Software Requirements Specification (SRS)
Angle Anglers

Team: 8
Authors: Christopher Coco, Evan Casasanta, Vraj Patel, Sebastian Gyger
Customer: Students in the 5-7th Grade
Instructor: Dr. James Daly

1 Introduction

This document contains the Software Requirement Specification (SRS) for Angle Anglers, an educational fishing video game. This SRS will describe the software's functionality and purpose.

1.1 Purpose

This SRS document's purpose is to describe the system's requirements and constraints. It will also outline the functionality of the software. It will contain a list of requirements for the software, diagrams that outline different components of the game, and describe a prototype of the game. This document will help developers have a solid vision and blueprint for what the project is trying to accomplish. It will also allow clients to learn more about what the project has to offer and the potential benefits of the game.

1.2 Scope

Angle Anglers is an edutainment game designed to improve student engagement and incentivize learning in the classroom by helping students learn about angle concepts in a fun and engaging way. Our intuitive design and robust features make learning about angles effortless. Students struggle with mathematics because of a lack of repetition in solving math problems. Our game teaches students about angles through an interactive fishing minigame that improves the students' skills through repetition. It will help them recognize patterns in the way to solve the problems which can help them improve their critical thinking skills.

1.3 Definitions, acronyms, and abbreviations

Angle Anglers - The title of the game this document describes.

Angle - A space between two lines at the point that they meet.

Degree - A unit of measurement used to define angles. A full circle is 360 degrees.

Acute Angle - An angle that is more than 0 degrees but less than 90 degrees.

Obtuse Angle - An angle that is more than 90 degrees but less than 180 degrees.

Right Angle - An angle that is exactly 90 degrees.

Supplementary Angle - Defines the angle that could be added to the current angle to make it equal to 180 degrees.

Identify Angles - Determining if an angle is acute, obtuse, or right.

Player - The person playing the game.

Movement Speed - How fast the player can move around the overworld.

Walk - The base movement speed the player moves around at.

Run - A way to let the player move around the map at twice their usual movement speed.

Hub World - A place for the player to walk around in and interact with various aspects.

Overworld - Another word for Hub World.

Fish - An in-game creature.

Species - A specific type of fish.

Minigame - An in-game activity which the player can complete to acquire fish.

Fishing - Fishing starts a minigame that gives the player a chance to catch a fish.

Currency - The equivalent of money that the player will earn and spend.

Bait - Purchasable in-game items that increases the odds that the caught fish is a previously uncaught species.

Fish Length - An attribute each fish has, where a longer fish earns the player more currency.

Rod - Purchasable in-game items that allow the player to fish.

Hook - An object that attempts to dig into a fish so it cannot escape.

Line - The string connecting the hook to the rod.

Shop - An in-game building where the player can purchase rods and bait using currency.

Fishing Spot - In-game locations where the player can cast their rods to fish.

Pier - In-game locations at which the fishing spots are located.

Body of Water - In-game locations containing piers.

Lake - One of the in-game bodies of water.

Ocean - One of the in-game bodies of water.

Fishing Academy - An in-game building that teaches the player the rules of the game.

Encounter Table - A graphic showing the distribution of the rarities of fish.

Science Lab - An in-game building that lets players view the encounter tables for each area.

Compendium - An in-game tool that tracks what fish the player has caught.

Flavor Text - In-game text that describes an object.

1.4 Organization

1.4.1 Organizational Structure of the SRS

The rest of the document is structured in a way to provide detailed information in a logical format. The subsequent section provides an overall description of the project, including product perspective, user characteristics, constraints, assumptions, and dependencies. Following this, the project requirements are defined to describe what the customer wants from the project. Afterward, all the diagrams (use case, class, sequence) are outlined. After that the instructions to run the prototype and some sample scenarios are included. The document concludes with references to the sources and the point of contact person regarding the document.

1.4.2 Rest of the SRS Content

The remaining sections of the SRS would provide detailed specifications for each aspect of the application, including:

- Interfaces
- Design Constraints
- Dependencies
- Requirements
- Diagrams
- Prototype
- Additional Specifications

2 Overall Description

Our game Angle Anglers is designed as an educational game involving various kinds of angles, with the goal of helping students expand their angular knowledge. It is a two-dimensional semi open world game that involves fishing.

This section will explain the context of the project and describe its interfaces. It will also explain the product's context and functions. The user characteristics will also be described to know what the ideal users of this product will be. The limitations or restrictions that impact the project are also described. Also, the assumptions and dependencies, and the apportioning of requirements are included here.

2.1 Product Perspective

Angle Anglers is an educational fishing game that is a standalone application with the goal of educating middle schoolers about angles and their properties. This game requires no integration with any larger system; however, it can be used in a classroom setting where each student is playing on their own personal computer or a school lab's computers. The game does not interact with any external sources and can be run on most basic computers without an issue.

Students playing our game will learn about angles and whether they are obtuse, acute, or right. They also can increase their mental math ability as they must find supplementary angles of the angle given. The game will help students get prepared for homework assignments and even the test that their teacher will eventually give them on angles.

2.2 User Characteristics

The user is expected to be a middle school student ages 10 to 13 between the years of 5th and 7th grade who is enrolled in a general mathematics class. Their skill level in dealing with angles is expected to be low when first playing the game, but after a while of playing their skill level will be expected to go up exponentially. Users also should have a basic understanding of gaming (knows arrow keys are used to move around) so they can easily figure out the map and what to do when they play the game for the first time.

2.3 Constraints

The math topics explored in this game are of the 7th grade level of the Massachusetts Mathematics Curriculum Framework, limiting the game to basic concepts about angles.

The game should be easy to install and utilize as it should be usable and playable by teachers and students with minimal computer experience. As such there should be clear and concise instructions and an easy to navigate interface.

Since the game is meant to be used for education it is imperative to strike a balance between fun and learning, actual teaching and studying should not be forsaken to

make the game more fun, while it should be entertaining enough to keep middle-school aged children engaged.

The program must be run on a computer using the Windows operating system, with the Windows version either 10 or newer to guarantee faultless playability. The program relies on input from a keyboard and mouse (or trackpad) and is not tested with any other input devices.

2.4 Assumptions and Dependencies

Some hardware assumptions of the user include that the user has a functional and modern computer along with a keyboard and mouse (or trackpad) connected to their computer. The only software requirement is that the computer that the user is using has a modern operating system, for example Windows 10 or 11. Basic English reading comprehension is required to play the game.

2.5 Apportioning of Requirements

There are some means of user customization: you can purchase new fishing rods and new bait. A few things that have been discussed that are beyond the scope of our current project are online features allowing players to play on the same server. This will allow the game to have a social aspect to it which will increase interaction between students if played in the classroom which has the potential to have them be more engaged in the content. Another feature would be multiple towns and map areas that the player can visit to allow for the game to also have exploration elements to increase engagement. These areas would include different towns and potential traveling to places on boats.

3 Specific Requirements

1. The game shall require a computer to be played that meets certain minimum requirements.
    1.1. The computer must be able to receive keyboard input.
    1.2. The computer must be able to receive input from a mouse or trackpad.
    1.3. The computer must have a screen to output images onto.
2. The game shall have a hub world.
    2.1. The player shall be able to walk around in and explore the hub world.
        2.1.1. The player shall have an option to double their movement speed by running.
    2.2. The player shall have access to various buildings aiding them during gameplay.
        2.2.1. The fishing academy shall help the player learn the angle concepts needed to play the game as well as explain how the game works.
        2.2.2. The shop shall allow players to acquire new rods.
        2.2.3. The science lab shall show players the encounter tables for each fishing area to help show players where they may be missing fish.
    2.3. There shall be areas in the overworld where the player can fish.
        2.3.1. There shall be a lake with a pier.
        2.3.2. There shall be an ocean with a pier.
3. The game shall have a minigame that involves fishing.
    3.1. This minigame shall be started by casting a fishing rod in the overworld.
        3.1.1. Upgrading the fishing rod shall make the problems more difficult but increase the odds of catching certain fish.
    3.2. The game shall generate a fish whose species is determined based on the area's encounter table, the player's rod, and any bait used.
        3.2.1. Audio and visual cues shall prompt the player to react to start the minigame.
    3.3. The player shall time a hit in a 180-degree semicircle to determine the angle used for the problems.
        3.3.1. The first problem requires the player to identify whether an acute, obtuse, or right angle was formed.
        3.3.2. The second problem requires the player to calculate the supplementary angle for the line.
    3.4. After successfully answering the two questions in the minigame, the fish shall be caught.
        3.4.1. The fish's length shall be longer the faster the player completed the minigame.

3.5. The fish shall escape if the student does not answer within a given time.

4. There are multiple areas to fish in with each area having its own encounter table.
5. Players shall have access to a compendium that shall record their progress on collecting all the fish present in the game.
    5.1.1. The compendium shall be divided by area and record the longest length of that fish, and the number caught.
    5.1.2. In the compendium each fish shall have a small amount of flavor text that provides the player with some trivia about the species.
    5.1.3. Each area shall have a small amount of flavor text that provides the player with some trivia about the area's ecosystem.
6. Currency shall be able to buy different rods and bait.
    6.1. Players shall gain currency by catching fish.
7. The game shall have entertaining music in the background.
    7.1. The game shall have an option to turn off the background music.
8. The game shall provide tools to aid the player in understanding how to play the game.
    8.1. The game shall provide "how-to-play" instructions when the player initially plays the game.
    8.2. The game shall provide help to the user when stuck by providing the correct answer to the question.
9. The game shall have an engaging user interface (UI).
    9.1. The game shall have an inviting and bright visual style.
        9.1.1. The game shall have large, easy-to-understand prompts and menus.
    9.2. The game shall be a top-down point of view.
10. The game shall contain audio and visual cues to aid the player.
    10.1. The visual and audio cues shall play in sync to avoid confusion when playing without sound.

4 Modeling Requirements

The following section contains all the diagrams for the project. These diagrams include a use case diagram, a class diagram, and two sequence diagrams. Each diagram will have supporting descriptions and content to help better describe it.

4.1 Use Case Diagram

The following diagram contains use cases for Angle Anglers. Every use case has a label in the center of the circle and the lines that connect them represent how each of the use cases interact with each other. The primary functions that the player can perform are “Fish at Spot,” “View Science Lab,” "Buy Item", and “View Academy.” These are the core functionalities of the game. Fish at Spot will allow the user to fish, which is the main gameplay loop of our game. Fishing will teach different concepts regarding angles and will reward the player with currency that they can use to buy new items. “View Science Lab” and “View Academy” both provide the player with information. The academy provides information about how to play the game and the underlying math concepts needed to play the game. The science lab provides information about where you can find each fish. Buy Item lets the player use the currency that they have earned to buy new items like fishing rods which will alter the difficulty of the game.

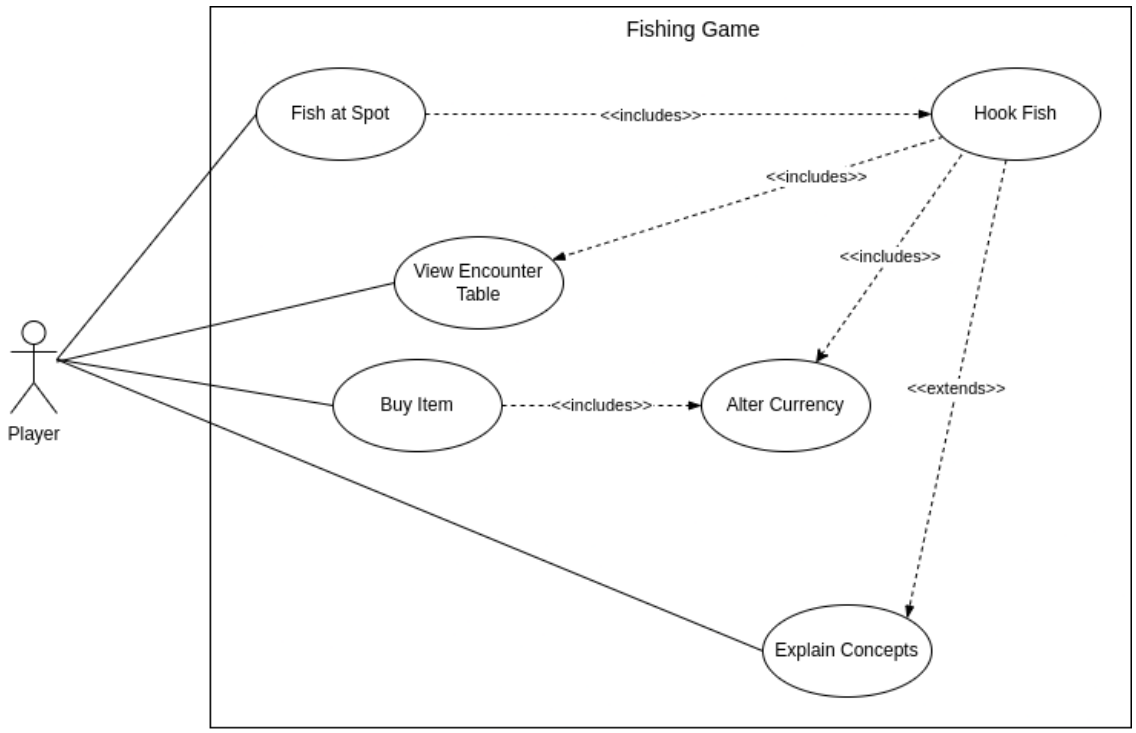

Figure 1 - Use Case Diagram

Use Case Name:	Fish At Spot
Actors:	Player
Description	The player selects one of the fishing spots in the overworld to fish at.
Type:	Primary
Includes:	Hook Fish
Extends:	None
Cross-refs:	Requirement 5
Uses cases:	Hook Fish

Use Case Name:	Hook Fish
Actors:	Player
Description	After selecting a spot to fish in, the user will wait for a fish to bite and once one does, they will attempt to catch it. If they are successful, they will receive rewards and if they fail their answers will be corrected. If they get any wrong answers, they will be correct, and the right answer will be displayed.
Type:	Primary
Includes:	Increase Currency, View Encounter Table
Extends:	Explain Concepts
Cross-refs:	Requirement 4
Uses cases:	Increase Currency, View Encounter Table, Explain Concepts

Use Case Name:	View Encounter Table
Actors:	System

Description	Retrieves and displays information about the fish appearance rates for the area.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirement 5.1
Uses cases:	None

Use Case Name:	Alter Currency
Actors:	System
Description	Alters the player's currency on different events. Operations include increasing and decreasing the player's currency.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirement 6
Uses cases:	None

Use Case Name:	Explain Concepts
Actors:	System
Description	Explain how to play the game, its features, and the underlying math concepts to the game.
Type:	Secondary
Includes:	None
Extends:	None
Cross-refs:	Requirement 9
Uses cases:	None

Use Case Name:	Buy Item
Actors:	Player

Description	The player purchases different items from the shop which will aid the player and give them unique effects or alter the difficulty.
Type:	Primary
Includes:	Spend Currency
Extends:	None
Cross-refs:	Requirement 7
Uses cases:	Spend Currency

4.2 Class Diagram

The following diagram contains information about the classes that make up Angle Anglers. Their names, fields, and methods are all outlined. All the different classes that make up the game revolve around the player and their interactions with them. The player itself holds information about the player like profile name and currency and has the actions of casting their fishing rod and reeling in a fish. One of the things that the player can interact with is fishing spots. These spots are where fish can be caught. Fishing spots all have an associated encounter table which holds information about the probabilities of finding each fish in that area. Encounter tables are affected by a difficulty which is determined by the player's fishing rod. This difficulty will affect the numbers generated for a problem along with increasing the probability of a rarer fish. The next thing that the player can interact with is the Store. The Store handles all purchases that the player may make and holds items that the player can buy. These items are from a common interface which holds basic information about that item type. The two item types that inherit from this interface are the Rod and Bait classes. The Rod class contains an effect which is a mapping to a difficulty. The bait class contains the type which is the rarity that it will affect and a reroll amount that it offers. The final thing that the player can interact with is the Compendium. This is used to keep track of game progress that the player makes like statistics of their gameplay and the number of fish that they have caught. The Compendium holds information about a Fish Species which is a class that represents each kind of fish and holds various information about them.

4.2.1 Class Diagram

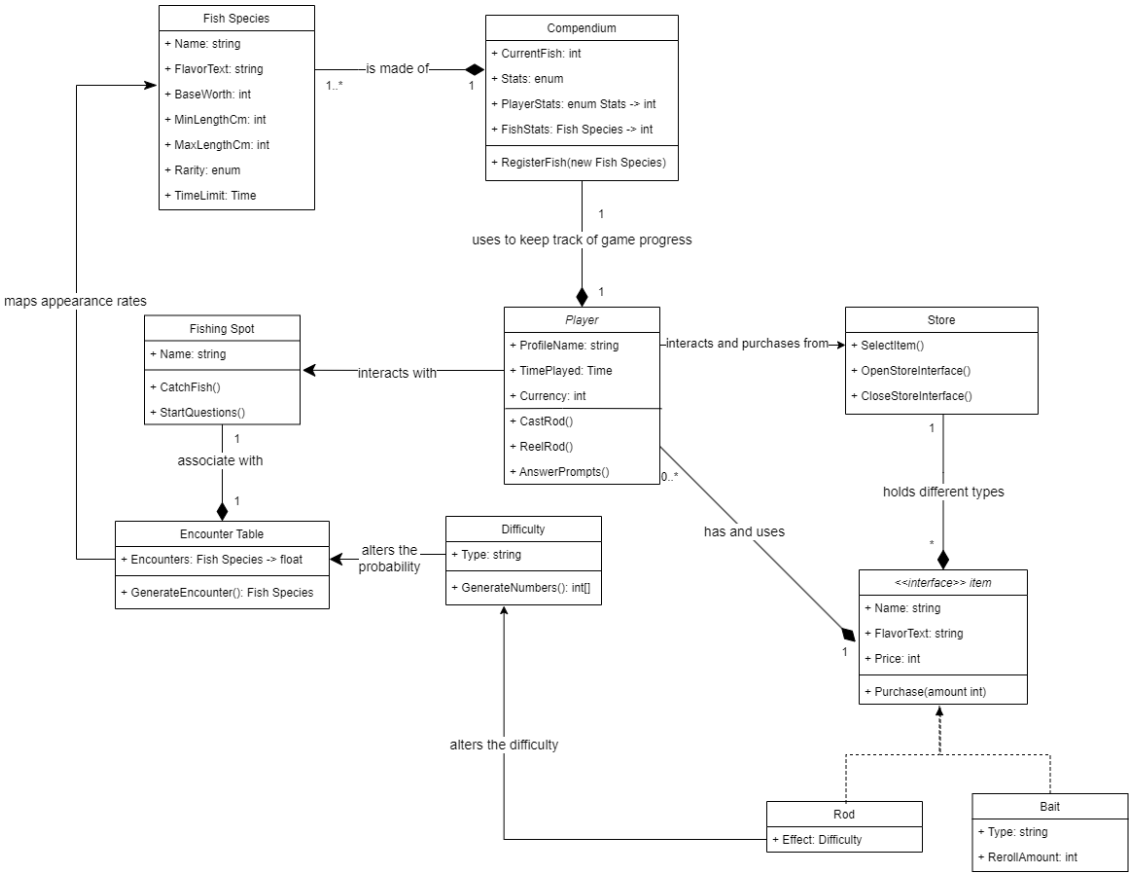

Figure 2 - Class Diagram

4.2.2 Class Descriptions

Class Name	Player		
Description	Represents the player and their actions.		
Extends	None		
Attributes	ProfileName	string	The player's profile name that they assign to themselves.
	TimePlayed	Time	The amount of time that the player has played the game for.
	Currency	int	The amount of currency that the player has.
Operations	CastRod()	Casts the players fishing rod in the area that they are standing.	
	ReelRod()	Reels in the players fishing rod.	
	AnswerPrompts()	Allows the player to submit their answers while fishing.	
Relationships	Interacts and purchases from the Store.	The player can interact with the store to buy items with the currency that is allotted.	
	Has and uses Items.	The player can use the items that they own to perform various actions and provide modifiers.	
	Interacts with Fishing Spot.	The player can interact with the different fishing spots to fish.	

Class Name	Fishing Spot		
Description	Represents a spot where the player can fish.		
Extends	None		
Attributes	Name	string	The name of the fishing spot.
Operations	CatchFish()	Generates a fish for the player to catch.	
	StartQuestions()	Display and start the questions that the player needs to answer to catch a fish.	
Relationships	Associates itself with an Encounter Table.	Each fishing spot will have an encounter table.	

Class Name	Encounter Table		
Description	Represents the probabilities of the fish that can be caught in the fishing spot.		
Extends	None		
Attributes	Encounters	Map<Fish Species, Float>	Maps a Fish Species to a probability.
Operations	GenerateEncounter(): Fish Species	Generates a fish species based on the table.	
Relationships	Maps appearance rates of Fishing Species.	The main job of Encounter Tables is to map to what probability different fish appear at.	

Class Name	Difficulty		
Description	Represents the difficulty of the game.		
Extends	None		
Attributes	Type	string	The type of difficulty.
Operations	GenerateNumbers()	Generates numbers for the questions based on the type of difficulty.	
Relationships	Alters the probability of an Encounter Table.	The difficulty will alter the probabilities in the Encounter Table.	

Class Name	Store	
Description	Represents the store where the player can buy different items.	
Extends	None	
Operations	SelectItem()	Select an item to purchase from the store.
	OpenStoreInterface()	Open the store GUI that displays the information and items available at the store.
	CloseStoreInterface()	Close the store GUI.
Relationships	Holds different types of items.	The store holds different objects that inherit from the item interface.

Class Name	Item (interface)		
Description	An interface that acts as a basis for different item types.		
Extends	None		
Attributes	Name	string	The name of the item.
	FlavorText	string	A brief description of the item that is displayed to the player.
	Price	int	The amount of currency that the item costs.
Operations	Purchase(amount int)		Purchase an X amount of that item.

Class Name	Rod		
Description	Represents a fishing rod that the player can use. Different fishing rods have different effects on the difficulty.		
Extends	Item		
Attributes	Effect	Difficulty	Which Difficulty the fishing rod is linked to
Relationships	Alters the difficulty		The fishing rod that is currently equipped determines the difficulty that is used for generating numbers for the problems.

Class Name	Bait		
Description	Represents bait that the player can use and buy. Bait can be used to increase the rerolls on finding a certain rarity of fish.		
Extends	Item		
Attributes	Type	string	The rarity that the bait increases a chance for.
	RerollAmount	int	The amount of rerolls the bait provides.

Class Name	Compendium		
Description	Represents a collection of data about the player’s gameplay.		
Extends	None		
Attributes	CurrentFish	int	The total number of fish that the player has caught.
	Stats	enum	An enum that represents all the different statistics tied to the player.

	PlayerStats	Map<enum Stats, int>	A table of different statistics about the player. For example, total fish caught, total currency earned.
	FishStats	Map<Fish Species, int>	A table of different statistics about each species of fish. For example, the number of this type caught.
Operations	RegisterFish(Fish Species)	Registers a new fish to the compendium.	
Relationships	The compendium keeps track of game progress.		

Class Name	Fish Species		
Description	Represents a species of fish.		
Extends	None		
Attributes	Name	string	The name of the species.
	FlavorText	string	A short little description of the fish.
	BaseWorth	int	The base amount that the fish is worth.
	MinLengthCm	int	The minimum length in centimeters that the fish can be.
	MaxLengthCm	int	The maximum length in centimeters that the fish can be.
	Rarity	enum	The rarity of the fish that can be from

			Common, Uncommon, or Rare
	TimeLimit	Time	The amount of time before the fish gets away while attempting to catch it.
Relationships	Make up part of the Compendium.		Each Fish Species has an entry in the compendium which has details about the species and the statistics involving it.

4.3 Sequence Diagrams

The following sequence diagrams represent two different processes that are a part of Angle Anglers. The first is the process for catching fish and the other is the process of buying an item. Both subsections will contain the sequence diagram itself and then a description describing the process.

4.3.1 Catching a Fish

This sequence diagram shows the process of the player catching a fish. The player begins by casting their rod which they then wait for a fish to bite. After one has bitten, they need to reel in the fish. If they successfully react in time, the system will start the minigame which requires generating a problem that is based on the difficulty of the game. Then, the player is given the problem which they are then required to solve. If they solve it successfully a fish is then generated and returned to the player to be used to update their stats, collection, and increase their currency.

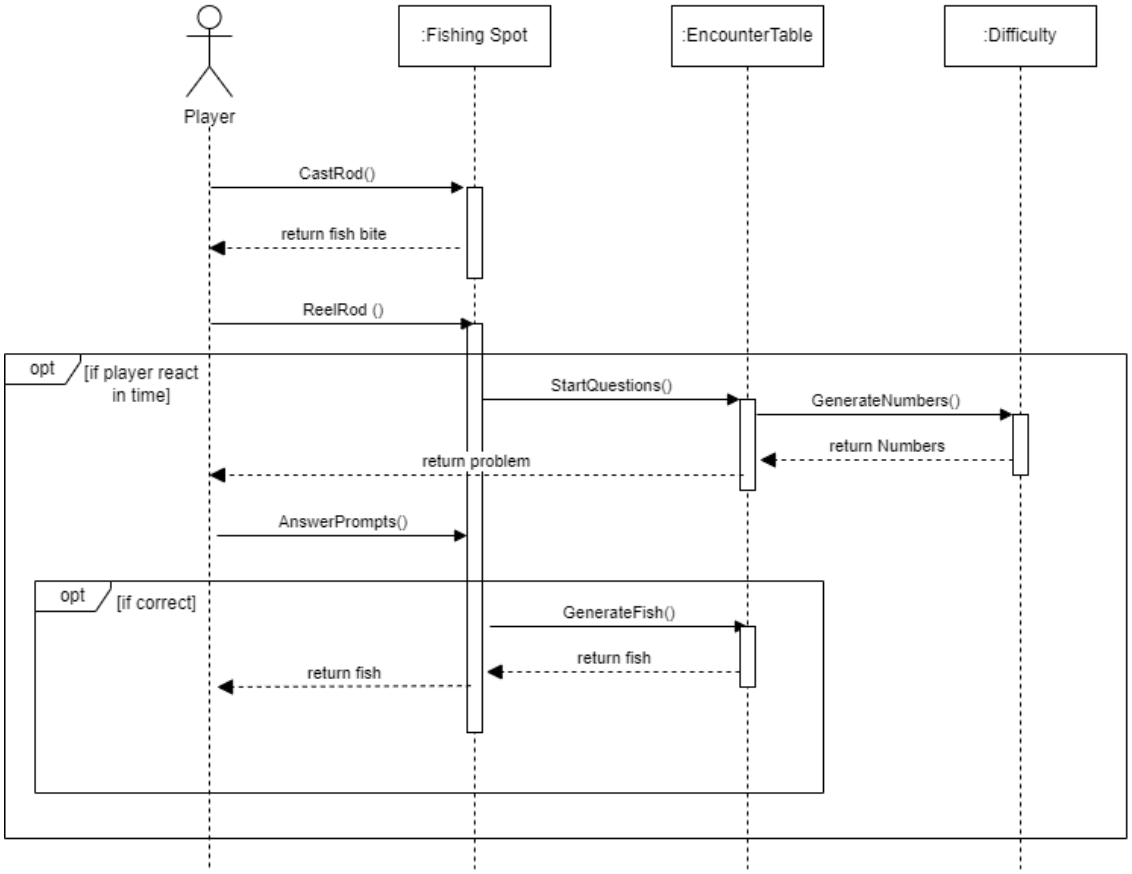

Figure 3 - Catching a Fish Sequence Diagram

4.3.2 Buying an Item

This sequence diagram shows the process of a player buying an item from a shop. The first thing that happens is the player triggers opening the store interface. This will be through walking up to the entrance of the building. Once the interface is open the player will select an item that is available in the store. The selected item will be purchased if the player has enough currency and once the selected item is purchased it is then returned to the player to check if they have it. After that the player can choose to close the store interface or buy another item.

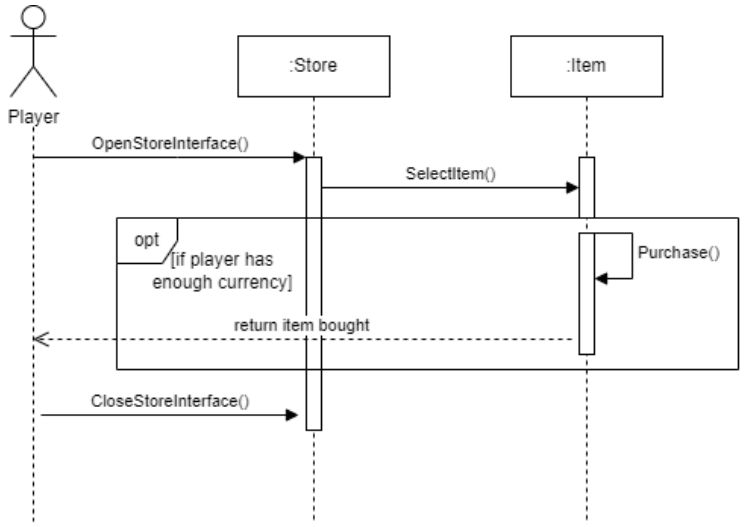

Figure 4 - Buying an Item Sequence Diagram

4.4 State Diagram

Below is a state diagram that describes the flow of the states that the player will be in while playing the game. The player starts on the main menu and then can either choose to quit the game or start it. The player has some options while walking around the world. The first is the main gameplay loop, fishing. The player starts by preparing to cast a rod and if they do that successfully the next states revolve around the questions that are given. If correct they move on to the next question that will be asked and if they are incorrect then the correct answer will be displayed. If a fish is successfully caught it will be displayed and then the player is returned to either casting a rod or just walking away. The other major function is entering a building which will display an interface for the contents of that building. The academy and science lab both have information displayed and then the player can exit, however the store allows the player to buy items as well. The other thing that can be done is to pause the game. When you pause a pause menu will be displayed which allows you to go back to the main menu or just exiting the game.

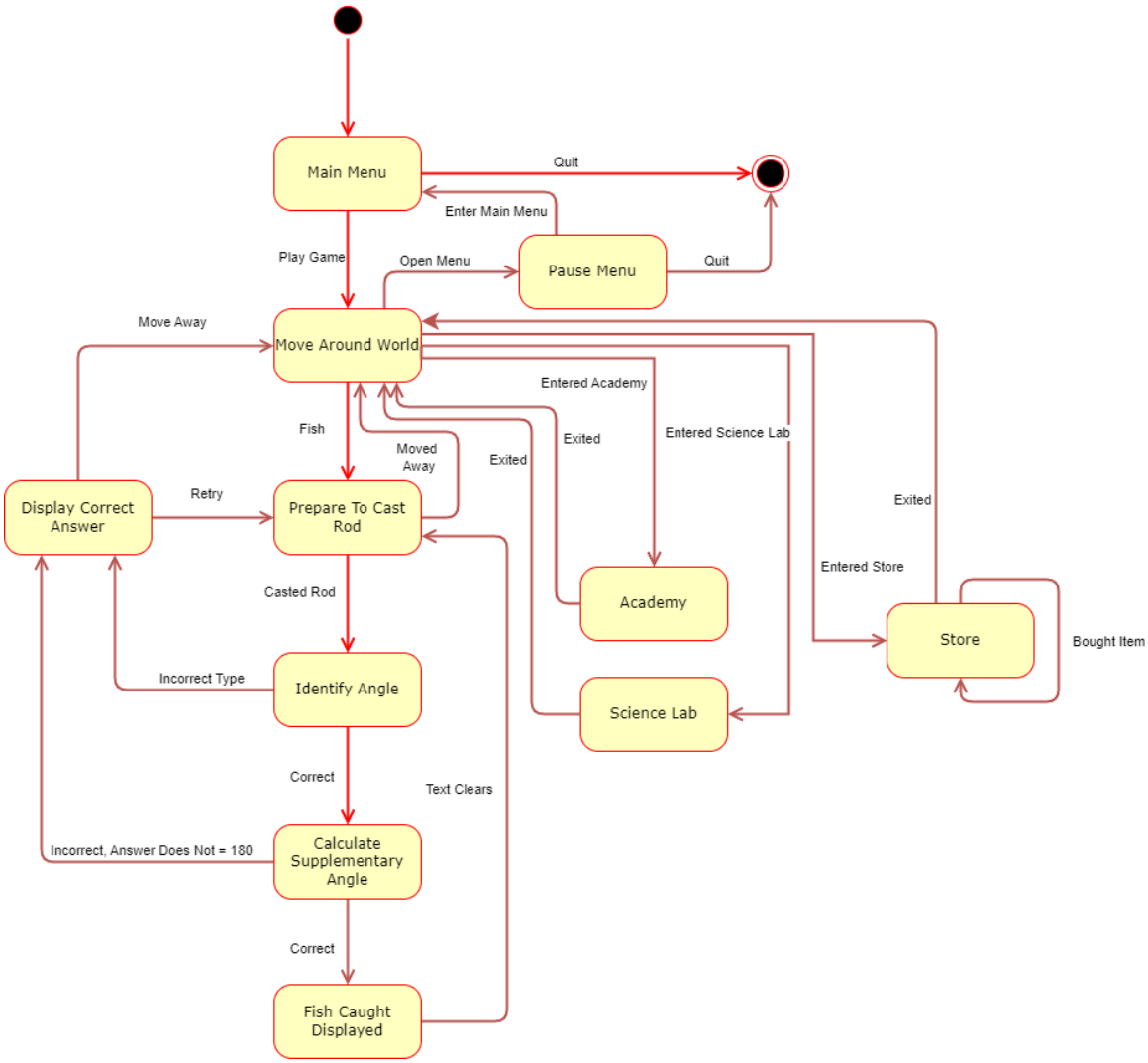

Figure 5 - State Diagram

5 Prototype

The prototype of Angle Anglers will include a main menu and a small portion of the game. The player will be able to move around a basic overworld with a few features that include the different buildings (Academy, Store, and Science Lab) and fish at two separate locations. Each location will contain three different fish for the player to catch which are one of each rarity. There will also be three fishing rods the player can unlock which will alter the difficulty of the game.

5.1 How to Run Prototype

The game can be downloaded from our website (https://angle-anglers.github.io/download.html) under the downloads table. Once there click on the downloads button and it will take you to a Google Drive link with the game zipped. Unzip the game and do not remove any of the folders or files; it may break something. Once unzipped open the Build folder and run the AngleAnglers.exe file located inside it. At the current moment, the game is only compiled for the Windows operating system.

5.2 Sample Scenarios

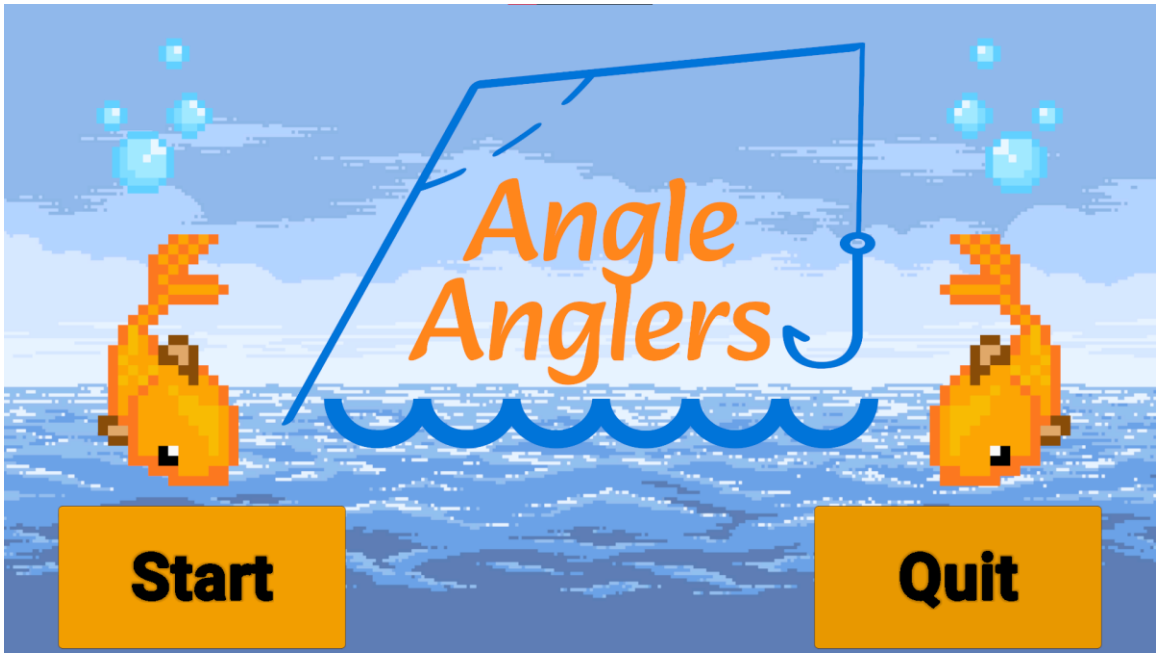

Figure 6 - The main menu of the game which is shown whenever you open the game. The player can either quit or start the game.

Figure 7 - An overlay that displays the controls when you first load into the game to help them navigate around the world.

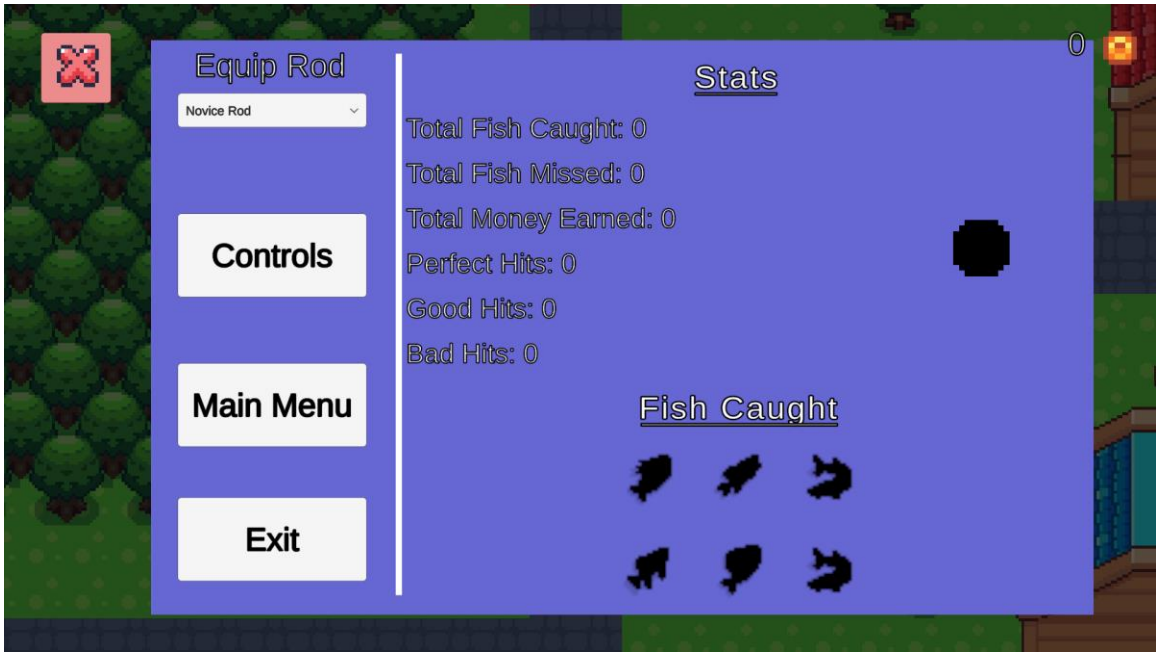

Figure 8 - The game's pause menu shows the players stats, see the fish they caught, allows them to change fishing rods, view the controls, go back to the main menu, and exit the game.

Figure 9 - A zoomed out view of the map. There are two fishing spots, the beach, and a pond. Along with three buildings that offer different functionality.

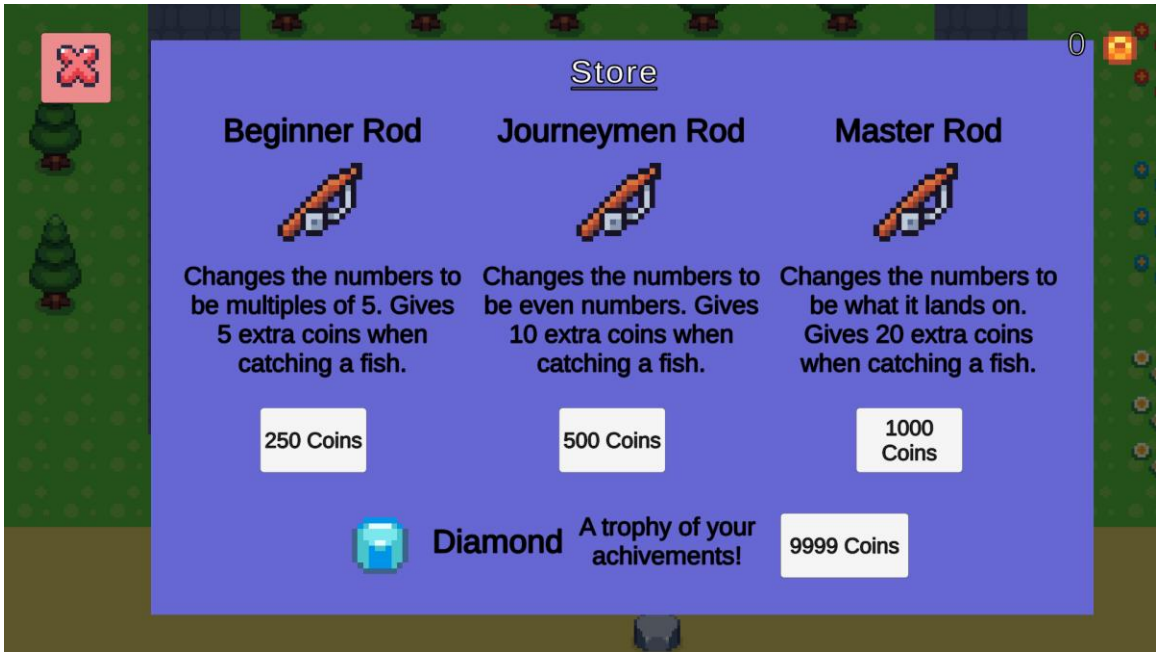

Figure 10 - The store where the play can buy new rods. There are three rods that the player can purchase and each of their effects are described here. A Diamond can also be purchased as a little achievement.

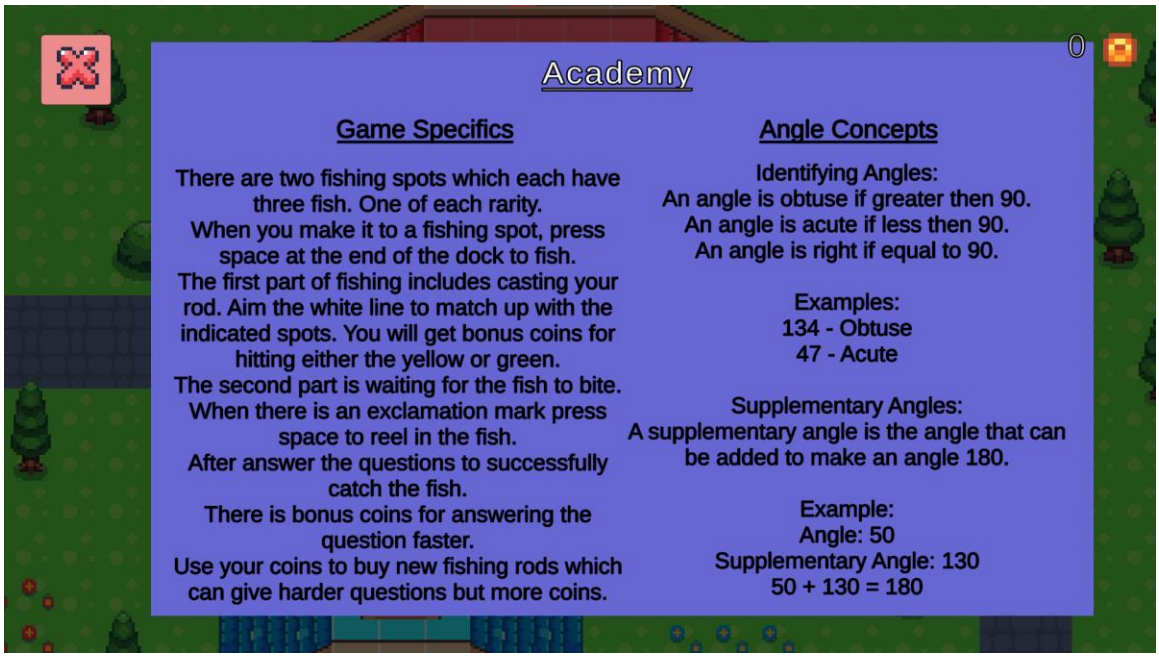

Figure 11 - The academy that describes how to catch a fish and explains the angle concepts that will aid the player in catching a fish.

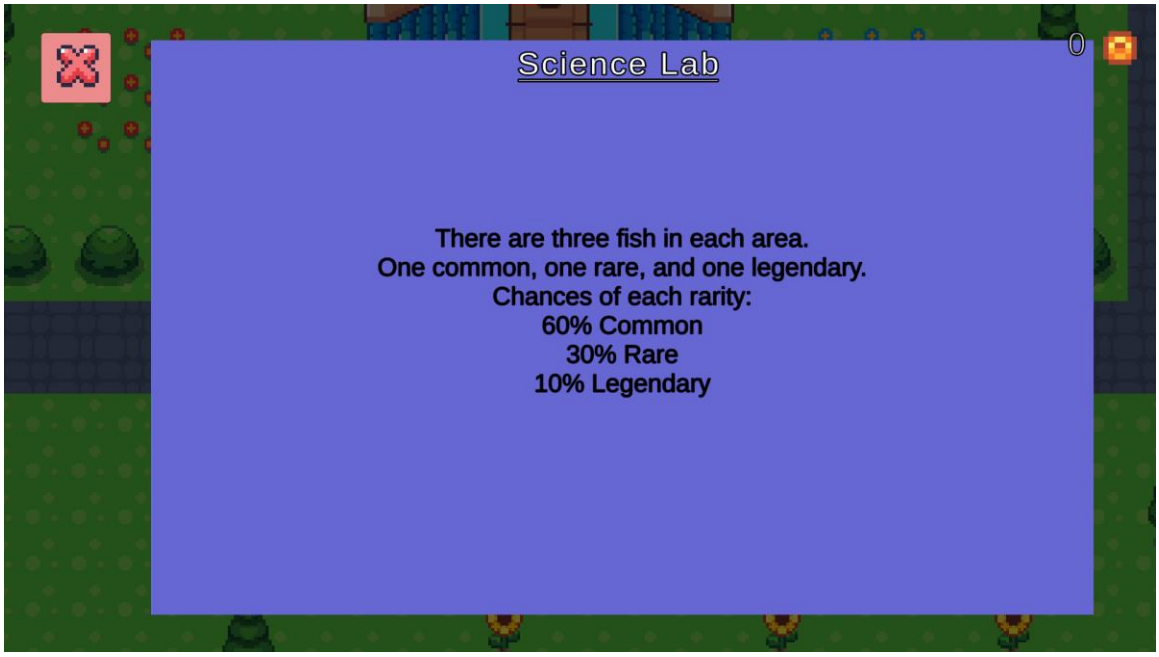

Figure 12 - The science lab that describes how rare each type of fish is.

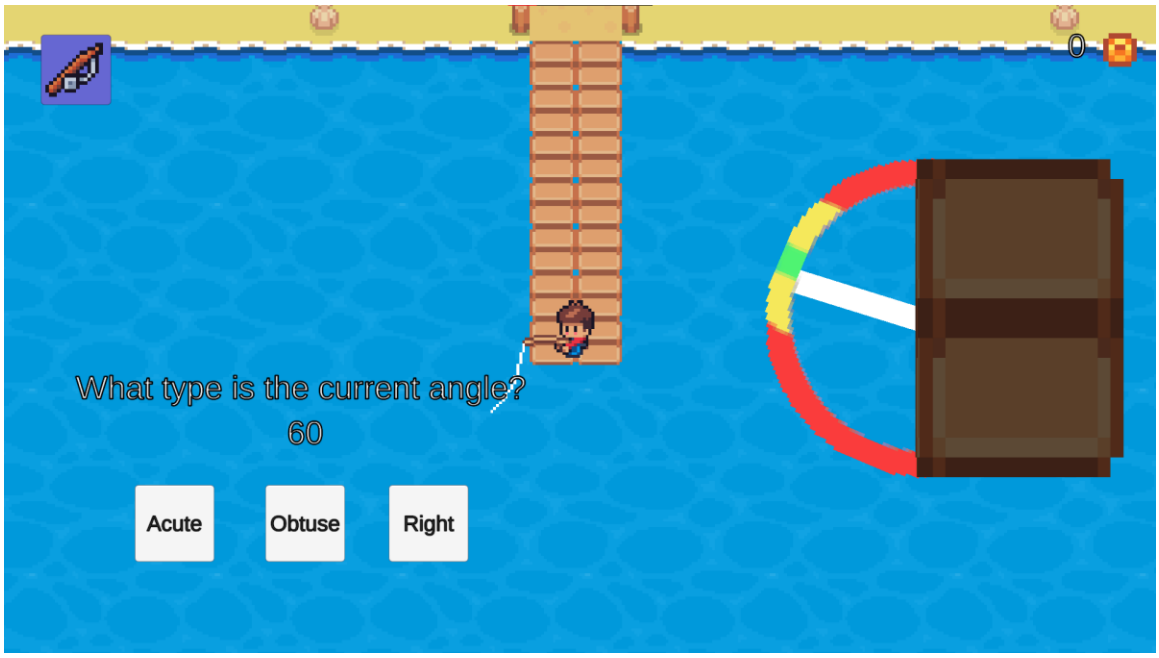

Figure 13 - The player identifies an angle while catching a fish.

Figure 14 - The player gets the answer wrong, and they are corrected.

Figure 15 - The player enters the supplementary angle.

Figure 16 - The player successfully catches a fish, and they are rewarded with 42 coins.

6 References

[1] *beam*, sinneschlösen (Pixabay), 2022, https://pixabay.com/music/video-games-sinnesloschen-beam-117362/

[2] *Angle Anglers*, E. Casasanta, C. Coco, S. Gyger, V. Patel, Software Engineering Group 8, 2024, https://angle-anglers.github.io/index.html

[3] *Keyboard Keys for UI*, Dreammix (Itch.io), 2022, https://dreammix.itch.io/keyboard-keys-for-ui

[4] *Massachusetts Curriculum Framework for Mathematics,* Massachusetts Department of Elementary and Secondary Education, 2017, https://www.doe.mass.edu/frameworks/math/2017-06.pdf

[5] *Ocean Background,* KnoblePersona, 2016, https://opengameart.org/content/ocean-background

[6] *RPG Fish,* Croatz (Itch.io), 2022, https://croatz.itch.io/rpg-fish

[7] *Sunnyside World*, Ddanieldiggle (Itch.io), 2024, https://danieldiggle.itch.io/sunnyside

[8] *Unity Documentation,* Unity Technologies, 2005-2024, https://docs.unity.com

7 Point of Contact

For further information regarding this document and project, please contact **Prof. Daly** at University of Massachusetts Lowell (james_daly at uml.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.