people with low vision and blindness.

While *Foodbot Factory* was not initially designed with accessibility in mind, this study shows that audio-game features can be effectively implemented retroactively to allow players to play without visuals. More work needs to be done to fine-tune and test these features but ultimately this work takes *Foodbot Factory* one step closer to universal accessibility.

7 CONTRIBUTIONS & CONCLUSION

This thesis is motivated by three gaps in recent literature.

First, while serious games have been widely discussed, the development methodologies and implementations of these games are often overlooked in lieu of discussing effectiveness [66, 114, 32], categorization of serious games [99], the adoption of serious game approaches in other research [33], and gamification theory [78]. Game development methodologies including project management methods, communication processes, tool recommendations, development pitfalls, and other workflow-related advice are often absent in serious game work.

Second, most video games and serious games, including *Foodbot Factory*, are not completely accessible and often cannot be played by individuals with disabilities. Greater work is needed to encourage and facilitate the development of accessible games [70, 74].

Third, the audio game community, including players who experience low vision and blindness, is eager for quality audio games and there is a lack of research detailing audio game development [73, 58, 38].

Based on these motivations, this thesis provides two main contributions.

First, it provides a set of recommendations for serious game developers based on the learning experiences of the *Foodbot Factory* developers, found in Chapter 4. These are pragmatic recommendations that focus on keeping project scope achievable, finding sustainable project management approaches, and embracing growth. When faced with challenges that seem insurmountable, student developers should take a step back, reduce the scope, and simplify their approach.

Second, this thesis contributes to audio game and accessibility research, by providing a discussion of the approaches used to update *Foodbot Factory* with accessibility features for players with low vision and blindness, and describes a usability study with recommendations for future updates. *Foodbot Factory*'s accessibility is analyzed using the Web Content Accessibility Guidelines [10] and the Game Accessibility Guidelines [1], then is updated with accessibility features inspired by audio game research. These features include keyboard support, audio menus for navigating and selecting options in a list, full text-to-speech voiceover support, and a new audio mini-game, all playable without visuals.

Thus, this thesis statement can be revisited:

> Can audio game research be used to update *Foodbot Factory* with accessibility features to support players with low vision and blindness?

There are two parts to the statement above that should be addressed individually. First, can audio game research be used to update *Foodbot Factory* with accessibility features? Absolutely, yes. Keyboard controls, audio menus, full text-to-speech integration, and an audio-mini game were all added to *Foodbot Factory* based on recommendations found in audio game research. With little help from the author, all participants in a usability study were able to successfully complete a *Foodbot Factory* play session without any visuals. Audio game research and audio games, in general, can provide a great deal of inspiration for accessibility features and should be utilized more in general accessibility work. Second, can these features support players with low vision and blindness? The user study results present a strong case that the accessibility features added to *Foodbot Factory* would be effective, as participants were able to play successfully while visuals were disabled. However, the user study was held online with adult participants known to the author and no participants are known to experience low vision and blindness. While this study provided excellent feedback, it only represents a starting point, as the true needs and expectations of players, including children, who experience low vision and blindness are unknown and their feedback is essential for ensuring that *Foodbot Factory*'s accessibility features are as effective as possible.

Ultimately, until audio games and accessible features in video games become widely prevalent and effective, greater work is required, particularly work that describes the development of accessible games and is aimed at getting games into the hands of users with disabilities. Through user studies, accessibility research, audio game research, and audio games in general, we can learn more about current approaches and find more opportunities for improvement. Together researchers and developers can encourage inclusive design during the development process and ensure games are made for *all* players.

A

APPENDIX A: FULL ITERATION DETAILS FOR THE DEVELOPMENT OF FOODBOT FACTORY

The following sections detail each iteration of *Foodbot Factory*, its related playtests and its evolving requirements list. The playtest results in the following sections are anecdotal observations made by the author while observing players. For the full playtest details, data, and results please refer to Brown et al. and Froome et al. [41, 63].

A.1 ITERATION 1

Although *Foodbot Factory* was developed following an iterative approach, using ideas from Agile and Scrum, most of the team did not have much prior experience with game development and the team's working culture had yet to be established. Therefore, the development team did not adopt an Agile approach during the earliest phase of the project and later transitioned, after early conceptual decisions about the app had been made. The early meetings were generally used to conceptualize the high-level decisions of the project from each of the education, health, and serious game perspectives. Each week the full Bodyzone team met to discuss the concept for the app and would verbally decide on some tasks for the week. This meeting structure continued for the first few months of the project.

The app's concept was somewhat vague to begin with. Under the original proposal, the Bodyzone team was tasked with creating three health curriculum related applications for elementary school kids. The team's first focus being an app with a nutrition theme using the upcoming 2019 Canada's Food Guide. This information formed the development team's first, somewhat vague, requirements lists:

- Nutrition application, based on the upcoming Canada's Food Guide.
- Gameplay that can convey health messages.
- Facilitates student/teacher interaction.
- Fun and engaging.

Using these requirements a few game concepts were brainstormed and pitched during meetings:

A.1.1 *Town Explorer*

A town exploration game where players can travel to different parts of town to learn about foods. The locations in the game are all food related (bakery, grocery store, restaurant, etc.) and allow students to learn about different foods with a real-world connection. As players complete tasks at each location they would be rewarded with different clothes or items used to customize a player avatar.

This design idea did meet the requirements at the time, a nutrition application with fun gameplay that could convey health messages. However, it was somewhat ambitious and while it presented the high-level gameplay, it did not contain ideas for how the nutrition messaging would be conveyed to players.

A.1.2 *Nutrient Shooter*

Inspired by the retro arcade game Galaga, players control a small ship inside of a stomach. As foods and nutrients enter the stomach, players need to zap foods they think are bad and allow good foods to pass.

The goal of this idea was to incorporate the nutrition messaging into the gameplay without the need for text, allowing players to learn by playing. It's important to note here that considering foods good or bad was a naive idea presented at the time by the author, and was later discarded in *Foodbot Factory* for messaging that presented foods as more or less nutritious and encouraged players to make better choices, such as making your own chicken nuggets at home.

A.1.3 *Visual Novel*

Players engage in a short session of nutrition based dialogue, then are quizzed about the messaging.

This is the design the team chose to pursue and the design that would later serve as the basis for the final iteration of *Foodbot Factory*. It was the simplest idea at the time, however this simplicity served two main purposes.

First, as the specific messaging of the upcoming Canada's Food Guide had yet to be released, this idea would allow flexibility during development. Dialogue systems and tools could be implemented in advance, then later when the food guide was released, specific dialogue with messaging could be created and easily added into the game.

Second, the current requirements list was vague and the team had to learn more about how the target demographic, elementary school kids, interacted with this type of app in the classroom. A small scope allowed the team to develop a proof-of-concept app that could be taken to a classroom for kids to play. This was the start of the team's iterative player-based development style. The team could observe, interact with, and learn from players then update the requirements list to be more specific to their needs.

A.1.4 *Food Log*

An interactive collection of the foods players have learned about during play. Players can select a food from the log to read more about it.

Similar to the Visual Novel idea, the Food Log is included in the final version of *Foodbot Factory*, as seen in 4. The Food Log was created to meet the "Facilitate student/teacher interaction" requirement. After a student completed a dialogue and quiz section, the related foods would be added to the log. Once the entire class completed the section, the teacher could then ask students to open the food log and have a group discussion about the related foods. An early mock-up of the Food Log, shown in 11, included a nutrition fact table that would update to shown information about each food, however this feature was not implemented as it was out of scope for the project.

Figure 11: An early mock-up of the Food Log.

A.1.5 *Food 4 Thought*

After a few weeks of discussion, *Food 4 Thought*, the first iteration of *Foodbot Factory* was in development. *Food 4 Thought* used the visual novel game as its base, with a short dialogue section providing students with messaging about the importance of drinking water followed by a related quiz section. However, during development the team felt that this idea may not be engaging enough to meet the "Fun and engaging" requirement. Although playing a game during class time encouraged students to participate, the development team felt that reading and answering quiz questions was not a big enough departure from simply using a textbook. Therefore, the development team inserted a variation of the *Nutrient Shooter* idea into the game, called the *Food Drop* mini-game, serving as a break-for-play in the middle of the dialogue section. The *Food Drop* had the players moving a robot left and right at the bottom of the screen (using their finger on a tablet's touch screen) while foods and garbage dropped from the top. Players had to move the robot to catch an indicated food and avoid the garbage.

Food 4 Thought became a short game about a scientist and his robot. The scientist explained that he was researching nutrition and needed the student's help to answer some questions. First the scientist introduced the topic in a dialogue section, then asked the students to collect some foods using his robot in a *Food Drop* section, then talked more about drinking water with more dialogue, and finally asked the students a few related multiple choice questions. The game took about 10 minutes to complete, then the related foods would be unlocked in the food log.

Figure 12: An early prototype of the *Food Drop*.

A.1.6 *The First Playtest*

The first iteration ended with a playtest with elementary school students from the Durham Catholic District School Board [41]. Pairs of students were provided two

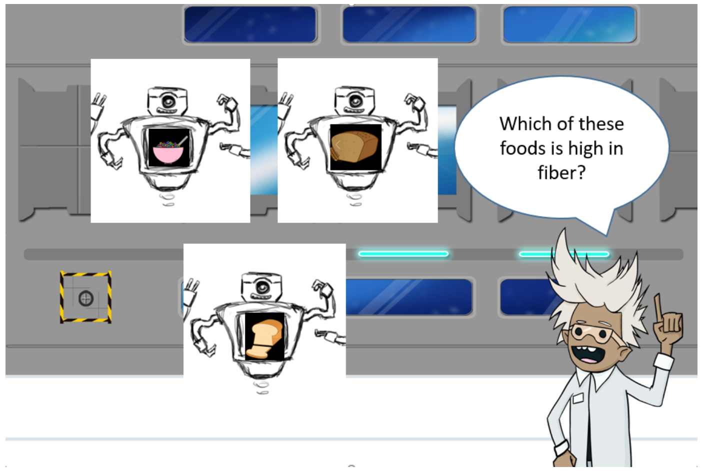

Figure 13: An early prototype of a dialogue quiz in *Foodbot Factory*.

Android tablets to use in tandem while a researcher observed. Overall, students enjoyed the game, especially the students from grades 4 and 5. Some students skipped through most of the dialogue to get to the *Food Drop*, they also ignored the quiz questions and would select random answers. Only some students opened the food log after the main game, but the students who did enjoyed reading all of the information about specific foods.

Food 4 Thought contained a few usability issues, easily observed during the playtest. First, almost all students had trouble advancing dialogue, as the app required them to tap a specific spot on the displayed speech bubbles. Students would try tapping anywhere on the screen a few times before trying to tap the dialogue bubble and would occasionally miss the bubble when tapping throughout the game. For example, the white dialogue bubble shown in Figure 1. This was easily corrected later by allowing players to tap anywhere on the screen to advance dialogue, as is common in other similar games and was the interaction method most of the observed players defaulted to.

Second, many students did not understand the rules of the *Food Drop* and tried catching all falling items on the screen, rather than the food indicated during the tutorial. Tutorialization took place in a brief snippet of the dialogue section before the transition to the *Food Drop*. While the scientist explained the rules, players had yet to be shown the *Food Drop* screen, including the robot, the interface, or the indicated food. Thus, with no visual context, players did not understand the game and many had to be told the rules by the researchers. This was later corrected by transitioning to the *Food Drop* screen before the tutorial and including arrows to accompany the dialogue, shown in Figure 14. Also, the final version of the *Food Drop* uses Foodbot's belly to remind players which food to catch, as seen in Figure 3.

Figure 14: The dialogue tutorial for the *Food Drop*, which includes pointing hands.

Finally, the time it took players to complete the *Food Drop* was inconsistent. To complete the *Food Drop* students needed to catch a certain number of the indicated food before advancing. However, each item dropped was randomized from a list of foods and garbage which led some students to be stuck for many minutes as they rarely encountered the indicated food, while others only got to play for a few seconds when the indicated food dropped in abundance. The rules of the *Food Drop* were later changed to correct this issue. Rather than requiring players to collect a certain number of the indicated food, players were given 30 seconds to catch as many as they could. A weight was also added to the game's programming logic so the indicated food would drop more often.

A.1.7 *Retrospective*

In hindsight, during this early phase of the project, the development team made two main mistakes.

First, the development team did not have a project manager or task documentation. After each Bodyzone meeting, tasks would be loosely decided and members would meet back the following week to speak about their progress. Nearing the end of this phase, the development team needed to "crunch" as they closed in on a scheduled playtest. Having someone on the team to document specific task details, including clear expectations and timelines, would have helped the team achieve its goals much more efficiently. For a development team of three to four members, this Scrum master-type role could have been taken on by any of the developers or shared among the team. Had the development team properly outlined their tasks, scope, and deadlines with formalized project management, the "crunch" could have been avoided.

Second, the developers did not have their own separate meetings apart from the full Bodyzone team meetings. The full meetings did cover important high-level aspects of the project, including the game mechanics, story, and health content of the game. However, these meetings did not cover important development-specific details, such as system implementations, Unity project organization, asset creation, and other task-specific details. Using an Agile or Scrum-based approach and holding stand-up style meetings once or twice weekly could have helped the developers stay on track and work together to solve any unexpected issues that came up during the week, ensuring tasks were progressing smoothly.

Regardless, the first iteration of *Foodbot Factory* was a success. More specifically, the Bodyzone team was able to conceive and develop a good starting point for the app in a couple of months that the end users enjoyed playing, even with some usability and project management issues.

A.2 ITERATION 2

A.2.1 *Overview*

The second main iteration of development was spent clearly defining the high-level design decisions for the application and upgrading the Unity project with a fully-fledged dialogue system, while finally imposing developer-based project management strategies.

Based on the positive feedback of the first playtest, the Bodyzone team decided to keep the visual novel gameplay as the basis for the application. Though until this stage, the premise of the application, including its name, had yet to be decided. Visual novel games generally use character-driven narratives in which the story and inter-character relationships evolve based on the dialogue options the player chooses throughout, for example, this is the case in the popular visual novel *Coffee Talk* [11]. The setting in these visual novel games literally serves as the backdrop, informing the game's visual style, background art, and character designs. However, since the Bodyzone team's goal was to teach about food and nutrition, this character-driven narrative would not work. Thus, after multiple meetings discussing and brainstorming, the Bodyzone team decided on a variation of the "Town Explorer" game idea mentioned in the first iteration with a food-driven narrative, where the characters and setting all serve to facilitate nutrition-based learning.

In *Foodbot Factory*, two food scientists Robbie and Rebecca have created the Foodbots, robots who help villagers make healthy eating decisions. A few variations of the main Foodbot appear throughout the game. Two Breadbots, shown in 15, robots shaped like bread, teach the player about grains, while Superbot, shown in 16, a caped Foodbot, flies around town completing chores in exchange for information about protein foods. Robbie, Rebecca, and many Foodbots travel to a lab, a bakery, a deli, and various other food-related locations to learn about food.

Figure 15: The Breadbot variations of Foodbot.

Visual novels are dialogue-based games, and players mainly interact with the game through dialogue displayed as text on the screen. Players interact by pressing or tapping to advance the dialogue and occasionally selecting a dialogue response from a list. The dialogue is often accompanied by a visual of a character speaking, called a character portrait, that changes emotion based on context. Depending on their scope, these dialogue systems often require implementations that support branching dialogue, voiceovers, language localization, and tools for developers to quickly create and implement scripts. With the decision to pursue a visual novel design, *Foodbot Factory* would now need a robust dialogue system. During this iteration, the Unity project would need to support branching dialogue, dynamic character portraits, and a tool for editing scripts. Greater details regarding the requirements, design, and implementation are discussed below.

This dialogue system became the largest technical challenge of the project, however implementing a few Agile-based project management techniques ensured the scope was achievable and development ran smoothly. The development team began using two Trello Kanban boards for tracking tasks. First, a board used for art assets (character portraits, backgrounds, food items). Art assets were tracked using a sprint-style method. Each Monday the team met to review the needed assets in the backlog and select the highest priority items to be completed that

Figure 16: The Superbot variation of Foodbot.

week. Generally, the time it takes to complete similar art assets is consistent, so as the sprints progressed the team could easily estimate how many assets could be completed in a week and planned accordingly. In contrast, while using the second Kanban board for feature implementations, sprint cycle length varied greatly. Estimating the time required to complete programming tasks can be difficult, especially for students who are often learning "as they go". Thus, programming sprints lasted days to weeks but were being clearly tracked and communicated during stand-up style meetings held once or twice a week. This method, while still flexible, provided the needed structure that the previous iteration had been missing and ensured features and the dialogue system were implemented efficiently.

A.2.2 *Requirements*

Most of the requirements from the first iteration were met by the current design. The nutrition-based visual novel would meet the "nutrition application" and "gameplay that can convey health messages" requirements, and the food log would "facilitate student/teacher interaction". However, based on observations and player feedback "fun and engaging" was not completely met. Students found there was a lot of dialogue and little play, and often resorted to skipping through dialogue quickly to reach the food drop [41]. Thus, "Dialogue isn't tedious" was added to the requirements list, encouraging the team to create a dialogue system that enforces the "fun and engaging" requirement.

Two other requirements were added to the list based on the playtest findings. First, "students should not need teacher instruction to play". There were multiple instances of usability issues that led students to ask the researchers for direction [41], discussed above in the first iteration section. Ultimately, this application should allow teachers to disengage from the class while they play, then reengage to discuss the nutrition information when the class finishes playing. The application should provide sufficient guidance for students so they don't need to ask for assistance, or simply get frustrated when instructions in the game aren't clear.

Second, "modules should take 10 to 15 minutes to complete". Most students completed the playtest in about 10 minutes, which was long enough to convey the needed nutrition messages while still providing flexibility for teachers. The final *Foodbot Factory* application contains five modules that can easily be slotted into a schedule individually or used in tandem for longer sessions.

Thus, the updated requirements became:

- Nutrition application, based on the unknown Canada's Food Guide.
- Gameplay that conveys health messaging.
- Facilitates student/teacher interaction.
- Fun and engaging.
- Dialogue isn't tedious.
- Students should not need teacher instruction to play.
- Modules should take 10 to 15 minutes to complete.

A.2.3 *Design and Tasks*

During this iteration, the project progressed from the conception phase into full development. Now that *Foodbot Factory* had a concrete visual novel design and premise, the development team could clearly define features and tasks needed for the project. Generally, tasks were broken down into "content" and "tech" themes. Content tasks detailed the creation of art assets (character portraits, backgrounds, etc.), audio (sound effects, background ambience, etc.), and dialogue script writing. Tech tasks detailed programming, Unity project organization, and implementing the deliverables from content tasks.

Keeping the scope contained for this iteration, the goal was to re-create the Drinks module from the previous iteration but with an upgraded dialogue system, new art, and script. The breakdown of content and tech tasks were as follows:

A.2.3.1 *Content*

- Overview of the premise for the Drinks module.
- Character designs for Robbie, Rebecca, Foodbot, and two villagers.
- Multiple dialogue assets for Robbie and Rebecca.
- Basic dialogue assets for Foodbot and villager.
- Animation for Foodbot's movement in the *Food Drop*.
- Lab and village background art.
- Lab and village background audio.
- Updated art for water, pop, and milk.
- Script in a Word document.

The first step for deciding on content tasks was to decide on a premise for the Drinks module. After some brainstorming, it was decided that the premise would be about Robbie and Rebecca fixing a Foodbot whose "Drink functions" were broken. The Foodbot would be brought to their lab where they would re-teach the Foodbot messages about drinks, then they would take the Foodbot out into the town to be quizzed by villagers. Having a concrete premise then allowed the development team to detail the needed content tasks above.

The most time-consuming of these content tasks was creating the designs for all the characters and creating their dialogue assets. Designs were discussed in multiple steps, where team members would bring ideas and references to the artists on the team, the artists would create multiple sketches for the team to discuss, then finally the artists would create the final assets based on the team's feedback. Also, to ensure that the dialogue was fun and interesting, the team wanted to use a common visual novel mechanic where the character portraits change expressions during dialogue. However, to keep the scope lower, only variations of the portraits for Robbie and Rebecca were created, as they were planned to be re-used in future modules. These variations included neutral, happy, sleepy, and excited, to be used interchangeably throughout the dialogue. Examples of these portraits are shown in Figures 17, 18, and 19.

The process of writing the script also proved to be challenging. The scripts for all the *Foodbot Factory* modules were great collaborative efforts between all members of the full Bodyzone team. Nutrition messaging needed to be well informed by members of the Health faculty, the structure of the dialogue and quiz questions needed to be informed by members of the Education faculty, and finally system designs needed to be enforced by the game developers. Over multiple iterations of the project, the process for writing the scripts became as follows: (1) the health members decide on the main foods and messages for the module and provide developers with a short overview, (2) the developers create a rough draft of the premise and script for the module, (3) the Bodyzone team provides feedback on the script, (4) the script is updated and implemented into the game by developers, (5) the Bodyzone team plays the module and reviews the script, (6) repeat steps 3 through 5 as needed. This feedback and update cycle was often repeated multiple

Figure 17: Neutral character portrait variation.

Figure 18: Happy character portrait variation.

Figure 19: Sleepy character portrait variation.

times as different perspectives would read and play through the script. This review process also needed to be revisited for all modules when Canada's Food Guide was released in 2019, as the messaging in the scripts needed to match.

It's important to note here that reviewing a Word document and actually playing a game can change a team member's perspective of the dialogue content. For example, in Foodbot Factory each line of dialogue is limited to one or two short sentences, to keep text properly contained and readable on speech bubbles. Early in the process, team members would often add updates to the script that did not fit this line length limitation, leading to more cycles of the review process. Some sections also became too long, segmented, or tedious to play, even though reading through the script document only took a few minutes. Thus, it became important for developers to continually provide playable builds of the game alongside scripts for review and to clearly communicate the limitations of the system for team members that were not familiar with visual novel-style games.

A.2.3.2 *Tech*

- Overview of the premise for the Drinks module.
- Character designs for Robbie, Rebecca, Foodbot, and two villagers.
- Multiple dialogue assets for Robbie and Rebecca.
- Basic dialogue assets for Foodbot and villager.
- Animation for Foodbot's movement in the *Food Drop*.
- Lab and village background art.
- Lab and village background audio.
- Updated art for water, pop, and milk.
- Script in a Word document.

As the project moved out of the prototyping phase, the development team decided to create a new Unity project and move over any of the code and art that was planned to be re-used. This allowed the team to purge anything that wasn't needed for the future, removing clutter from the project, and implement organizational standards for the project's file and folder structure. While re-creating the Unity project did add some additional development time up front, it made future development quicker overall as developers could easily collaborate within the project by knowing where to find files, assets, or code other team members had contributed. The team also implemented coding standards for the C# scripts used for programming in Unity. Some of these standards included adding "m_" to member variables in classes, using tabs instead of spaces, and adding a comment to each class that describes its purpose. Again, these standards add consistency to the project to help developers work collaboratively and stay organized.

To create a visual novel with different scenarios and mini-games mixed throughout multiple unique modules, the Unity project required a system for switching between Unity scenes in a predetermined order. Each scene can be thought of as a discrete portion of gameplay. For example, the dialogue section where Robbie and Rebecca discuss drinks messaging in the lab is a scene and a *Food Drop* mini-game is another scene. This system had to be flexible enough to support multiple different scene types and allow developers to easily create new sets of scenes used for modules in later iterations. The implementation created during this iteration used a SceneManager object holding a list of scenes to visit in linear order and an index used to keep the current place in the list. When a module is started, the SceneManager is updated with the corresponding list of scenes. Other scene-specific objects could be given access to this manager to let it know when to move to the next scene. In later iterations, the SceneManager was updated to support branching paths for the animal protein module, so players could choose which locations to visit, rather than being forced along a predetermined path.

The dialogue system for *Foodbot Factory* had to be flexible and easy to use. The goal was to develop a tool that the developers could use to implement dialogue scripts with added functions for controlling the in-game scene. These functions would be written alongside the dialogue itself and could be used for branching dialogue paths, supporting quiz questions, controlling characters, and anything else that might be needed in the future. After some research, the development team decided to implement the plug-in Yarn Spinner [31]. Yarn Spinner is an open-source dialogue tool that has been used in popular indie games such as *Night in the Woods* and *A Short Hike*. Yarn Spinner is capable of supporting the needed functions described above and is easily expandable since it is open-source. Yarn Spinner did take the team a few weeks to integrate and customize for *Foodbot Factory*. However, developing a new system with all the same functions would have likely taken longer to develop and test. Using a well-established tool with a robust feature set injects predictable quality into the project, making the game more enjoyable to develop and play.

A.2.4 *Playtest*

After a few months of part-time development, the new drinks module was complete and ready for testing. Gameplay consisted of four dialogue sections that alternated between three variations of the Food Drop and ended with a quiz section.

Similar to the previous playtest [41], the researchers used a semi-structured observation method with a printout containing a few pre-determined questions, such as "Did the player skip any dialogue?", and a large section for rough notes. The app was installed on six Android tablets and grade 4 students from a local elementary school played around a table, so they could be observed by two researchers at once. Students were asked to play through the drinks module, then were taken aside in pairs to be interviewed briefly about their experience [41].

Students smiled and laughed along with the dialogue, some even reading out loud using different funny voices for all of the characters. Students expressed that they liked the story and told the researchers about their favourite characters, Robbie being a common choice. Though, some students still skipped through dialogue to reach the mini-games as quickly as they could and commented on the large amount of reading.

A few usability issues were observed during the playtest. First, the dialogue bubble containing the quiz question, shown in 2, did not persist while players needed to make a selection, so some players forgot or missed the question and had to make a random choice. This was later changed so that the dialogue bubble would persist until students selected a response. Next, there was no loading screen feedback, so after tapping the button to start the module the game would freeze briefly before transitioning to the first dialogue scene and some students would begin frantically tapping the button until the game responded. A loading screen was added in the next iteration, so students received instant feedback for starting a module. Finally, the Food Drop from the previous iteration had been updated with new visuals using a Foodbot to catch foods. Some students found the Foodbot's acceleration to be too slow, especially when changing directions, which led to some frustration as players couldn't catch the required food. Foodbot's acceleration variable was easily increased in the next iteration.

Overall, the feedback was very positive and students expressed they would love to play again when more modules were available. Even with some dialogue skipping and usability issues, the team felt very confident that the visual novel gameplay was a good fit for the project and decided to pursue creating more modules in the next iteration.

A.3 ITERATION 3 AND BEYOND

Iteration 3 followed much of the same process as the previous iteration. The Bodyzone team selected the food topic, grains foods, for the next module, and the development team created new content and tech tasks to organize the development. During this process the development team once again reviewed and updated the requirements list, using feedback from the playtest. The team concluded that the requirements did not need to be altered, however, the team felt that the "Dialogue isn't tedious" requirement was not being met for some players. In response, the Bodyzone team implemented two new features to make the dialogue sections more accessible and engaging.

First, a voiceover setting was added to the app which allowed players to listen to the dialogue as it was being presented. This feature was recommended by the education members of the Bodyzone team to increase accessibility, as reading levels among students in grades 4-6 can vary significantly. The voiceovers were implemented using a plug-in RT Voice, which uses the text-to-speech voices available on a device's operating system. Unfortunately, using onboard text-to-speech voices gives the developers less control; the tone, pitch, and volume of the voices can be unpredictable. However, the development team was able to integrate RT Voice and Yarn Spinner together to process dialogue lines at run time, which means dialogue did not need to be pre-recorded and scripts could still be easily updated, lowering the costs of adding voiceovers.

Second, the development team created a new scene type, a hybrid between dialogue and mini-game scenes called dialogue interactions. Dialogue interactions have the player tap and drag objects on the screen to reveal new dialogue text, rather than simply tapping anywhere to advance linear dialogue. For example, the dialogue interaction used in the grain foods module has the player looking through a microscope at the parts of a grain. They choose to learn about the endosperm, germ, and bran in the order they choose, as shown in 5. Then they are directed to drag away the bran and germ, mimicking the refining process used to create refined grains. The intention of these iterations is to provide another method for delivering nutrition messaging while giving the player a break from linear reading.

This iteration and each after ran smoothly using the process outlined in iteration 2. As each module was completed, it was tested with students from a local elementary school for feedback and to identify any minor usability issues [41]. Feedback was consistently positive and no large changes were required. Much of the work beyond this iteration was content focused, as no new systems needed to be added. The work primarily consisted of collaborating on scripts, creating art and audio assets, and implementing it all into scenes. Though, there were a few design changes and tasks of note.

A variation of the *Food Drop* called the *Food Sort*, was created to add variety to the mini-game sections. In the *Food Sort*, foods fall onto conveyor belts and players must sort them into corresponding buckets. Players can tap the conveyor belts to change the direction they move the food.

As mentioned previously, when the content of Canada's Food Guide was available to the Bodyzone team the scripts needed to be reviewed and updated to ensure all nutrition messaging was consistent. Thanks to the insight of the Health members of the Bodyzone team, only minor changes were required; the foods and modules previously chosen were consistent with the food guide. Most of the updates were semantic in nature, for example changing the word "grains" to "grain foods". The main menu was also updated to use the sectioned-plate design from the food guide [14] as the interface to select a module, as shown in Figure 20. This menu change also included the decision to not gate the modules, so players could play all modules in any order.

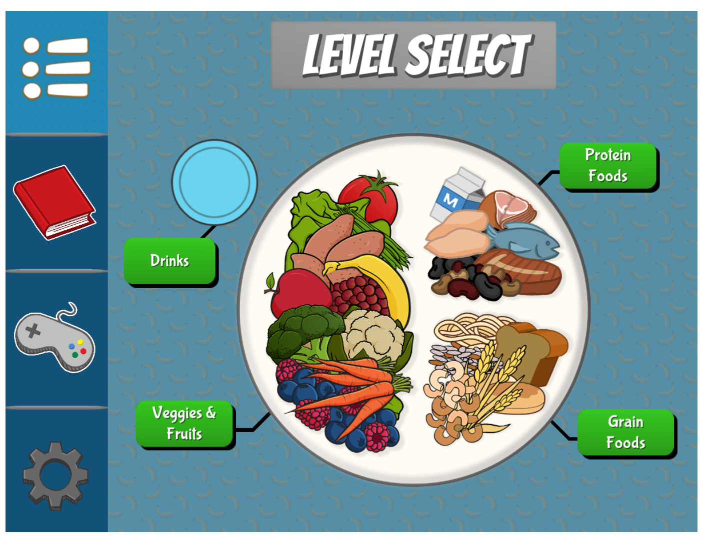

Figure 20: The module selection screen in *Foodbot Factory*, designed to mimic Canada's Food Guide.

As the development team became comfortable with the Yarn Spinner integration for *Foodbot Factory*, the team made a few "quality-of-life" upgrades to make the scripts shorter and easier to use. Originally, the team was using commands such as "«Move Robbie OffScreenLeft»" and "«Move Rebecca OnScreenLeft»" to move characters on screen, together these commands move Robbie off screen and then bring Rebecca on screen. The team noticed this pattern of swapping the positions of two characters was quite common so they added a "swap" command. Thus, the commands above could be shortened to "«Swap Robbie Rebecca»". Another notable dialogue change was a shorthand for changing character expressions. Originally, to change a character's expression the script would contain the SetSprite command, for example "«SetSprite Rebecca Happy»" would change Rebecca's character portrait to the happy variation. These expression changes were very common, often occurring with each line of dialogue. Thus, the team created a shorthand by using emoji-style characters at the end of each dialogue line. For example, the dialogue line in the script now read "Rebecca: Welcome to the lab! :)". Since this line ends with the ":)" text, symbolizing a sideways smiley face, the dialogue would be read and Rebecca's expression would change to the happy variation. Other expression shorthands included, ":|" for neutral, ":O" for excited, and ":S" for sleepy.

The last notable change came near the end of the project after one of the final playtests which focused on the protein foods module. For context, this module has a caped Foodbot variation, named Superbot, who flies around town to multiple locations to learn about different protein foods. After a short introduction, players were able to travel to three separate locations in the order they chose. During the playtest, the team observed that the module was too long to play, roughly 20 minutes. As 10-15 minutes was the goal, based on the team's requirement and to be consistent with the other modules, the module needed to be updated to be shorter. After some discussion, the Bodyzone team decided to split the module into two smaller modules, an animal protein foods module, and a plant protein foods module. The module's segmented structure easily allowed the developers to purge the plant protein scenes and create a new module. This also allowed the scripts for each module to grow slightly and include more detailed nutrition messaging.

APPENDIX B

B.1 USABILITY STUDY CONSENT FORMS

Script (*Posted in the google form*)

Foodbot Factory (2D & AR) Pilot Study Consent Form

We are inviting you to participate in a research study regarding accessibility features of a mobile serious game.

We are teachers and researchers at the Ontario Tech University (OTU), that are studying the use of the new Foodbot Factory, which is a mobile serious game (i.e., a game designed for educational purposes) created to teach children in Grades 4 and 5 about healthy eating. This app has two versions: 2-Dimensional (2D) and a newly developed Augmented Reality (AR). Both contain the same nutrition information and they only differ on how the information is presented to children. This project has been informed by experts in game development, dietetics, and education, to create a fun and interactive platform for healthy eating education. In order to optimize our app's development, we need feedback from teachers, parents and/or caregivers, in addition to children, throughout the development process to ensure that what we are creating is appropriate, engaging, and most importantly educational for this group of children.

We are asking for you consent to participate in our research project called, "Foodbot Factory". This project includes the use of a digital environment containing various apps and resources that students, teachers and parents can access. The app will help with the teaching and learning of nutrition curricula, with an emphasis on healthy eating messages and nutrients that relate to chronic disease.

Once consent is given, you will need to download the remote desktop tool Parsec and create an account in order to play the app during the session. You will be provided with a checklist to prepare yourself for the Zoom call. This checklist includes: installing Zoom on your computer, installing Parsec on your computer, checking the internet connection, ensuring you are located in a quiet, distraction-free location. On the day of the Zoom call, you will be emailed the call details. You will join a Zoom call where they will play with the app, describe their experience with the app, and if the app is adequate for children.

Information will be collected from adult participants at various times during the project. We will gather information in the following ways:

• We will observe you during the Zoom online session and take notes about how you participate in the Zoom session, how you interact with the app, comments you make and/or ideas you share

• We would like you to complete an online questionnaire after interacting with the app

An overview of the research tools and time allotted for each tool include:

- Piloting of the app: ~20-25 minutes of game play.
- Discussion: ~20-25 minutes

- Questionnaire (Google form): ~10 minutes total

Only data collected from adult participants who have given consent to participate in the research study will be used in the analysis and reporting of findings. Findings from this project may be published in journals and presented at conferences.

Voluntary Participation and Withdrawal
You may withdraw from the study at any time without penalty. You can choose not to answer specific questions. If, during any of the above- described activities, you decide not to participate, you can end involvement in the activity by approaching the moderator that you want to withdraw from the study and exit the Zoom call. You may communicate refusal verbally or non-verbally.

You should not feel obligated or compelled to participate in the research for any reason. Participation in the research is entirely optional and you will not be penalized in any way if you do not participate.

Confidentiality
The information you provide will not be stored with personal identifiers, nor will you be identified in any recorded or published comments. The transcripts from the Zoom calls proceedings, as well as any other data collected, will be stored securely at Ontario Tech University under the lead researcher's supervision and will be destroyed after three years or once all data has been published. By consenting to participate, you do not waive any legal rights or recourse.

Participation Benefits and Risks
Potential benefits for participation in this study include: learning new digital tools that align to Canada's Food Guide and gaining understanding of nutrition, and healthy eating.

Potential risks include: feeling pressure to share or embarrassment from sharing work and feeling coerced into participating in the research because the rest of adult participants appear to be. Risks will be dealt with in the following ways: researchers/moderator/volunteers will explain to participants that they have the right to pass when it comes time to sharing their work. They will also explain that participation in the research portion of the activities is entirely voluntary and that you may choose not to participate at any point, without penalty.

Your agreement on the consent form indicates that you have read this letter, understand its contents, and authorize your participation in this research project.

If you have questions concerning the research study, please contact Principal Investigator, Dr. JoAnne Arcand at 647.296.8426 or joanne.arcand@ontariotechu.ca.

Any questions regarding your rights as a participant, complaints or adverse events may be addressed to Research Ethics Board through the Research Ethics Officer – researchethics@ontariotechu.ca or 905.721.8668 x. 3693.

This study has been reviewed by the Ontario Tech University Research Ethics Board REB [#14879] on [insert date].

Thank you for considering your participation in this research study.
* Required
I have read the Letter of Information above. I understand the purpose of this research.

I understand that I have the right to withdraw at any point in time from the study and that withdrawal will not have any negative consequences for me. I also understand that all information collected is for educational/research purposes only.

By agreeing below, I consent to participate in this research study.
Your Name: *

Your answer
Date: *
Date

I consent to participate in the Foodbot Factory Pilot Study *
Yes
No

BIBLIOGRAPHY

[1] Game accessibility guidelines. URL http://gameaccessibilityguidelines.com/full-list/.

[2] Abd gwo-jen, h., 2013. a collaborative game-based learning approach to improving studentsâ learning performance in science courses. *Computers education*, 63. ISSN 0360-1315.

[3] Audiogames.net forum, 2003. URL https://forum.audiogames.net/.

[4] Dig rush, 2015. URL https://www.youtube.com/watch?v=EHDSxadw_Pw&ab_channel=Ubisoft.

[5] Black screen gaming blog, 2016. URL https://blackscreengaming.com/.

[6] Digital Games, Design, and Learning: A Systematic Review and Meta-Analysis. *Review of Educational Research*, 86(1):79–122, 2016. ISSN 00346543. doi: 10.3102/0034654315582065.

[7] A hero's call, Dec 2017. URL https://audiogames.net/db.php?action=view&id=Aheroscall.

[8] Drive, Dec 2020. URL https://audiogames.net/db.php?action=view&id=drive.

[9] Ontario education act, 2020. URL https://www.ontario.ca/laws/statute/90e02.

[10] Web content accessibility guidelines, 2020. URL https://www.w3.org/WAI/standards-guidelines/wcag/.

[11] Coffee talk, 2020. URL https://store.steampowered.com/app/914800/Coffee_Talk/.

[12] Gaming without sight: Levelling up, 2021. URL https://caniplaythat.com/2021/12/03/gaming-without-sight-levelling-up/.

[13] Audiogames.net game list, Mar 2023. URL https://audiogames.net/list-games/.

[14] Canada's food guide, 2023. URL https://food-guide.canada.ca/en/.

[15] Fortnite usage and revenue statistics, 2023. URL https://www.businessofapps.com/data/fortnite-statistics/.

[16] Jira website, 2023. URL https://www.atlassian.com/software/jira.

[17] Rt voice, 2023. URL https://assetstore.unity.com/packages/tools/audio/rt-voice-pro-41068.

[18] Trello website, 2023. URL https://www.atlassian.com/software/trello.

[19] Unity asset store, 2023. URL https://assetstore.unity.com/.

[20] Doozy ui, 2023. URL https://assetstore.unity.com/packages/tools/visual-scripting/doozy-ui-manager-203601.

[21] Market size of the video games industry in the united states from 2013 to 2023, 2023. URL https://www.statista.com/statistics/246892/value-of-the-video-game-market-in-the-us/#statisticContainer.

[22] Github, 2023. URL https://github.com/.

[23] Gitlab, 2023. URL https://about.gitlab.com/.

[24] Google style guides, 2023. URL https://google.github.io/styleguide/.

[25] Jaws text-to-speech, 2023. URL https://www.freedomscientific.com/products/software/jaws/.

[26] Nv access download, 2023. URL https://www.nvaccess.org/download/.

[27] Storyblocks, 2023. URL https://www.storyblocks.com/audio/search.

[28] Unity documentation, 2023. URL https://docs.unity.com/.

[29] Unreal engine 5, 2023. URL https://www.unrealengine.com/.

[30] Vector toons, 2023. URL https://vectortoons.com/.

[31] Yarn spinner, 2023. URL https://yarnspinner.dev/.

[32] Alaa Abd-Alrazaq, Eiman Al-Jafar, Mohannad Alajlani, Carla Toro, Dari Alhuwail, Arfan Ahmed, Shuja Mohd Reagu, Najeeb Al-Shorbaji, and Mowafa Househ. The effectiveness of serious games for alleviating depression: Systematic review and meta-analysis. *JMIR Serious Games*, 10, 1 2022. ISSN 22919279. doi: 10.2196/32331.

[33] Fernando Almeida and Jorge Simoes. The role of serious games, gamification and industry 4.0 tools in the education 4.0 paradigm. *Contemporary Educational Technology*, 10:120–136, 2019. ISSN 1309517X. doi: 10.30935/cet.554469.

[34] Ronny Andrade, Melissa J. Rogerson, Jenny Waycott, Steven Baker, and Frank Vetere. Playing blind: Revealing the world of gamers with visual impairment. *Conference on Human Factors in Computing Systems - Proceedings*, (August), 2019. doi: 10.1145/3290605.3300346.

[35] Oana Balan, Alin Moldoveanu, and Florica Moldoveanu. Navigational audio games: An effective approach toward improving spatial contextual learning for blind people. *International Journal on Disability and Human Development*, 14(2):109–118, 2015. ISSN 21910367. doi: 10.1515/ijdhd-2014-0018.

[36] Sule Biyik Bayram and Nurcan Caliskan. Effect of a game-based virtual reality phone application on tracheostomy care education for nursing students: A randomized controlled trial. *Nurse Education Today*, 79

(February):25–31, 2019. ISSN 15322793. doi: 10.1016/j.nedt.2019.05.010. URL https://doi.org/10.1016/j.nedt.2019.05.010.

[37] Eileen E. Birch, Reed M. Jost, Krista R. Kelly, Joel N. Leffler, Lori Dao, and Cynthia L. Beauchamp. Baseline and clinical factors associated with response to amblyopia treatment in a randomized clinical trial. *Optometry and Vision Science*, 97:316–323, 5 2020. ISSN 15389235. doi: 10.1097/OPX.0000000000001514.

[38] Olimar Borges and Marcia Borba Campos. "I'm Blind, Can I Play?" Recommendations for the Development of Audiogames. *11th International Conference, UAHCI 2017 Held as Part of HCI International 2017*, 10278(October):487, 2017. doi: 10.1007/978-3-319-58703-5. URL http://link.springer.com/10.1007/978-3-319-58703-5.

[39] Olimar Borges and Marcia De Borba Campos. "I'm Blind, Can I Play?" Recommendations for the Development of Audiogames. *11th International Conference, UAHCI 2017 Held as Part of HCI International 2017*, 10278(October): 487, 2017. doi: 10.1007/978-3-319-58703-5. URL http://link.springer.com/10.1007/978-3-319-58703-5.

[40] Jeanne H Brockmyer, Christine M Fox, Kathleen A Curtiss, Evan McBroom, Kimberly M Burkhart, and Jacquelyn N Pidruzny. The development of the game engagement questionnaire: A measure of engagement in video game-playing. *Journal of experimental social psychology*, 45(4):624–634, 2009.

[41] Jacqueline Marie Brown, Robert Savaglio, Graham Watson, Allison Kaplansky, Ann LeSage, Janette Hughes, Bill Kapralos, and Jo Anne Arcand. Optimizing child nutrition education with the Foodbot factory mobile health app: Formative evaluation and analysis. *Journal of Medical Internet Formative Research*, 22(4), 2020. ISSN 14388871. doi: 10.2196/15534.

[42] Paul Cairns, Christopher Power, Mark Barlet, and Greg Haynes. Future design of accessibility in games: A design vocabulary. *International Journal*

of Human Computer Studies, 131(February):64–71, 2019. ISSN 10959300. doi: 10.1016/j.ijhcs.2019.06.010. URL https://doi.org/10.1016/j.ijhcs.2019.06.010.

[43] Antonio Calvo-Morata, Cristina Alonso-Fernández, Manuel Freire, Iván Martínez-Ortiz, and Baltasar Fernández-Manjón. Serious games to prevent and detect bullying and cyberbullying: A systematic serious games and literature review. *Computers and Education*, 157(November), 2020. ISSN 03601315. doi: 10.1016/j.compedu.2020.103958.

[44] Wilfred Van Casteren and Wilfred Van Casteren. The waterfall model and the agile methodologies : A comparison by project characteristics the waterfall model and the agile methodologies : A comparison by project characteristics academic competences in the bachelor 2 assignment: Write a scientific article on 2 software development models. 2017. doi: 10.13140/RG.2.2.36825.72805. URL https://www.researchgate.net/publication/313768756.

[45] Verolien Cauberghe and Patrick De Pelsmacker. Advergames. *Journal of Advertising*, 39(1):5–18, 2010. ISSN 00913367. doi: 10.2753/JOA0091-3367390101.

[46] Ching-Yi Chang, Han-Yu Sung, Jong-Long Guo, Bieng-Yi Chang, and Fan-Ray Kuo. Effects of spherical video-based virtual reality on nursing students' learning performance in childbirth education training. *Interactive Learning Environments*, 0(0):1–17, 2019. ISSN 1049-4820. doi: 10.1080/10494820.2019.1661854. URL https://doi.org/10.1080/10494820.2019.1661854.

[47] Sumedha Chauhan. A meta-analysis of the impact of technology on learning effectiveness of elementary students. *Computers and Education*, 105:14–30, 2017. ISSN 03601315. doi: 10.1016/j.compedu.2016.11.005. URL http://dx.doi.org/10.1016/j.compedu.2016.11.005.

[48] Meng-Tzu Cheng, Jhih-Hao Chen, Sheng-Ju Chu, and Shin-Yen Chen. The use of serious games in science education: a review of selected empirical

research from 2002 to 2013. *Journal of Computers in Education*, 2(3):353–375, 2015. ISSN 2197-9987. doi: 10.1007/s40692-015-0039-9.

[49] Matteo Ciman, Ombretta Gaggi, Teresa Maria Sgaramella, Laura Nota, Margherita Bortoluzzi, and Luisa Pinello. Serious Games to Support Cognitive Development in Children with Cerebral Visual Impairment. *Mobile Networks and Applications*, 23(6):1703–1714, 2018. ISSN 15728153. doi: 10.1007/s11036-018-1066-3.

[50] Erika Corona, Filippo Eros, and Pani Diee. A review of lean-kanban approaches in the software development.

[51] Ticianne G.R. Darin, Rossana M.C. Andrade, Lotfi B. Merabet, and Jaime Hernán Sánchez. Investigating the mode in multimodal video games: Usability issues for learners who are blind. *Conference on Human Factors in Computing Systems - Proceedings*, Part F1276:2487–2495, 2017. doi: 10.1145/3027063.3053177.

[52] Ticianne G.R. Darin, Rossana M.C. Andrade, Lotfi B. Merabet, and Jaime Hernán Sánchez. Slup: A standard list of usability problems in multimodal video games designed for people who are blind. In *2018: Companion Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*, 2018.

[53] Carlos Vaz De Carvalho, Martín Llamas Nistal, Manuel Caeiro-Rodríguez, Andrea Bianchi, Melani Hromin, Hariklia Tsalapatas, Olivier Heidmann, and Alper Metin. Using video games to promote engineering careers. *International Journal of Engineering Education*, 34(2):388–399, 2018. ISSN 0949149X.

[54] Steffi De Jans, Liselot Hudders, Laura Herrewijn, Klara Van Geit, and Veroline Cauberghe. Serious games going beyond the call of duty: Impact of an advertising literacy mini-game platform on adolescents' motivational outcomes through user experiences and learning outcomes. *Cyberpsychology*, 13(2), 2019. ISSN 18027962. doi: 10.5817/CP2019-2-3.

[55] Athanasios S Drigas and Georgia Kokkalia. Jucs$_2$0$_1$0$_1$499$_1$510$_D$rigas.Pdf.20 (10) : 1499 − −1510, 2014.

[56] Ilana Dubovi, Sharona T. Levy, and Efrat Dagan. Now I know how! The learning process of medication administration among nursing students with non-immersive desktop virtual reality simulation. *Computers and Education*, 113:16–27, 2017. ISSN 03601315. doi: 10.1016/j.compedu.2017.05.009. URL http://dx.doi.org/10.1016/j.compedu.2017.05.009.

[57] Zoran Ereiz. Scrum without a scrum master. 2019.

[58] Rickard Falk, Handledare Niklas, RÃ¶nnberg Examinator, and Jonas LÃ¶wgren. Engaging gameplay for audio games examensarbete utfÃ¶rt i medieteknik vid tekniska hÃ¶gskolan vid linkÃ¶pings universitet. URL http://www.ep.liu.se/.

[59] Theresa M. Fleming, Lynda Bavin, Karolina Stasiak, Eve Hermansson-Webb, Sally N. Merry, Colleen Cheek, Mathijs Lucassen, Ho Ming Lau, Britta Pollmuller, and Sarah Hetrick. Serious games and gamification for mental health: Current status and promising directions. *Frontiers in Psychiatry*, 7 (JAN), 2017. ISSN 16640640. doi: 10.3389/fpsyt.2016.00215.

[60] Frans Folkvord and Jonathan van ât Riet. The persuasive effect of advergames promoting unhealthy foods among children: A meta-analysis. *Appetite*, 129 (July):245–251, 2018. ISSN 10958304. doi: 10.1016/j.appet.2018.07.020.

[61] Fernando Fraga-Varela, Esther Vila-Couñago, and Esther Martínez-Piñeiro. The impact of serious games in mathematics fluency: A study in Primary Education. *Comunicar*, 29(69):115–125, 2021. ISSN 19883293. doi: 10.3916/C69-2021-10.

[62] Fernando Fraga-Varela, Esther Vila-Couñago, and Ana Rodríguez-Groba. Serious games and mathematical fluency: A study from the gender perspec-

tive in primary education. *Sustainability (Switzerland)*, 13(12), 2021. ISSN 20711050. doi: 10.3390/su13126586.

[63] Hannah M. Froome, Carly Townson, Sheila Rhodes, Beatriz Franco-Arellano, Ann LeSage, Rob Savaglio, Jacqueline Marie Brown, Janette Hughes, Bill Kapralos, and Jo Anne Arcand. The Effectiveness of the Foodbot Factory Mobile Serious Game on Increasing Nutrition Knowledge in Children. *Nutrients*, 12(11):2–15, 2020. ISSN 20726643. doi: 10.3390/nu12113413.

[64] Ombretta Gaggi, Claudio Enrico Palazzi, Ombretta Gaggi, Claudio Enrico Palazzi, Matteo Ciman, Giorgia Galiazzo, and Sandro Franceschini. Serious Games for Early Identification of Developmental Dyslexia r r ACM Reference Format :. *Computers in Entertainment*, 15(2), 2017.

[65] Franco Eusébio Garcia and Vânia Paula de Almeida Neris. Design Guidelines for Audio Games. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8005 LNCS, pages 229–238. 2013. ISBN 9783642392610. doi: 10.1007/978-3-642-39262-7_26. URL http://link.springer.com/10.1007/978-3-642-39262-7{_}26.

[66] Sarah Victoria Gentry, Andrea Gauthier, Beatrice L.Estrade Ehrstrom, David Wortley, Anneliese Lilienthal, Lorainne Tudor Car, Shoko Dauwels-Okutsu, Charoula K. Nikolaou, Nabil Zary, James Campbell, and Josip Car. Serious gaming and gamification education in health professions: systematic review, 3 2019. ISSN 14388871.

[67] Mustafa Girgin. Use of Games in Education: GeoGuessr in Geography Course. *International Technology and Education Journal*, 1(1):1–6, 2017. ISSN 2602-2885.

[68] Evelim L.F.D. Gomes, Celso R.F. Carvalho, Fabiana Sobral Peixoto-Souza, Etiene Farah Teixeira-Carvalho, Juliana Fernandes Barreto Mendonça,

Roberto Stirbulov, Luciana Maria Malosá Sampaio, and Dirceu Costa. Active video game exercise training improves the clinical control of asthma in children: Randomized controlled trial. *PLoS ONE*, 10(8):1–11, 2015. ISSN 19326203. doi: 10.1371/journal.pone.0135433.

[69] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[70] Michael James Heron. Cultural Integration and the Accessibility of Gaming. *The Computer Games Journal*, 5(3-4):91–94, 2016. ISSN 2052-773X. doi: 10.1007/s40869-016-0028-x.

[71] Marion A Hersh and Barbara Leporini. An overview of accessibility and usability of educational games. *Student usability in educational software and games: Improving experiences*, pages 1–40, 2013.

[72] Kimberly Hieftje, Tyra Pendergrass, Tassos Kyriakides, Walter Gilliam, and Lynn Fiellin. An Evaluation of an Educational Video Game on Mathematics Achievement in First Grade Students. *Technologies*, 5(2):30, 2017. doi: 10.3390/technologies5020030.

[73] Smoke J. Audio games and the never ending cycle of mediocracy, Feb 2020. URL https://blog.blackscreengaming.com/smokes-thoughts/audio-games-and-the-never-ending-cycle-of-mediocracy/02/17/2020/.

[74] Lukas Dominik Kaczmarek, Michał Misiak, Maciej Behnke, Martyna Dziekan, and Przemysław Guzik. The Pikachu effect: Social and health gaming motivations lead to greater benefits of Pokémon GO use. *Computers in Human Behavior*, 75:356–363, 2017. ISSN 07475632. doi: 10.1016/j.chb.2017.05.031.

[75] Bill Kapralos. Learning about serious game design and development at the k-12 level. *International Journal of Information and Learning Technology*, 38: 316–327, 2021. ISSN 20564899. doi: 10.1108/IJILT-01-2021-0003.

[76] Bill Kapralos, Fuad Moussa, Karen Collins, and Adam Dubrowski. Instructional Techniques to Facilitate Learning and Motivation of Serious Games. *Instructional Techniques to Facilitate Learning and Motivation of Serious Games*, pages 79–101, 2017. doi: 10.1007/978-3-319-39298-1.

[77] David Kaufman. Videogames in education - comparing studentsâ, student teachersâ and master teachers opinions and experiences. pages 101–105, 2013. ISSN 2184-5026. doi: 10.5220/0004383701010105.

[78] Jeanine Krath, Linda SchÃŒrmann, and Harald F.O. von Korflesch. Revealing the theoretical basis of gamification: A systematic review and analysis of theory in research on gamification, serious games and game-based learning. *Computers in Human Behavior*, 125:106963, 2021. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2021.106963. URL https://www.sciencedirect.com/science/article/pii/S0747563221002867.

[79] Richard L. Lamb, Leonard Annetta, Jonah Firestone, and Elisabeth Etopio. A meta-analysis with examination of moderators of student cognition, affect, and learning outcomes while using serious educational games, serious games, and simulations. *Computers in Human Behavior*, 80:158–167, 2018. ISSN 07475632. doi: 10.1016/j.chb.2017.10.040. URL https://doi.org/10.1016/j.chb.2017.10.040.

[80] Darina Lynkova. 49 video game statistics [click the "start" button], Aug 2019. URL https://review42.com/video-game-statistics/.

[81] Erin M. McTigue and Per Henning Uppstad. Getting Serious About Serious Games: Best Practices for Computer Games in Reading Classrooms. *Reading Teacher*, 72(4):453–461, 2019. ISSN 00340561. doi: 10.1002/trtr.1737.

[82] Christian Moro, Zane Stromberga, and Allan Stirling. Virtualisation devices for student learning: Comparison between desktop-based (Oculus Rift) and mobile-based (Gear VR) virtual reality in medical and health science edu-

cation. *Australasian Journal of Educational Technology*, 33(6):1–10, 2017. ISSN 14495554. doi: 10.14742/ajet.3840.

[83] Tahiri Najoua and El Alami Mohamed. A new evaluation technique through serious games for children with ASD. *International Journal of Emerging Technologies in Learning*, 15(11):202–217, 2020. ISSN 18630383. doi: 10.3991/IJET.V15I11.12843.

[84] Steve Nebel, Sascha Schneider, and Günter Daniel Rey. Mining learning and crafting scientific experiments: A literature review on the use of Minecraft in education and research. *Educational Technology and Society*, 19(2):355–366, 2016. ISSN 14364522.

[85] Jannie Nørlev, Katrine Sondrup, Christina Derosche, Ole Hejlesen, and Stine Hangaard. Game Mechanisms in Serious Games That Teach Children with Type-1 Diabetes How to Self-Manage: A Systematic Scoping Review. *Journal of Diabetes Science and Technology*, 2021. ISSN 19322968. doi: 10.1177/19322968211018236.

[86] Tanner Olsen, Katelyn Procci, and Clint Bowers. Serious games usability testing: How to ensure proper usability, playability, and effectiveness. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice: First International Conference, DUXU 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part II 1*, pages 625–634. Springer, 2011.

[87] Aleksandra E. Olszewski and Traci A. Wolbrink. Serious gaming in medical education: A proposed structured framework for game development. *Simulation in Healthcare*, 12:240–253, 8 2017. ISSN 1559713X. doi: 10.1097/SIH.0000000000000212.

[88] George P. Papanastasiou, Athanasios S. Drigas, and Charalabos Skianis. Serious games in preschool and primary education: Benefits and impacts on

curriculum course syllabus. *International Journal of Emerging Technologies in Learning*, 12(1):44–56, 2017. ISSN 18630383. doi: 10.3991/ijet.v12i01.6065.

[89] Nick Preston, Andrew Weightman, Justin Gallagher, Martin Levesley, Mark Mon-Williams, Mike Clarke, and Rory J. O'Connor. A pilot single-blind multicentre randomized controlled trial to evaluate the potential benefits of computer-assisted arm rehabilitation gaming technology on the arm function of children with spastic cerebral palsy. *Clinical Rehabilitation*, 30(10): 1004–1015, 2016. ISSN 14770873. doi: 10.1177/0269215515604699.

[90] Daniel M. Rodriguez, Maree Teesson, and Nicola C. Newton. A systematic review of computerised serious educational games about alcohol and other drugs for adolescents. *Drug and Alcohol Review*, 33(2):129–135, 2014. ISSN 09595236. doi: 10.1111/dar.12102.

[91] Magnus Sabel, Anette Sjölund, Jürgen Broeren, Daniel Arvidsson, Jean Michel Saury, Klas Blomgren, Birgitta Lannering, and Ingrid Emanuelson. Active video gaming improves body coordination in survivors of childhood brain tumours. *Disability and Rehabilitation*, 38(21):2073–2084, 2016. ISSN 14645165. doi: 10.3109/09638288.2015.1116619.

[92] J. Sauro. A practical guide to the system usability scale: Background, benchmarks best practices. *Independent Publishing Platform*, 2011.

[93] Ken Schwaber and Jeff Sutherland. The scrum guide. 2020.

[94] Robert Shewaga, Alvaro Uribe-Quevedo, Bill Kapralos, Kenneth Lee, and Fahad Alam. A serious game for anesthesia-based crisis resource management training. *Computers in Entertainment*, 16, 2018. ISSN 15443981. doi: 10.1145/3180660.

[95] Rebekah Shultz Colby and Richard Colby. A Pedagogy of Play: Integrating Computer Games into the Writing Classroom. *Computers and Composition*, 25 (3):300–312, 2008. ISSN 87554615. doi: 10.1016/j.compcom.2008.04.005.

[96] Brian A. Smith and Shree K. Nayar. The RAD: Making racing games equivalently accessible to people who are blind. *Conference on Human Factors in Computing Systems - Proceedings*, 2018-April:1–12, 2018. doi: 10.1145/3173574.3174090.

[97] Pamela C. Smith and Bernita K. Hamilton. The effects of virtual reality simulation as a teaching strategy for skills preparation in nursing students. *Clinical Simulation in Nursing*, 11(1):52–58, 2015. ISSN 18761399. doi: 10.1016/j.ecns.2014.10.001. URL http://dx.doi.org/10.1016/j.ecns.2014.10.001.

[98] S. Sreejesh, M. R. Anusree, and Abhilash Ponnam. Does game rules work as a game changer? Analyzing the effect of rule orientation on brand attention and memory in advergames. *Computers in Human Behavior*, 81:325–339, 2018. ISSN 07475632. doi: 10.1016/j.chb.2017.12.034.

[99] Marios Stanitsas, Konstantinos Kirytopoulos, Elise Vareilles, and (K Kirytopoulos. Facilitating sustainability transition through serious games: A systematic literature review.

[100] Viktoria Stray, Nils Brede Moe, Sintef Dag, and I K SjÃžberg. Daily stand-up meetings start breaking the rules preprint 1.

[101] Yoon Hi Sung and Wei Na Lee. Doing good while playing: The impact of prosocial advergames on consumer response. *Computers in Human Behavior*, 106(November 2019):106244, 2020. ISSN 07475632. doi: 10.1016/j.chb.2020.106244. URL https://doi.org/10.1016/j.chb.2020.106244.

[102] Isabelle Dela Torre and Imran Khaliq. A study on accessibility in games for the visually impaired. Institute of Electrical and Electronics Engineers Inc., 6 2019. ISBN 9781728124049. doi: 10.1109/GEM.2019.8811534.

[103] Michael Urbanek, Peter Fikar, and Florian Güldenpfennig. About the sound of bananas - Anti rules for audio game design. *2018 IEEE 6th International*

Conference on Serious Games and Applications for Health, SeGAH 2018, pages 1–7, 2018. doi: 10.1109/SeGAH.2018.8401361.

[104] Michael Urbanek, Florian Güldenpfennig, and Michael Habiger. Creating audio games online with a browser-based editor. *ACM International Conference Proceeding Series*, pages 272–276, 2019. doi: 10.1145/3356590.3356636.

[105] Zeph M.C. van Berlo, Eva A. van Reijmersdal, and Esther Rozendaal. Adolescents and handheld advertising: The roles of brand familiarity and smartphone attachment in the processing of mobile advergames. *Journal of Consumer Behaviour*, 19(5):438–449, 2020. ISSN 14791838. doi: 10.1002/cb.1822.

[106] Devika Vashisht and Sreejesh S. Pillai. Are you able to recall the brand? The impact of brand prominence, game involvement and persuasion knowledge in online â advergames. *Journal of Product and Brand Management*, 26(4): 402–414, 2017. ISSN 10610421. doi: 10.1108/JPBM-02-2015-0811.

[107] R. A. Virzi. Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors*, 1992.

[108] Will Wade and David Porter. Sitting playfully: Does the use of a centre of gravity computer game controller influence the sitting ability of young people with cerebral palsy? *Disability and Rehabilitation: Assistive Technology*, 7(2):122–129, 2012. ISSN 17483107. doi: 10.3109/17483107.2011.589485.

[109] Thomas Westin, Jaeun Jemma Ku, Jérôme Dupire, and Ian Hamilton. *and WCAG 2 . 0 â A Gap Analysis*, volume 4. Springer International Publishing. ISBN 9783319942773. doi: 10.1007/978-3-319-94277-3. URL http://dx.doi.org/10.1007/978-3-319-94277-3{_}43.

[110] Kyle Wilcocks, Bill Kapralos, Alvaro Uribe Quevedo, Fahad Alam, and Adam Dubrowski. The anesthesia crisis scenario builder for authoring anesthesia crisis-based simulations. *IEEE Transactions on Games*, 12:361–366, 12 2020. ISSN 24751510. doi: 10.1109/TG.2020.3003315.

[111] Pieter Wouters, Christof van Nimwegen, Herre van Oostendorp, and Erik D. van Der Spek. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, 105(2):249–265, 2013. ISSN 00220663. doi: 10.1037/a0031311.

[112] Stelios Xinogalos and Margarita Maria Tryfou. Using Greenfoot as a tool for serious games programming education and development. *International Journal of Serious Games*, 8(2):67–86, 2021. ISSN 23848766. doi: 10.17083/ijsg.v8i2.425.

[113] Atlanta Xp and Mike Cohn. User stories applied for agile software development. 2004.

[114] Yu Zhonggen. A meta-analysis of use of serious games in education over a decade, 2019. ISSN 16877055.

[115] L. Zoccolillo, D. Morelli, F. Cincotti, L. Muzzioli, T. Gobbetti, S. Paolucci, and M. Iosa. Video-game based therapy performed by children with cerebral palsy: A cross-over randomized controlled trial and a cross-sectional quantitative measure of physical activity. *European Journal of Physical and Rehabilitation Medicine*, 51(6):669–676, 2015. ISSN 19739095.