Memorial University
Faculty of Engineering and Applied Science

Implementation Plan for Waggle 2.0

Online Swarm Robotics Simulator

Design Project I
ECE 7010
Dr. Dennis Peters

Prepared by G8
Emily Wiseman 201628518
Christian James 201630571
Nadia Shalaby 201108248
Abigayle Hickey 201732641

Submitted on
6 August 2021

Executive Summary

Waggle is a swarm robotics application created by Dr. Andrew Vardy that allows users to familiarize themselves with the concept of swarm robotics through a visual simulation. Dr. Vardy proposed implementing an updated version of Waggle, and the capstone team of Abigayle Hickey, Christian James, Emily Wiseman, and Nadia Shalaby accepted the proposal. The proposed Waggle 2.0 will involve an improved simulation window, a tutorial for new users to follow, challenges for evaluating users, a broader range of robot simulations and level environments, and various features to aid in the education of students by teachers using the program.

Various meetings were held between the project team and their supervisor and client, Dr. Vardy, to further develop the Waggle 2.0 requirements. The team divided the overall criteria for project success into three general categories: the requirements for maintaining the current standards and features of Waggle, the requirements for the proposed additional features, and the possible extra element to be implemented if time permits. Overall, the key design features will involve the web application frontend and backend architecture, user login and authentication using auth0, and the simulation features for students and teachers using Waggle 2.0.

Project management will utilize various tools and methods to help efficiently track and dictate the course of Waggle 2.0 software development. The team will use issue tracking, Scrum, peer code review, Atlassian's Jira, and Atlassian's Confluence for their collaboration. Following that, the group split the project work into three epics: a minimum viable product, Waggle 2.0, and Waggle 2.1 with its extended features over Waggle 2.0.

In general, the team's potential concerns involve collaboration with external resources, and more importantly, having the time required to implement the various feature proposed in the Waggle 2.0 scope. Budget is not a primary concern as this is a web application, and the group has found ways to alleviate or remove most costs. Through the development team being composed of all students and Waggle being an educational tool, most of the possible expenses with developing a web application have been taken care of due to those factors.

Table of Contents

List of Acronyms & Abbreviations

MUN Memorial University of Newfoundland

UI User Interface

UML Unified modeling language

UX User Experience

List of Figures

List of Tables

1 Introduction

Waggle is a swarm robotics application that allows users to familiarize themselves with the mechanics of swarm robotics through visual simulation aid. Waggle features simulations for different swarm robotic AI and is designed to be user-friendly to allow a soft learning curve while understanding the basics of programming these mechanics onto robots [1].

Waggle 1.2: The Online Swarm Robotics Lab

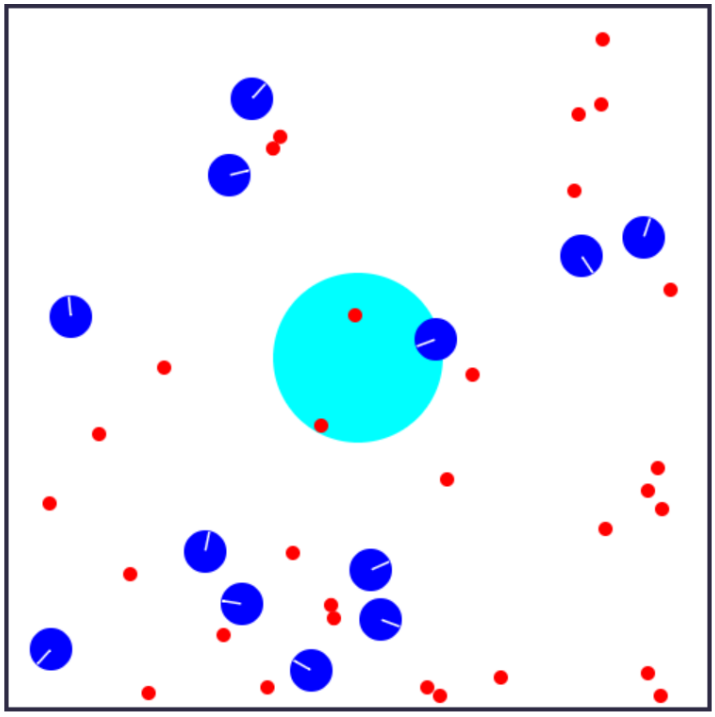

Figure 1: Example Waggle Simulation of swarm robots sorting pucks

Dr. Andrew Vardy created Waggle as a tool for teaching the basics of swarm robotics. Waggle can currently demonstrate five types of swarm robotic simulations, but to do so effectively is dependent on a teacher guide to explain how to use the application and help when students are stuck. As shown in Figure 1 above, an example swarm robotics "problem" would be for the robots (blue) to collaboratively gather the pucks (red) efficiently, without external communication [1].

The current project includes updating Waggle to increase independent user accessibility and add the ability to expand the types of swarm robotic problem simulation users can create. It is desirable for Waggle to

become an effective learning tool for swarm robotics, so additional features to achieve this are the main objective. The following report will discuss the group's implementation plan for Waggle. This plan will include the tentative design, potential limitations and challenges and the management and success criteria to achieve the final design. Throughout the term, the planned implementation of Waggle was decided upon through collaboration between the team and client, Dr. Vardy.

2 Project Description and Background

The Waggle product was created and designed by the client and project supervisor, Dr. Andrew Vardy. Dr. Vardy proposed implementing an updated version of Waggle that included easier accessibility, a broader range of simulations and higher learning functionality. With this extended scope for the previous Waggle, Waggle 2.0 could begin development. The assigned team consists of four computer engineering students, each with specialized backgrounds that will tackle various parts of Waggle, such as development, styling, and backend criteria for webpage development.

Typically, team meetings were held bi-weekly, with a significant focus on scope construction and design queries for Waggle 2.0.

Table 1: Project Team

Member	**Background**
Andrew Vardy	Project Supervisor and Waggle Client
Abigayle Hickey	Frontend Development
Christian James	Software Testing and Debugging
Nadia Shalaby	Backend Development
Emily Wiseman	Full-stack Development

2.1 Problem Definition

Waggle is a largely teacher-dependent learning tool that functions best with demonstrative assistance to use and simulate swarm robotics. Waggle comes with a PowerPoint slide deck that can act as an independent tutorial for users; however, a more dynamic tool is desirable to achieve independent learning better and ensure comprehension of the simulations.

Waggle currently has five simulation “levels” that feature various swarm robotic problems but has no option for users to create levels with different swarm robotics mechanics. It is desired for Waggle to be able to explore a broader spectrum of swarm robotics. Waggle also lacks additional features that can heighten usability to help learn swarm robotics in university equivalent school systems.

Waggle's current limitations include the accessibility and learning curve of using the simulations. Waggle's goal is an increased understanding of swarm robotics; thus, the learning curve for the

programming language and software used should be minimal. Waggle lacks robot or background customization features, which enhance usability and broaden the spectrum of simulation purposes. The design of Waggle is for both beginner and advanced usage; thus, Waggle allows for both simple drag and drop and programming-based customizations.

To increase the learning tool functionality of Waggle, challenges and student-teacher aspects will be added. Challenges to test the user's understanding, providing a score and solution to swarm robotics problems. Students will have individual logins that will display their scores and save the simulations they create. Teachers will have a separate login that will display the statistics of their students. The statistics page will show which students have completed which challenges and the scores they received on them.

Teachers will also have the option of creating levels or challenges, which they can display to their students. These levels and challenges will serve as an effective evaluation tool for teachers to analyze their students' comprehension of the subject.

Finally, for a practical, real-world demonstration, Waggle will be able to export a file that can determine the behavior of actual swarm robots. In this way, students will simulate their robots and then physically see the outcome of those behaviors they created in a real-world setting.

2.2 Market Review

Various learning tool simulators vary on their intended audience, such as simulators that encompass easy-to-use tools for beginners and advanced simulators that would require training to utilize. To make the correct design decisions, the team researched comparable simulation applications. Four different current technologies were compared with Waggle's design to aid in constructing the updated Waggle's scope.

NetLogo is an abstract, multipurpose simulator that functions primarily as a learning tool for students and teachers. Waggle will aim for a slightly more realistic simulator comparatively, with more features for user customization. Waggle will use the same basics for visually changing its simulation using sliders and inputs to appease the beginner side of the tool. In addition to this visual programming, NetLogo appeals to advanced users with the ability to create simulations entirely using code. NetLogo created its own programming language for its simulations, but Waggle will differ by using a more real-world applicable programming language. Therefore, Waggle will be similar to NetLogo by supplying a visual programming language for beginners and an advanced programming language for practical, real-world use [2].

Open Roberta is a primarily children-focused simulator that uses the visual programming approach, such as drag and drop elements, and a visual programming language called Blockly. Open Roberta is an excellent comparator for Waggle's beginner-oriented visual programming aspect, as users can customize robot obstacles and watch the outcome [3].

Gazebo is a robot simulator that focuses on highly realistic simulations. It uses a robust physics engine that creates a significant learning curve for its utilization. This software targets professionals aiming to test and simulate their robots realistically, not someone that is seeking to learn the basics [4]. V-Rep is another simulation tool like Gazebo, in that it also aims for realism and thus has a similar learning curve for use [5].

With this research conducted, Waggle has a constructed concept, with identified similarities and differences between similar learning tool simulators. Both NetLogo and Open Roberta are appropriate tools to compare with Waggle and create a realized starting point for some of the features Waggle will include.

3 Project Scope

Though Waggle has an original functional version, it was clear that the redesign will be easier to perform from scratch. The following are the features we will be implementing in order from highest to lowest priority.

- Simulation Window
    - Blank simulation window
    - Pre-clustering
    - Clustering
    - Sorting
    - Fireflies
    - Pheromones

  The simulation window for each type will include:

    - Predefined starting code (for all simulators except the blank window)
    - Programming panel
    - Robot and background customization
    - Simulation changes using toggles and slides
    - Import/ export button of code

- Tutorial on creating simulations
- About Page (Information on swarm robotics and Waggle purpose)
- Challenges and grading features
- User page and saved simulations
- Teacher page with statistics
- Teacher level/challenge designer
- Export of simulation code to real robots

The current Waggle application features the complete Simulation Window aspect of the design, except the robot and background customization feature and the broader, blank simulation window. Though a complete redesign will take place, the backbone of this window is readily available. Its completion, regarding styling, user accessibility and bug handling, is the top priority and considered required for the

application to have full functionality. The remaining features aid in the learning aspect of swarm robotics and are deemed as additional functionalities, time permitting.

4 Project Success Criteria

Through discussions with Dr. Vardy, requirements for this project were determined as listed below. The implementation and success of these elements will dictate the success of the project overall. For convenience and organization, the requirements have been separated into three categories:

1. Maintenance of Current Standards and Waggle 1.0 Features
2. Proposed Features
3. Possible Requirements, Time Permitting

4.1 Requirements for Maintenance of Current Standards and Waggle 1.0 Features

- Waggle 2.0 will implement the main functionality contained in Waggle 1.0:
    - Swarm Robots visual simulation area
    - Robots represented by on-screen coloured circles with outer sensors
    - The ability for robots to move (animation required)
    - The tool will include the swarm robotics algorithms:
        - Pre-clustering
        - Clustering
        - Sorting
        - Fireflies
        - Pheromones
    - Ability to code the robot simulation with Python or a visual programming language
    - Ability to export code into either a pdf or a .py file
- Waggle 2.0 will not remove any of the existing mentioned features above.
- Waggle 2.0 will allow changes to the number of sensors on each robot.
- Waggle 2.0 will contain sensors that are pentagon-shaped.
- Waggle 2.0 will use React.js framework.
- Waggle 2.0 will be written in TypeScript, JavaScript, HTML, CSS, SCSS, and follow a JavaScript Standard Style Guide when writing the code.
- Waggle 2.0 shall minimize the use of external dependencies unless justified.
- Waggle 2.0 shall be modular to increase reusability and readability.
- All design decisions shall be documented.

4.2 Requirements for Proposed Features

- Waggle 2.0 shall use auth0 to allow users to log in with Google and GitHub, thus maintaining a high level of user security guaranteed by Google and GitHub.
- Users shall be able to save simulations and code for a returning session.
- Waggle 2.0 will have a tool to drag and drop robots onto the robot simulator canvas component.
- Waggle 2.0 will have a Canvas area for the simulation of the robots.
- The tool shall contain a levels dropdown menu for changing the difficulty of the coding challenge.
- The tool shall contain a robot/bug editor.
- The tool shall contain an about page.
- The tool shall contain an undo and reset button.
- Waggle 2.0 will include an overview of swarm robotics and a brief introduction to Dr. Vardy's research on swarm robotics.
- Waggle 2.0 will have challenges for the users to complete, and each time the user completes a challenge, the next level will increase in difficulty, helping the user build their knowledge on swarm robotics.
- Waggle 2.0 will contain a leaderboard for students to submit their current level for a particular challenge and see other users' scores.
- Waggle 2.0 will allow each user to submit their code for review or grading to the teacher/ professor.

4.3 Possible Requirements, Time Permitting

- Waggle 2.0 will integrate with Group number 7 (swarm robotics) and program their robots by the code the user writes.
    - Waggle 2.0 will have the ability to export a JSON file containing parameters that the user selects in Waggle 2.0.
    - The user will transfer this JSON file to Group 7's physical robots.
    - Group 7 will integrate this JSON through control software to program their robots.
- Waggle 2.0 will be a tool used for teaching students swarm robotics in a classroom setting.

5 Detailed Technical Design

5.1 Architecture

The architecture for the Waggle 2.0 project can be broken into two sections, the application's frontend infrastructure and backend infrastructure. The application's frontend is the primary system for the project and deals with the logic behind the swarm robotics simulator, and the user interface for Waggle. Waggle's frontend is a React.js application that will:

- wrap the swarm robotics physics engine designed by Dr. Vardy
- provide modular configuration files that will allow the physics engine to perform different swarm behaviours
- provide a user interface
- handle the compilation of Python code or code from the visual programming language that will feed into the swarm physics engine

The backend of the Waggle project will handle global data such as user authentication, teacher and student login roles, and Waggle's grading system. Figure 2 illustrates the simplified system architecture for Waggle 2.0.

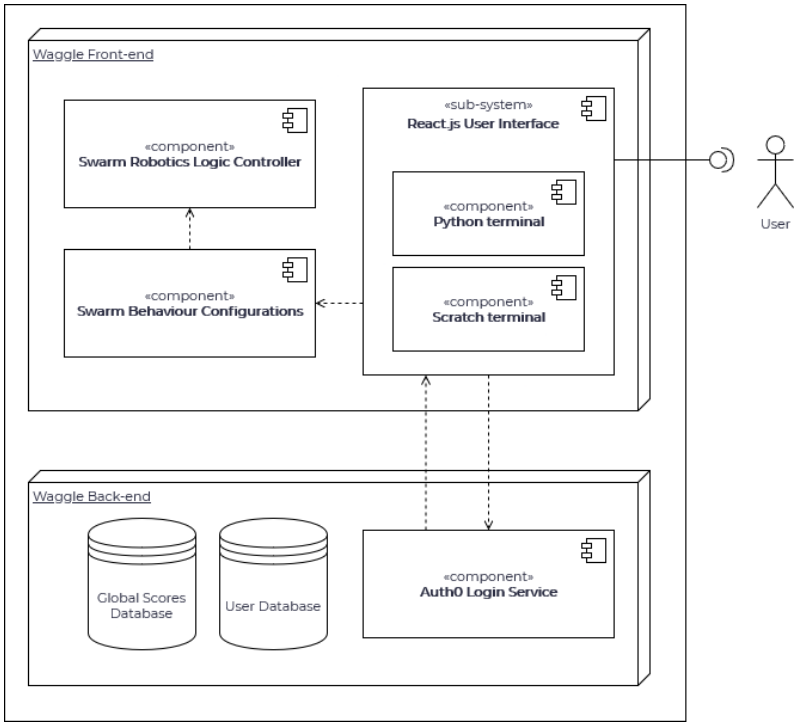

Figure 2: UML component diagram for Waggle 2.0's Architecture

5.2 Authentication

The team will build the authentication system will using the auth0/auth0-react library. This library provides authentication and authorization with any identity provider running on any stack on any device or cloud. Auth0 allows us to guarantee security by giving the option to sign in with GitHub or Google and by using security best practices. Not storing passwords on Waggle lowers potential security breaches by adversaries gaining access to the app. This authentication method also ensures the user's privacy and protects administrators from gaining unneeded information from the user. Within Memorial, all students use Google to access things such as Gmail, so we can be sure that signing in with Google will be preferred and familiar. Since our audience is computer-based students, we will also allow users to log in using GitHub. The auth0 library contains the auth0-react SDK that integrates with React.js providing a high-level API.

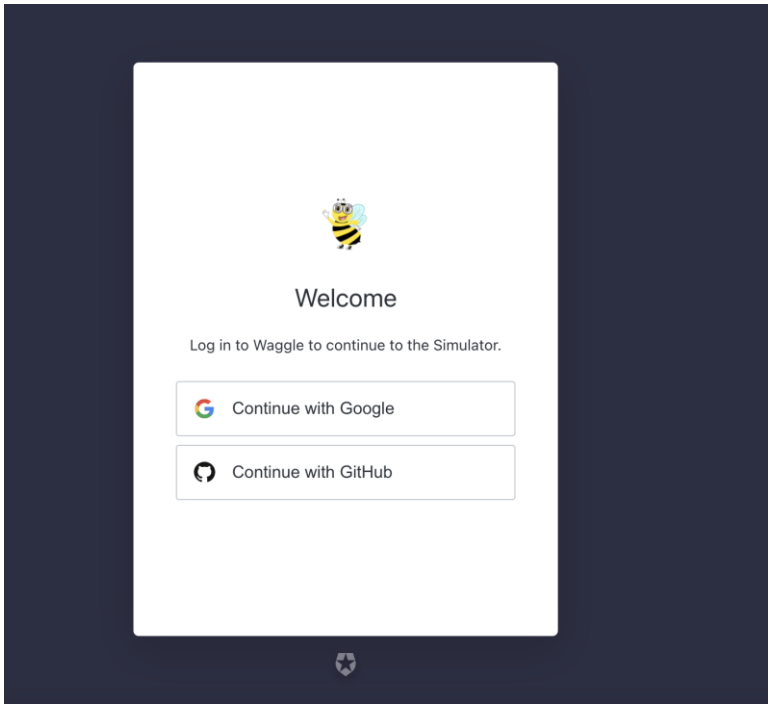

Figure 3: Waggle login using auth0

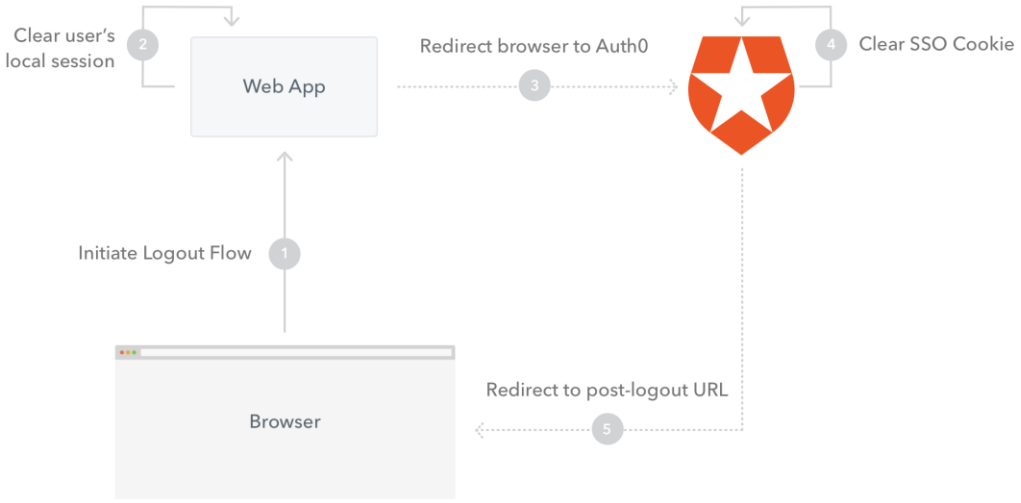

Figure 4: Auth0 Application Implementation

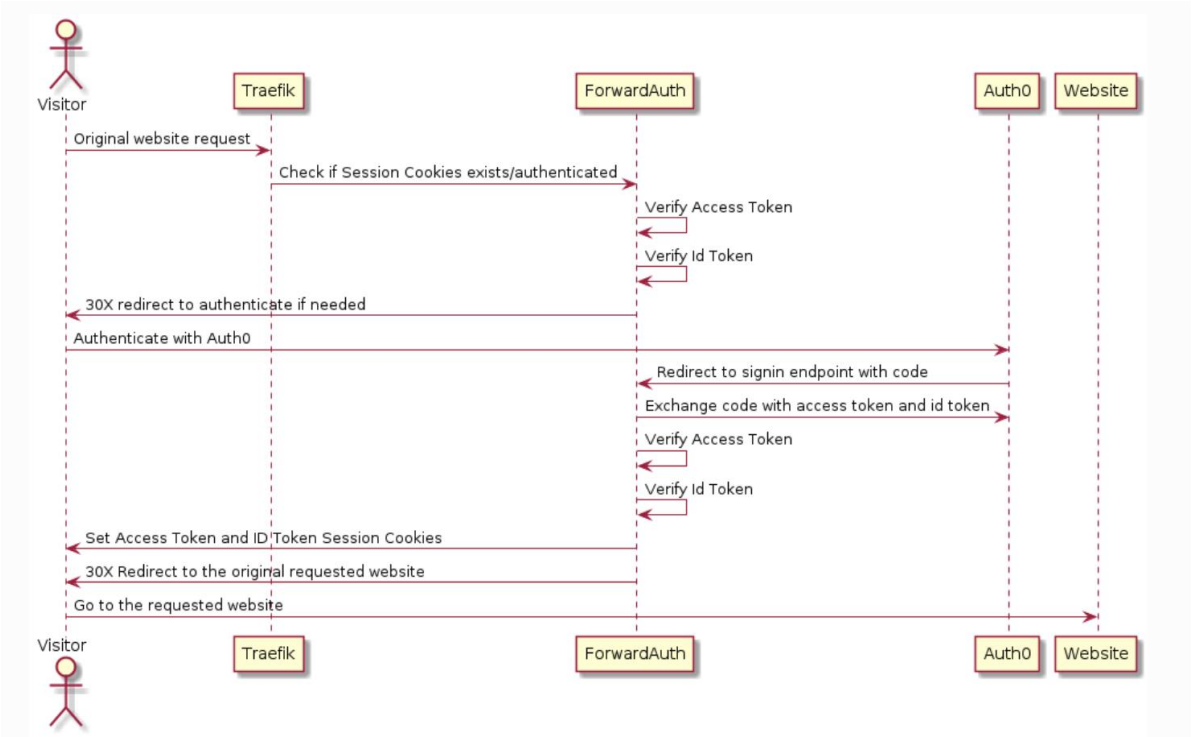

Figure 5: UML sequence diagram of auth0 authentication/authorization

5.3 Simulation Workflow

Waggle will have nested navigation, with various tabs for information about Waggle, simulations, challenges, and users' pages. The essential criteria from this are the simulation windows, which students will use to observe the swarm robotic behaviour and complete different “challenges” using swarm robotics concepts. Below is a sequence diagram of a student using a simulator and a teacher creating a new swarm robotic level.

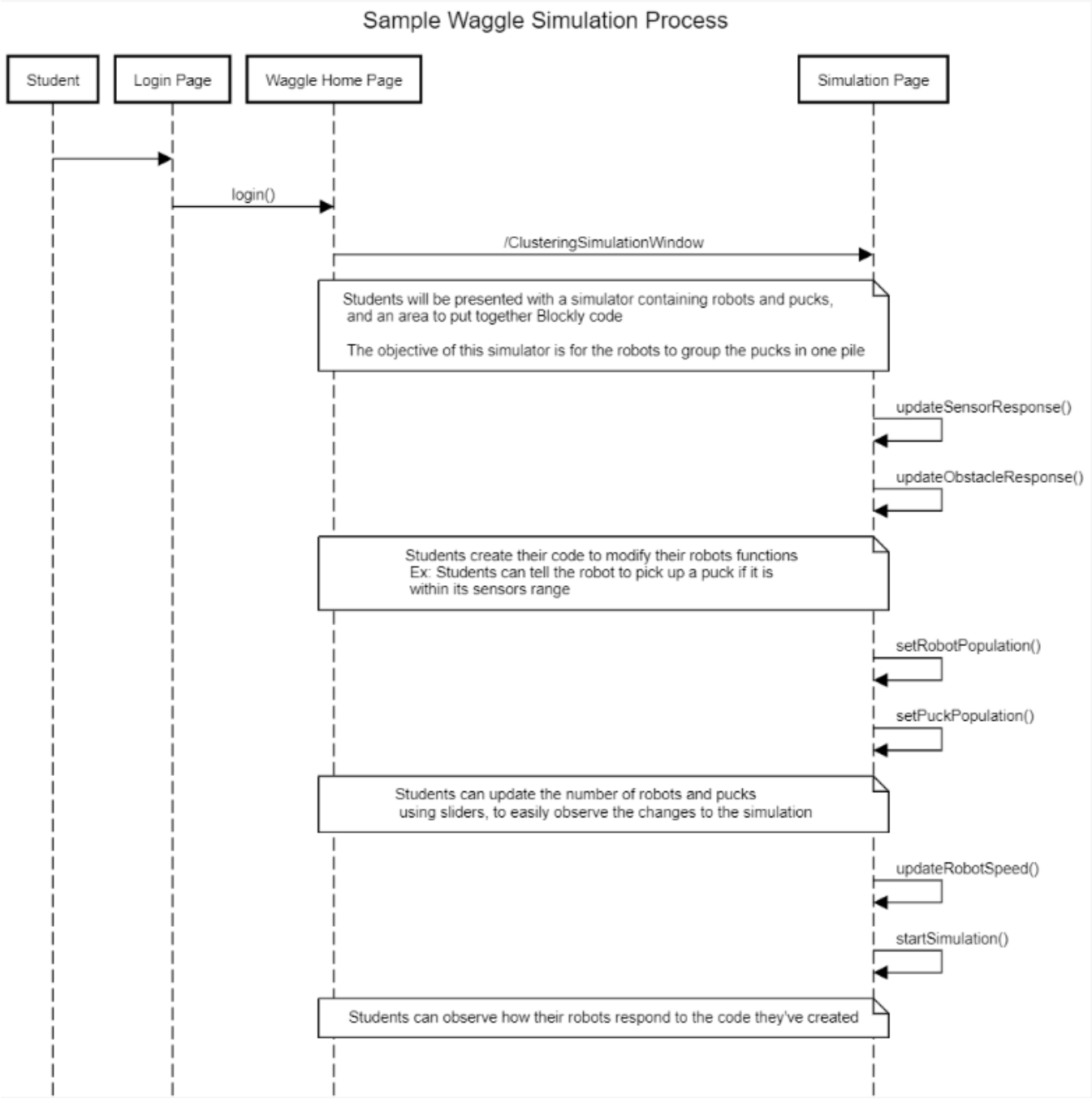

Figure 6: A sample sequence diagram showing students modifying a simulator to complete a swarm robotics challenge

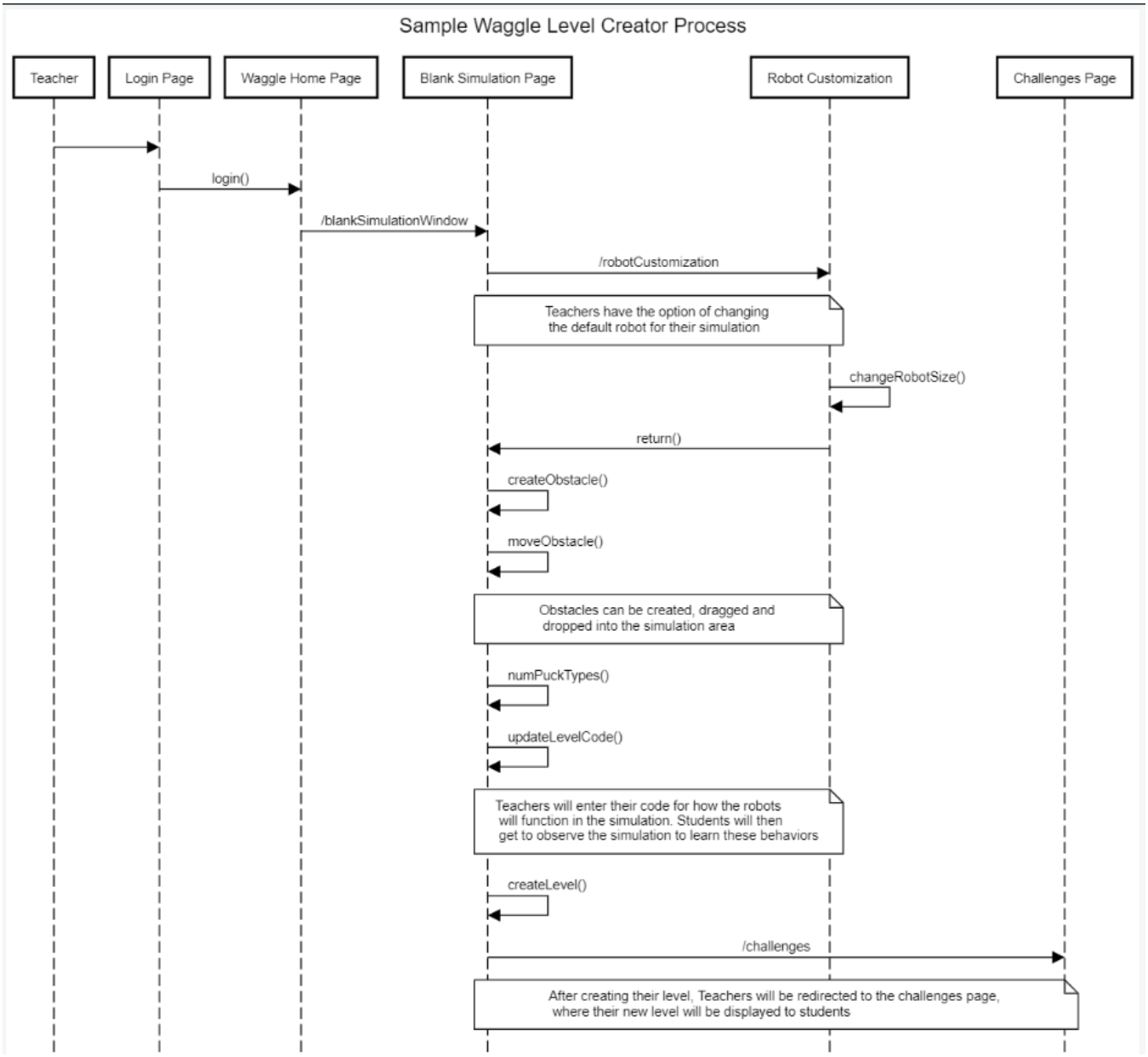

Figure 7: Sample sequence diagram displaying how a teacher can create a new swarm robotics level/challenge for students to use

6 Results, Evaluation, Validation

The team will compare the results of this project against the expected results in Table 2 below. Once the project is complete, Waggle should meet the following criteria: there shall be a working tool that provides the ability to simulate swarm robots, change the number of sensors on the robots, change the number of faces on each sensor, contain different challenges, and proceed through various levels while using the simulation environment and the coding area.

Table 2: Expected functioning outcomes of Waggle

The tool shall allow users to log in using Google and GitHub.
The tool shall allow users to simulate swarm robots.
• The tool will allow users to drag and drop robots onto the simulation environment canvas.
• The tool will have a pentagon-shaped sensor instead of a circle (in waggle 1.0).
The tool will include challenges and levels for the user to advance through them.
The tool will allow users to code the robots in Python.
The tool will allow the users to export the written code locally into a .py file.
The tool shall have a leaderboard
The tool shall include a levels dropdown menu, changing the difficulty of the coding challenge.
The tool shall contain Undo and Reset Buttons.
The tool shall contain a Demo/Tutorial.
The tool shall contain an about page.

The tool shall allow students to submit code to be graded or viewed by the teacher/professor.

7 Project Management Strategy

To date, the Waggle 2.0 project has been focused on planning the requirements for the project and determining its viability. The team has finalized the scope of Waggle 2.0 through six client meetings with Dr. Vardy. These meetings have been tracked using Atlassian's Confluence. The team has also completed market research, written a management plan, and filmed a pitch video for the project. These milestones, along with planned development goals, are illustrated in Figure 8.

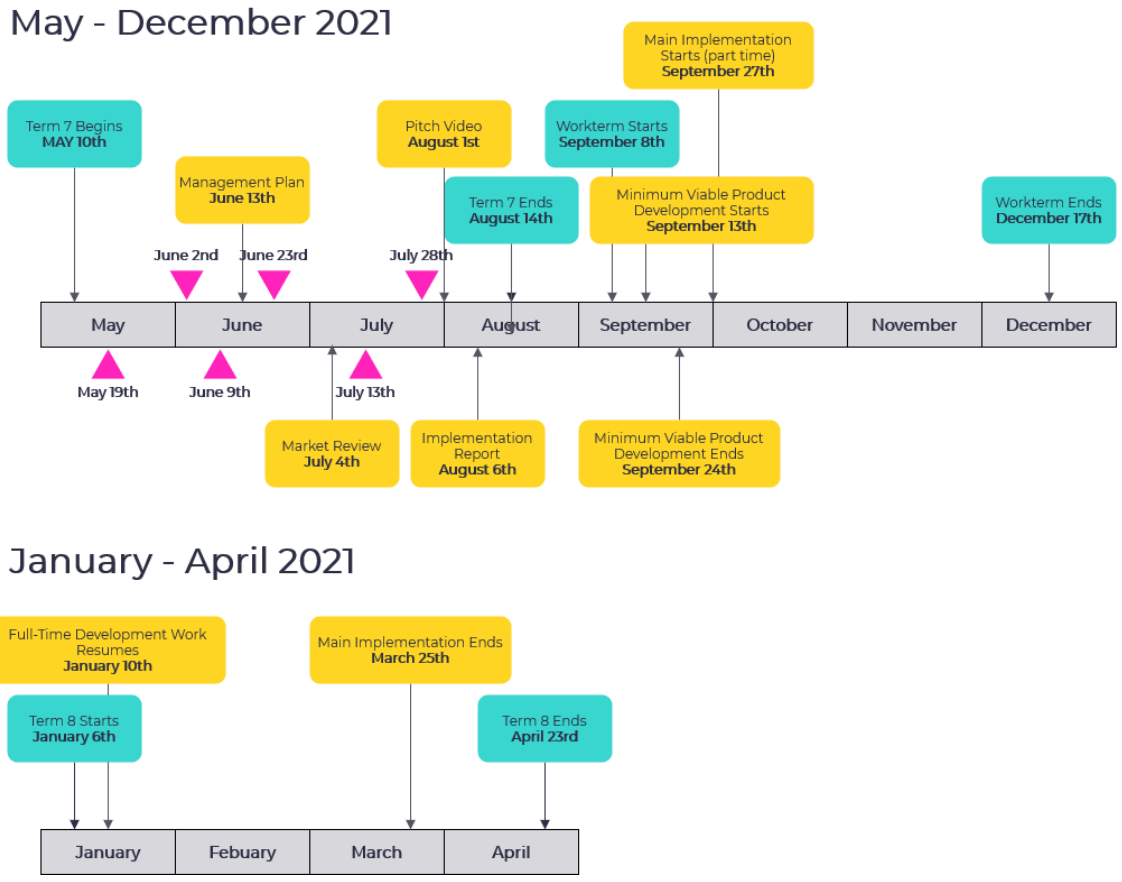

Figure 8: Timeline of project milestones

Waggle 2.0 software development will begin in Fall 2021 and use issue tracking, Scrum, and peer code review to manage the project. Issue tracking for Waggle 2.0 will be done using Atlassian's Jira platform and is how the team will divide and plan work. Before development, features for the site have been identified and categorized into three primary epics, as discussed in Table 3. Each developer will track their development time in a Jira issue when working on a feature.

To implement these epics, the project will use the Scrum approach to software management. Scrum works by dividing features into adaptive work cycles known as sprints. During each sprint, roughly every two

weeks, our team will focus on completing a predefined set of features. At the end of the sprint, the team will:

- Discuss what went well in a team retrospective and adjust work per developer if needed
- Plan for the next sprint and manage the project's issue backlog
- Adjust scope as necessary

The project uses Atlassian's Bitbucket for its source code management and will develop each feature as a separate branch corresponding to its Jira issue ID. Before merging a branch to main, the branch will undergo a code review. Peer code reviews improve code quality and ensure that the team understands all parts of the project.

Table 3: Waggle 2.0 Implementation Epics and preliminary development timeline

Epic	Description	Duration Estimate
Minimum Viable Product	In business, a minimum viable product describes a version of a product with just enough features to be usable by early customers. In the context of Waggle 2.0, this will be a basic implementation of all features included Waggle 1.2. The main work involved in this epic will be modularizing the existing Waggle 1.2 codebase.	Two weeks
Waggle 2.0 Implementation	The features that extended the current Waggle implementation into version 2.0. These features can be considered the scope of the capstone.	Sixteen weeks (full-time work)
Waggle 2.1 Extended Features	Additional features not considered pertinent to the scope of the project but considered "nice to have." The team will only implement these features if time permits.	TBD

8 Project Budget

Waggle 2.0 is a web application, and as such, it will incur the typical costs of any other website, including hosting, design, and development [6]. Professionally hosting a website is an ongoing cost that is not made possible with the project's budget. To eliminate the cost of hosting and allow easy maintenance by the client, Waggle 2.0 will be hosted through MUN ITS Computing Services.

The new version of Waggle 2.0 will feature user sign-in and customization. Implementing this will require a backend to support user data and a secure login method. The team will develop the website backed in-house using a free framework such as Django or Express.js. User authentication will be handled using Auth0, a secure authentication platform popular in the industry, substantially reducing security risks and development time. Auth0 is free to use for sites with less than 7,000 users [7].

The goal for Waggle 2.0 is to support users of different backgrounds and require little to no learning curve. A simple, easy-to-use user interface for Waggle will be critical to useability and supporting product adoption [8]. For the site's design, the team will be adapting a free dashboard user interface (UI) kit by Zvonimir Juranko [9]. UI kits are style guides with prebuilt components for developing mock-ups (a necessary visual requirements document) of websites. Sample mock-ups of the Waggle 2.0 site are shown in section Appendix A: Waggle 2.0 Sample UI Mock-ups.

Secondary external resources, including a domain name and logo to support user adoption, will be subject to the budget. A temporary logo for Waggle was developed by Wiseman and can be found in section Appendix B: Waggle Logo and Colour Palette. All costs associated with the project may be reduced due to educational services or provided free of charge.

Table 4: Bill of Materials

#	Name	Description	Value	Actual Cost
1	Bitbucket	Atlassian's source code management software. Free for teams under ten people. Valuation is based on a full-size start-up [1].	$300/month[2]	$0.00
2	Jira	Atlassian's issue tracking software, used by the team for management development. Free for teams under ten people. Valuation is based on a full-size start-up	$700/month[3]	$0.00
3	Confluence	Atlassian owned Wiki software. Used by the team to share documents and track meetings. Free for teams under ten people. Valuation is based on a full-size start-up [1].	$500/month[4]	$0.00
4	Visual Studio Code	A free-to-use integrated development environment created by Microsoft.	$0.00	$0.00
5	Web Hosting	Web hosting will be through MUN's ITS Computing Services. Valuation of web hosting is based on the cost of one year of site hosting through Digital Ocean with GitHub's Education Pack's $100 student credit discount applied [10].	$10.00/month	$0.00
6	Auth0	Authentication Service that handles login security and allows Waggle users to login using GitHub or Google credentials. Auth0 is free for up to 7,000 users [7].	$23/month[5]	$0.00
7	Domain name	The domain name cost is based on *waggle.org* and *waggl.ca* (waggle.ca is taken) from Google Domains [11].	$17.00/year	$17.00/year

Table 5: Research and Development budget

#	Name	Description	Value	Actual Cost
1	UI/UX Kit	The project will use a free dashboard UI kit by Zvonimir Juranko [9]. The valuation of a UI kit is based on the price of Dashboard UI Kit 3.0 (a React-specific UI kit), popular among react web developers [5].	$71.27	$0.00
2	UI/UX Mock-ups	The UI mock-ups, which illustrate Waggle's design and act as visual requirements, were created by Wiseman. To hire a UI Designer to create the mock-ups would cost $25 to $40 hourly [12]	$25-$50/hour	$0.00
3	Logo	Waggle's logo was created by Wiseman and can be viewed in Appendix B: Waggle Logo and Colour Palette. The valuation of getting a logo designed for the project is based on the cheapest, professional logo design available on Fiverr (a popular website for freelance services) [13].	$986.32	$0.00

[1] A start-up is defined as any company with less than 100 employees [18]
[2] Bitbucket's standard plan cloud charges $3 per user [21].
[3] Jira's standard plan cloud charges $7 per user [19].
[4] Confluence's standard cloud charges $5 per user [20]
[5] After 7,000 users Auth0 services start at $23 per month [7].

9 Risks and Concerns

The Waggle 2.0 implementation carries along with it a variety of risks and concerns that may become more apparent as development and management continue. The potential challenges visible at this stage of the process can be broken down into three primary categories: the required expertise, the time available for the project, and the project's partial dependence on Auth0.

9.1 Required Expertise

Due to the scope provided for the Waggle 2.0 implementation, the project as a whole will span over many areas of expertise to implement the best quality product. Waggle will require in-depth knowledge of frontend development for the majority of its primary features and a strong understanding of software backends for the Waggle backend the team will need to create.

To best accommodate the programming expertise required, the team assigned each member a primary role based on their skills developed throughout taken course material and their completed work terms. Still, despite that role assignment, there are areas of knowledge that the team will still need to learn or address to complete this project fully.

One area of expertise that has already come up through discussion that the group lacks is UI and UX design. Some ideas have arisen to address this concern potentially. Still, they would require a time commitment to learn some ideal design habits or collaboration with a design expert to ensure that the team's design seems to meet appropriate standards.

Along with the general expertise requirements for developing a web application, the entire project spans the concept of swarm intelligence and swarm robotics that the team will need to understand adequately. The client, Dr. Andrew Vardy, will address some questions or concerns around swarm robotics that the group may have. However, the project will still require the development team to understand swarm robotics to appropriately design the simulation behaviours, levels, and robot behaviour to fit the educational purpose of Waggle. Also, if the group starts exporting details for physical robot behaviour, it will require collaboration with the other swarm robotics group to properly work with their physical robots.

9.2 Time Available

Although frequent discussion has been made between the development group and the client to establish the project criteria, there is still the possibility that the group may not be able to meet the requirements on time. The actual complexity of the project will further be realized when development begins, and the team implements the various required elements. Along with that, the scope could potentially expand through team or client requirements that may change priority or arise throughout the development process. In these situations, the time available for the project may be constrained, which will further constrain the criteria the team can complete.

9.3 Auth0 Dependence

The login feature that Waggle will implement is utilizing the auth0 service to enable users to log into Waggle through their Google or GitHub accounts. This dependence on the auth0 service could change or become undesirable due to factors outside the team's control. In those cases, that would cause the group to re-evaluate their method of creating a login system for Waggle. Then, if another suitable service cannot replace auth0, the group would be required to develop their own implementation of a login system, which would involve a large amount of extra work, time, and security concerns to address.

References

[1] A. Vardy, "Waggle 1.2: The Online Swarm Robotics Lab," [Online]. Available: http://www.cs.mun.ca/~av/waggle/.

[2] U. Wilensky, "NetLogo," 2016. [Online]. Available: https://ccl.northwestern.edu/netlogo/.

[3] B. Jost, R. Budde, T. Leimback and A. Kapusta, "Open Roberta," 2016. [Online]. Available: https://lab.open-roberta.org/.

[4] A. Howard and N. Koenig, "Gazebo," 2019. [Online]. Available: http://gazebosim.org/.

[5] "V-REP," [Online]. Available: https://www.coppeliarobotics.com/.

[6] G. Stevens, "A Detailed Breakdown of Website Development Costs," HostingCanada, 19 May 2021. [Online]. Available: https://hostingcanada.org/website-development-costs/. [Accessed July 2021].

[7] auth0, "Flexible pricing for companies and developers," Okta, [Online]. Available: https://auth0.com/pricing. [Accessed July 2021].

[8] usability.gov, "User Interface Design Basics," usability.gov, [Online]. Available: https://www.usability.gov/what-and-why/user-interface-design.html. [Accessed June 2021].

[9] Z. Juranko, "XdDash UI Kit - Finance & Crypto," Behance, 5 December 2020. [Online]. Available: https://www.behance.net/gallery/109067173/XdDash-UI-Kit-Finance-Crypto. [Accessed July 2021].

[10] GitHub Education, "GitHub Student Developer Pack," GitHub Inc., [Online]. Available: https://education.github.com/. [Accessed June 2021].

[11] Google, "Google Domains," [Online]. Available: https://domains.google.com/registrar/. [Accessed June 2021].

[12] UpWork, "How much does it cost to hire a UI designer?," Upwork® Global Inc. , [Online]. Available: https://www.upwork.com/hire/ui-designers/cost/. [Accessed June 2021].

[13] Fiverr, "Professional Logo Services," Fiverr International Ltd., [Online]. Available: https://www.fiverr.com/search/gigs?query=logo&source=toggle_filters&search_in=category&search-autocomplete-original-term=logo&sub_category=49&ref=file_format%3Aai%2Cpng%2Csvg%7Cpackage_includes%3Ainclude_source_file%2Clogo_transparency%2Cprint_resolution%. [Accessed June 2021].

[14] "Color Hunt," [Online]. Available: https://colorhunt.co/palette/2c2e43595260b2b1b9ffd523. [Accessed June 2021].

[15] D. Huff, How to Lie with Statistics, New York: Norton, 1954.

[16] C. F. Belkacem Khaldi, "An Overview of Swarm Robotics: Swarm Intelligence Applied to Multi-robotics," *International Journal of Computer Applications*, vol. 126, pp. 31-37, 2015.

[17] M. a. U. M. a. S. M. a. E. W. Schranz, "Swarm Robotic Behaviors and Current Applications," *Frontiers in Robotics and AI*, vol. 7, p. 36, 2020.

[18] E. McGowan, "What Is a Startup Company, Anyway?," startups.com, 1 March 2018. [Online]. Available: https://www.startups.com/library/expert-advice/what-is-a-startup-company. [Accessed July 2021].

[19] Jira Software, "Plans and pricing," Atlassian, [Online]. Available: https://www.atlassian.com/software/jira/pricing. [Accessed July 2021].

[20] Confluence, "Cloud Plans and Pricing," Atlassian, [Online]. Available: https://www.atlassian.com/software/confluence/pricing. [Accessed July 2021].

[21] Bitbucket, "Cloud Plans and pricing," Atlassian, [Online]. Available: https://www.atlassian.com/software/bitbucket/pricing?tab=cloud-tab. [Accessed July 2021].

Appendix A: Waggle 2.0 Sample UI Mock-ups

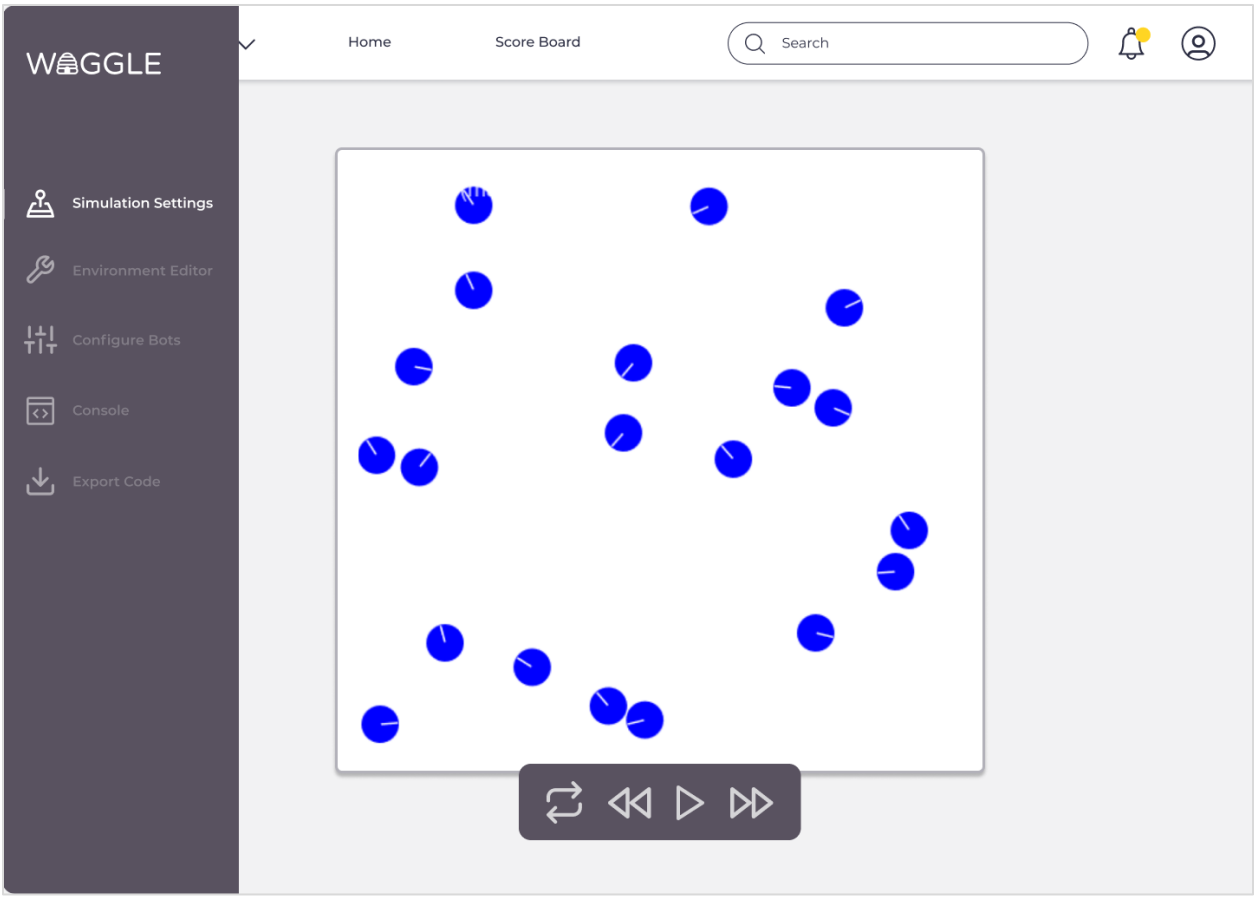

Figure 93: Waggle UI Mock-up illustrating main swarm simulation interface

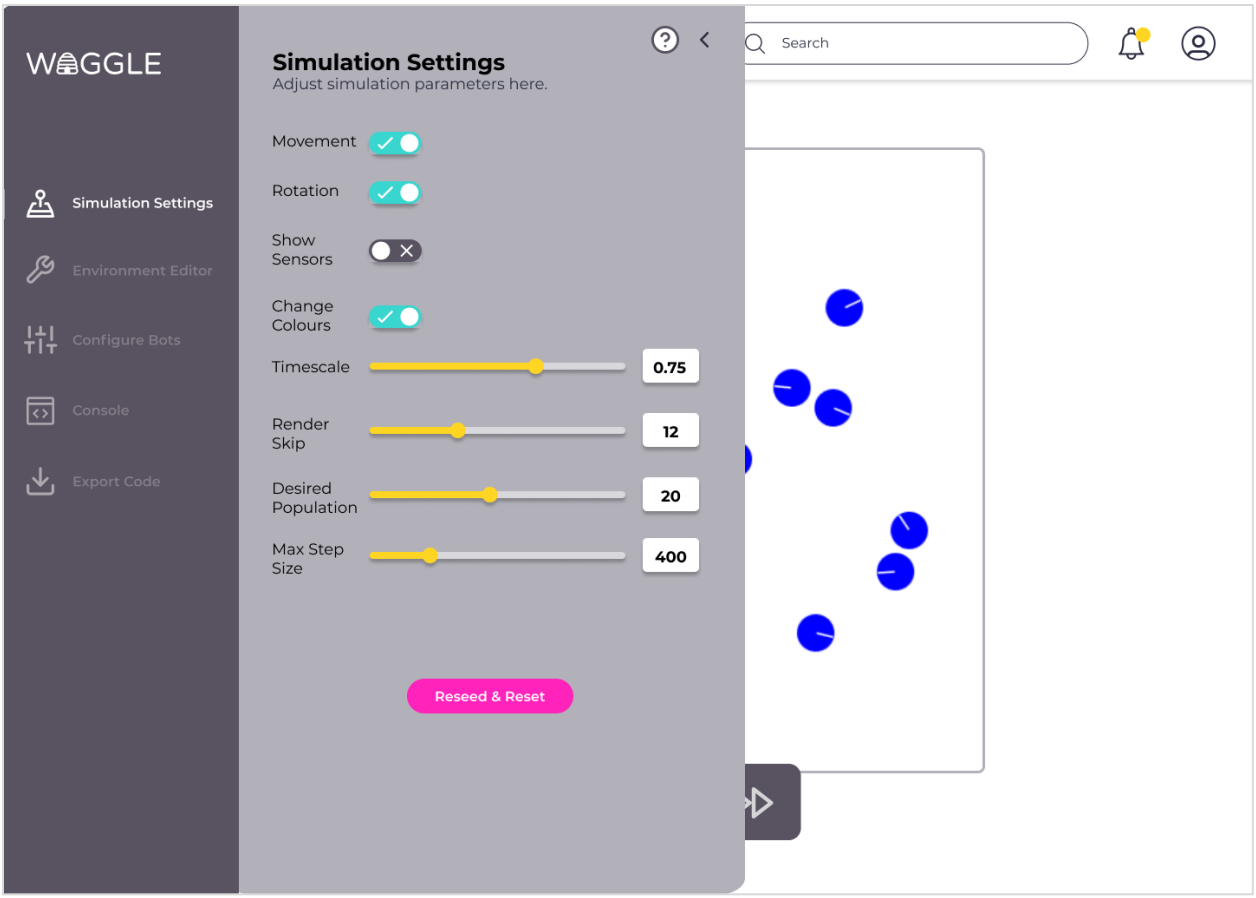

Figure 104: Waggle UI Mock-up showing available swarm simulation settings

Appendix B: Waggle Logo and Colour Palette

Figure 5: Logo for Waggle 2.0

Figure 116: Primary colour palette for Waggle 2.0 courtesy of Color Hunt [14]