

Software Requirements Specification (SRS)
Project Geotyper

Team: Project group 1 - Geo Typer

Authors: Jack Schnair, Josh Sullivan, Demetri Cassidy, John Swanson, Sam Patriquin

Customer: Massachusetts School Systems

Instructor: James Daly

1 Introduction

Geotyper, an edutainment game, is intended to provide its users with both enjoyment and tools to improve their typing, vocabulary, and geography skills. This document details the requirements for the creation of this project at various levels, and in addition describes how users will be able to interact with Geotyper. The remainder of this section further details the purpose of this document, and in addition section 1.4 provides an in-depth explanation of each section's purpose.

1.1 Purpose

The purpose of this SRS (Software Requirements Specification) document is to give our clients, such as interested school systems, and developers insight into the system that is being developed. The SRS will first contain an overview of this document and what is to be expected. It will then go into detail of what functions, goals, and constraints are involved in producing Geotyper. It will then detail the specific product requirements surrounding Geotyper. After that, various models will be shown to help convey intended use cases and the functionality of our system. Lastly, various samples from our prototype will be shown in effort to depict the functions that Geotyper will provide.

1.2 Scope

The product that we are developing is Geotyper, which is a web-based edutainment game. Our goal with Geotyper is to create a fun environment for students to learn in. Another goal of ours is for our product to be easily accessible, so Geotyper is being developed to be a web based game.

Geotyper's target demographic is students in the fifth grade of the United States educational system. Ideally students who play these games repetitively will gradually increase their proficiency in each topic: geography, vocabulary, and typing. Geotyper is not meant to be a complete educational system, and as such it should not be used as a tool to learn the basics of any of these skills. However, it is intended to complement a student's education, and provide additional ways for students to learn.

1.3 Definitions, acronyms, and abbreviations

- SRS - The abbreviation for software requirements specification
- Edutainment - A video game developed for the purpose of entertainment and education
- Geotyper - The name for our product, an edutainment game
- UI - User interface; a component that is able to be interacted with by the users
- User/Player - An individual person that is playing our game
- Town - The main location Geotyper takes place in, with its setting being inspired by a developing town

- NPC - Non-Playable Character; these are characters that the user will be able to interact with in order to access certain parts of Geotyper
- Minigame - One of three user playable game within Geotyper; each game has a distinct set of actions and goals and is started by talking to an NPC
- Scoreboard - An in town intractable object that will allow the user to see their highest ranking in each of the three minigames
- Git - A version control language
- Github - A website that provides cloud-base git repository hosting services
- HTML5 - Abbreviation for HyperText Markup Language; a markup language used to structure web pages
- WebGL - Abbreviation for Web Graphics Library; a JavaScript API intended to help render 2D and 3D graphics for web pages
- Unity - A game development framework that is capable of building to WebGL
- Indie game - Video games produced by small groups or individuals, rather than a studio
- Itch.io - A web-based hosting service for indie games
- WPM - Abbreviation for words per minute

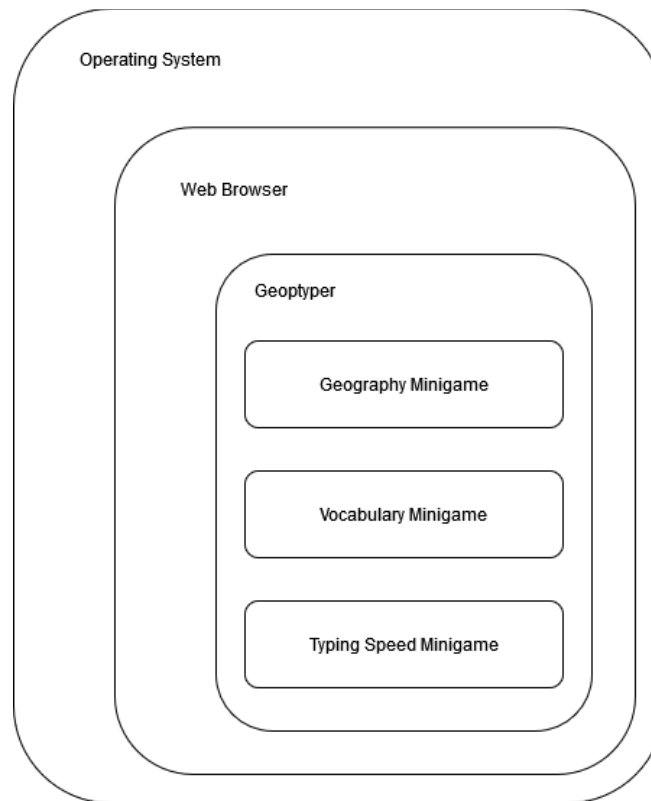
1.4 Organization

The following sections are intended to further describe Geotyper's goals, requirements, and functionality. Section 2 will contain the details regarding various functions and their purpose in Geotyper. It will also describe the expectations of our users and various constraints and requirements that we are operating within when constructing our product. Section 3 contains our product requirements, which provides a detailed description of what must be in Geotyper. Section 4 contains various models used to describe our product, with the purpose of them being to help bridge the gap between our goals for the project, our engineers' design, and our clients' understanding of our product. Section 5 goes into detail on how to access our prototype and our expected system requirements. In addition, several sample scenarios of our product in use are also discussed. Section 6 contains a list of all reference material that was used in the construction of this document and Geotyper, and section 7 contains a point of contact for future communication.

2 Overall Description

The following section contains high level descriptions regarding Geotyper, and its functions. It will detail some of the context around our product, including various constraints, how users will interface with it, and the operations that our product will provide. The core functionality of our system will also be detailed, in terms of the utility that our users can expect to derive from our product. We will also detail some of our expectations about our users, including who our product is targeted towards and what background we expect them to have. Lastly, we will discuss future requirements that are currently out of scope but have potential to be developed in the future.

Figure 2.1: Representation of system



2.1 Product Perspective

Geotyper is intended to be a standalone browser game that provides an entertaining method for students to practice their typing skills, while also learning new vocabulary terms and geography facts. Despite it being standalone, Geotyper could easily be incorporated into any already existing educational system. Geotyper will play within the browser, utilizing the WebGL framework. This method of serving our product should provide an engaging and accessible method for students to exercise their social studies and typing skills. In addition, serving Geotyper through the web makes it accessible when at-home/remote learning is necessary.

Our main mode of user interaction and communication will be through NPCs in the game. Geotyper will feature three minigames, one for each: vocabulary, geography, and typing. Users will be able to launch these minigames by talking to the game's corresponding NPC. Upon completion of a game they will receive a score indicative of how well they performed. In addition, users will be able to track their progress through a scoreboard, which will be accessible to the users through the town, similar to if the users were to access a minigame.

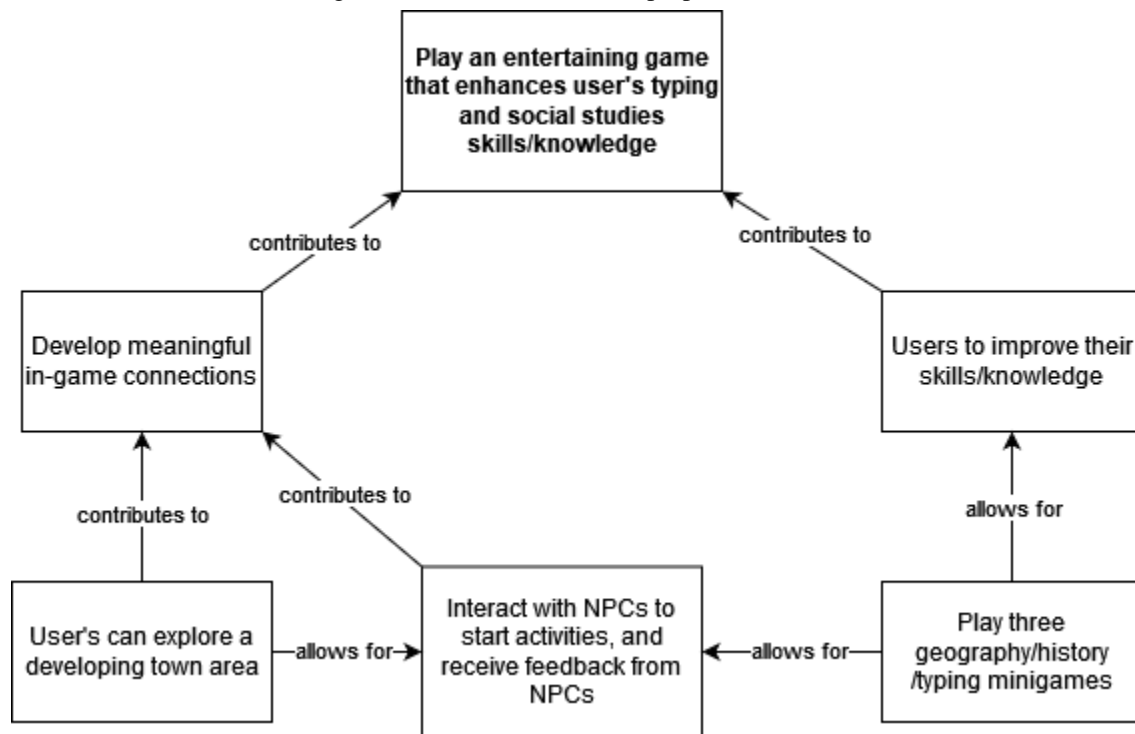
In terms of the constraints that our game will have, Geotyper will require both mouse and keyboard inputs as methods of input. It will also require users to have access to the internet, which will be the user's main limitation. Besides this, there will be minimal hardware or software constraints, as all that is required of the user is a system capable of running a web-browser, and a connection stable enough to maintain connection to our site. Lastly, as our game is intended for grade five, we will restrict our content to that which is relevant to that age group, based on the Massachusetts History and Social Science Framework.

2.2Product Functions

The main functions of Geotyper include: interacting with characters to start minigames, playing minigames, and monitoring their progress through a scoreboard. All of these functions are accessed by exploring the main town that Geotyper takes place in. Providing an entertaining way for user's to learn will encourage users to test their typing and geography skills, and make the game accessible to a wide range of users.

The goal of these functions is to provide users with an accessible way of playing games that add to their education, while requiring minimal overhead to access. Each instance of a minigame will be a fresh start, in the sense that minigames will not build off progress from the user's previous games. This will allow users to access our site and play any number of minigames. In addition, an interactive in-town scoreboard allows for the users to keep track of their progress in such a way that fits well with the theme of Geotyper.

Figure 2.2: Functions and their purpose/end result



2.3 User Characteristics

Geotyper's demographic is a student that is in the fifth grade of the US education system, or a student that is around ten or eleven years old with equivalent education experience. The knowledge expectations that we have for our users is that they: should have experience with basic US geography skills, such as being able to identify some states by name. They should also have experience with, but are not expected to have proficiency in typing. An estimation of their minimum typing speed is around 20 words per minute.

2.4 Constraints

As a typing game, Geotyper requires a keyboard in order to function, and there are no alternative control schemes. As such, there is currently no support for mobile devices. There are also no disability options for those with motor, visual, or cognitive disabilities, and as a result these affected users could be limited when playing Geotyper. While our game will feature some music, there will be no mandatory audio cues intended as a part of gameplay, so auditory disabilities will not be a problem. Further expansion upon Geotyper to make it more accessible to these affected students is detailed in section 2.6.

In terms of constraints to our target audience, Geotyper will likely be most successful in the United States, due to it being based on the Massachusetts History and Social Science Framework. In addition, as each state in the United States of America controls its own education, the Massachusetts History and Social Science Framework may not reliably reflect the education level of all fifth grade students in every state.

Geotyper's security requirements will be handled by Github where the game will be hosted using WebGL. Constraints for the game's successful launch are teachers' interest in a typing game that teaches vocabulary and United States geography and Geotyper's capability to engage and teach students.

2.5 Assumptions and Dependencies

As it is a web based game, users must have access to a browser that supports both HTML5 and WebGL. In addition, Geotyper will be reliant on a stable internet connection and assumes it is available in order to function. The game will be built using the Unity game engine. The game is run through the WebGL framework and relies on its ability to adapt the code designed in Unity to play on the user's internet browser of choice. The game is reliant on Github to host its source code and handle any updates. Using Unity's long-term support build, it is a safe assumption that any issues that arise within Unity will be resolved. As a result, it is assumed that any future updates to web browsers, WebGL, or Github will not negatively affect Geotyper and it will continue to function as expected.

In order to read the text in the game, users should have developed at least a fourth grade English reading level, as decided by the Massachusetts English Language Arts and Literacy Curriculum Framework. In short, the ELA framework details that a fourth grade student should be able to read and analyze text in such a way that allows them to understand its basic meaning (general idea of text, plot/meaning, etc)[4]. It is also assumed that the users have a basic understanding of computers (so they can access the game) and a rudimentary ability to type.

2.6 Apportioning of Requirements

Currently, Geotyper's in-game scope is limited to one interactable town and three minigames. However due to its modular design, Geotyper has numerous avenues for future expansion. Expansion upon the number of minigames could increase the game's ability to gain and retain user interest. It would also increase the game's ability to teach different areas of study in social sciences through different minigames. In addition, a greater variety of questions in the individual minigames would increase Geotyper's replayability which will further help Geotypers ability to teach students. The game's scope is currently limited to the expected capabilities of fifth graders. This could be expanded upon by adding in difficulty levels, which would allow the game to cater to other grades, such as from 4th to 6th to start. Lastly, a save game feature is planned for the future, which will enable expanding the session scoreboard to be a user's lifetime scoreboard

To further improve the educational aspect of Geotyper, studies on the game's ability to improve student's geography and vocab knowledge could be used to improve the game's difficulty; this could enable Geotyper to challenge students in each grade level at a rate more conducive to their development. Geotyper could also be used in future studies on the capability of video games to teach and engage students, which could improve its user retention.

Geotyper currently has no disability options and could be difficult to use for students with motor, visual or cognitive disabilities. Adding options to help disabled users such as settings for colorblindness, alternate control schemes, text to speech, and key sensitivity controls could expand the game's reach and capability to teach all students of different ability levels. In addition, the game will currently only be available in English but could be expanded to more languages, which would further expand the game's audience.

As it is a browser game, Geotyper is currently limited in its availability due to necessitating an internet connection. Building a Geotyper desktop or mobile app would expand the game's availability and potentially draw in parents and teachers that may prefer an app over a game on an internet browser.

3 Specific Requirements

1. The game will be a collection of replayable typing minigames
 - a. The game will take place in a developing town
 - b. The user will interact with NPCs to start minigames

- c. There will be vocabulary, geography, and typing minigames
 - i. All minigames will be single player
 - 2. The game will follow Massachusetts History and Social Science Framework
 - a. The game's difficulty will be based off of the expected knowledge of a fifth grade student
 - i. Students should have solidified their basic US geography knowledge, and should be able to identify some states. Their expected typing speed is around 20 WPM.
 - 3. The user should be able to open a settings menu when in town
 - a. Doing this pauses the game, preventing character movement and NPC interaction
 - b. The user can adjust the volume via a slider
 - i. After the slider has been selected, pressing the left arrow key will decrease the volume, up to a maximum until it is released
 - ii. After the slider has been selected, pressing the right arrow key will increase the volume, up to a maximum until it is released
 - iii. The user can also select a position on the slider with their mouse to manipulate the volume
 - c. The user can exit the settings by selecting an exit icon with their mouse
 - 4. The user should be able to control their character sprite with a keyboard
 - a. Different keys correspond to different actions
 - i. W or up arrow to move up
 - ii. A or left arrow to move left
 - iii. S or down arrow to move down
 - iv. D or right arrow to move right
 - v. ESC to open settings menu
 - vi. E to interact with NPCs or scoreboard
 - 5. The user can pick from three different minigames
 - a. Vocabulary Game
 - i. The game will take place over the course of two minutes
 - 1. The user will be displayed a definition, then be required to type the word
 - 2. Users will also be able to choose to be displayed a hint
 - a. Hints contain a list of three words, with one being the correct word
 - 3. Users score will increase as they provide correct answers, and they will lose points for incorrect answer and hints
 - b. Geography Game
 - i. The user will be displayed a picture of the a map with a state highlighted, and they will then guess the state
 - ii. The game will be completed once they have guessed all states
 - 1. Users will be score based on how long it takes them to complete all states and how many many states were guessed correctly

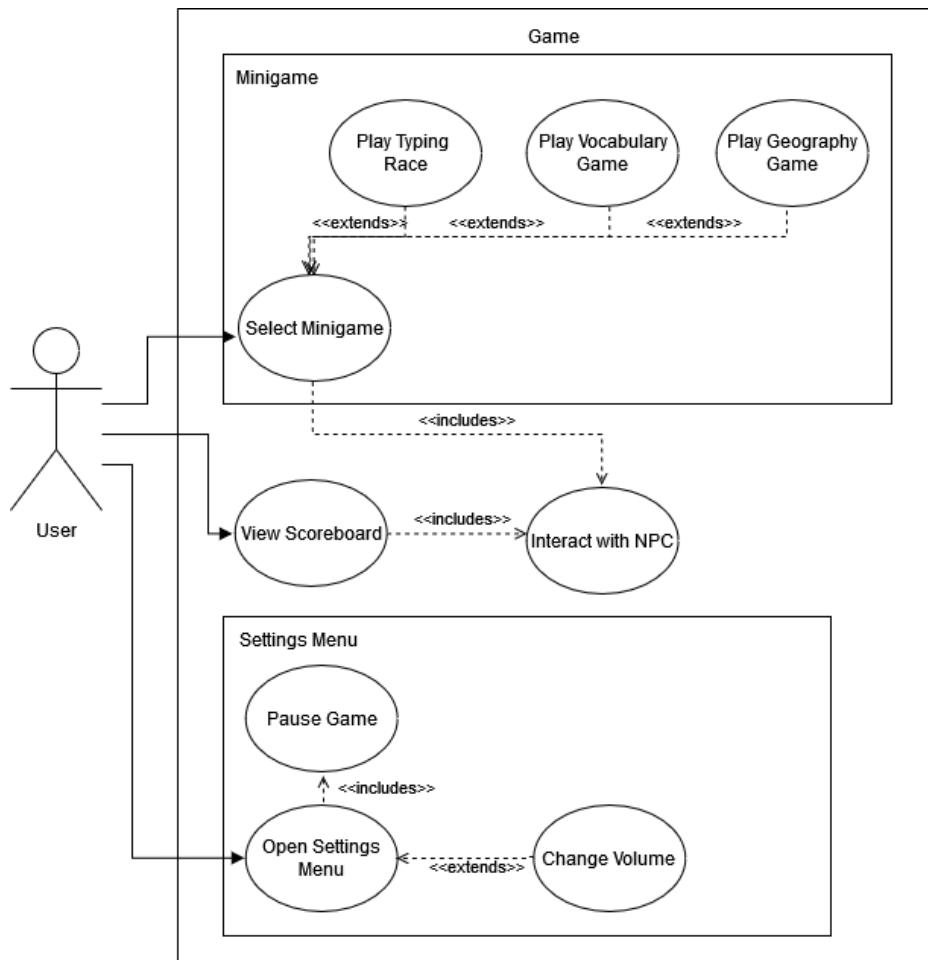
- c. Typing Game
 - i. The user will engage in a one player typing race
 - 1. Players must enter text to match displayed text with exact spelling and capitalization
 - a. User's target typing speed is 20 WPM
 - ii. Players receive score based on words completed and average WPM
 - d. The above game selection choices will be accessible by talking to various NPCs throughout the town
6. The user should be able to view a session scoreboard containing various statistics
 - a. The user's high scores in different games
 - i. Vocabulary Game: Shows the user's highest score
 - ii. Geography Game: Shows the highest amount of accurate answers
 - iii. Typing Game: Shows the highest average WPM count, and the user's highest score

4 Modeling Requirements

Use Case Diagram:

The use case diagram below depicts how the player will interact with the product. The three main functionalities the player can access are selecting minigames, viewing the scoreboard, and opening the settings menu. These functionalities are called use cases and can be related to other use cases through arrows that indicate inclusion and extensions. Use cases that are included by another use case are a part of that functionality. A use case that extends a functionality is a conditionally used case. These two arrows are indicated by the labels “<<include>>” and “<<extends>>”. In this particular use case diagram there is only one actor, the player, so every use case is used by them.

Figure 4.1: User’s capabilities



Use Case Name:	Open Settings Menu
----------------	--------------------

Actors:	Player
Description:	Menu appears on the screen that automatically pauses the game. Through the menu the player can change the volume and exit the game.
Type:	Primary
Includes:	Pause Game
Extends:	N/A
Cross-refs:	Requirements 4 and 5av
Uses cases:	This occurs when the player presses the “esc” key, pausing the game.

Use Case Name:	Pause Game
Actors:	Player
Description:	Freezes the game so that nothing happens until it is resumed.
Type:	Secondary
Includes:	N/A
Extends:	N/A
Cross-refs:	Requirement 4a
Uses cases:	The game always pauses when the settings menu is opened.

Use Case Name:	Change Volume
Actors:	Player
Description:	The player is able to adjust the volume of the game by incrementing or decrementing a slider. This is done by opening the settings menu, selecting adjust volume and using the left and right arrow keys to move the slider, or by selecting a position for the slider with the mouse.
Type:	Secondary
Includes:	N/A
Extends:	N/A
Cross-refs:	Requirement 4b
Uses cases:	Used when the player presses the up or down arrow key when “adjust volume” is selected in the settings menu.

1.

Use Case Name:	View Scoreboard
Actors:	Player
Description:	Brings up a scene that includes the user's session statistics of each game including relevant high scores.
Type:	Primary
Includes:	Interact with NPC
Extends:	N/A
Cross-refs:	Requirement 5 and 8
Uses cases:	When the player interacts with the scoreboard NPC the scoreboard will be viewed.

Use Case Name:	Interact with NPC
Actors:	Player
Description:	Will either initiate a dialogue with an NPC to determine what game to play or view the scoreboard depending on which NPC is interacted with.
Type:	Secondary
Includes:	N/A
Extends:	N/A
Cross-refs:	Requirements 5avi, 6, 7a
Uses cases:	Used when the player presses "E" next to an NPC.

Use Case Name:	Select Minigame
Actors:	Player
Description:	Depending on which NPC the player interacts with the player will be asked if they would like to play one of three games. If the player answers yes then the minigame will then be played, if no then the player will exit the dialogue.
Type:	Primary
Includes:	Interact with NPC
Extends:	N/A
Cross-refs:	Requirement 7

Uses cases:	Used when interacting with a townspeople NPC and the play game dialogue path is chosen.
-------------	---

2.

Use Case Name:	Play Typing Race
Actors:	Player
Description:	Starts the “typing race” game where the player is shown words and must attempt to type them with matching capitalization/spelling, while having their words per minute tracked and displayed to them.
Type:	Secondary
Includes:	N/A
Extends:	Select Minigame
Cross-refs:	Requirement 7a
Uses cases:	Used when the user selects “typing race” as their game choice.

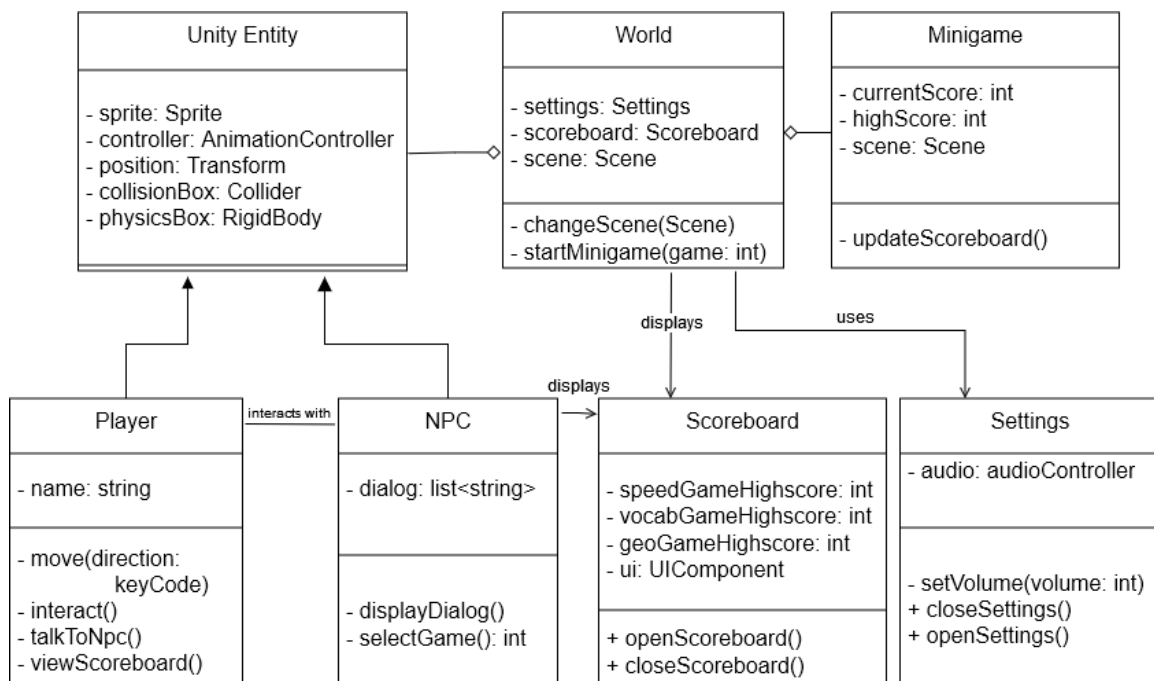
Use Case Name:	Play Vocabulary Game
Actors:	Player
Description:	Plays the “vocabulary game” where the player is shown definitions and has to type the word defined by it.
Type:	Secondary
Includes:	N/A
Extends:	Select Minigame
Cross-refs:	Requirement 7b
Uses cases:	Used when the user selects “vocabulary game” as their game choice.

Use Case Name:	Play Geography Game
Actors:	Player
Description:	Plays the “geography game” which is a game where the player is shown a highlighted state and will have to enter the correct state name.
Type:	Secondary
Includes:	N/A
Extends:	Select Minigame
Cross-refs:	Requirement 7c
Uses cases:	Used when the user selects “geography game” as their game choice.

Class Diagram:

The class diagram contains all of the classes that make up Geotyper. These include: “Unity Entity”, “World”, “Minigame”, “Player”, “NPC”, “Scoreboard”, and “Settings”. Each class is represented by a box in the diagram that has three sections: the name of the class, its attributes, and its functions. The different relationships between the classes can be seen by the type of arrow connecting them. Black triangle heads inherit from the class they are pointing to. Arrows with open heads point to classes that they use. The white diamonds represent aggregation, meaning that the class that the white diamond is next to contains multiple of the class the connection is coming from.

Figure 4.1: Class Diagram



Data Dictionary:

Element Name	Description
Unity Entity	An entity in the Unity engine is an individual “thing” that populates a game or program’s world. They are used to identify pieces of data that belong together.
Attributes	
sprite: Sprite	A 2D graphics object that can be used to represent objects in the game.
controller: AnimationController	A tool that manages animations and animation transitions for objects based on in-game conditions.
position: Transform	A component that represents the position, rotation, and scale of an object.
collisionBox: Collider	A component that defines the shape of an object and handles collisions with other entities in the game’s world.
physicsBox: Rigidbody	A component that utilizes physics simulation to handle the motion of an object.
Relationships	
A collection of components that make up and represent objects like the Player class and the NPC class. Entities exist within the game’s world.	
UML Extensions	
N/A	

Element Name	Description
Player	An in-game object that the user controls. This represents the game's protagonist.
Attributes	
name: String	A string that represents the player's name.
Operations	
move(direction: keyCode)	Triggers the user's character to move in a direction that corresponds to their key press.
interact()	This allows the user to interact with an NPC in the world that is nearby.
talkToNpc()	This triggers a nearby NPC object to initiate dialog with the user.
viewScoreboard()	This triggers the game's session scoreboard to appear on-screen.
Relationships	
The objects from the player class are able to interact with NPC objects and can trigger them to initiate dialog with the user. This class is also able to modify attributes of the Settings class.	
UML Extensions	
N/A	

Element Name	Description
NPC	An object in the game's world that responds to interactions from the Player class and initiates minigames.
Attributes	
dialog: list<string>	A list of strings that an NPC uses when talking with the Player class.
Operations	
displayDialog()	This operation displays a dialog string through the UI.
selectGame(): int	This occurs when selecting which minigame to play. Returns an integer value representing the chosen game.
Relationships	
As an entity, members of the NPC class occupy the game's world and respond to interactions from the Player class. Class members are capable of interacting with the Scoreboard class and causing members to display scores on-screen.	
UML Extensions	
N/A	

Element Name	Description
World	This class contains different elements of the game, such as entities and minigames, and facilitates the scenes in which these elements are located.
Attributes	
settings: Settings	An object containing the current game's settings selected by the user.
scoreboard: Scoreboard	An object containing high scores for their video games.
scene: Scene	A Unity object containing the assets relevant to the user's current visuals.
Operations	
changeScene(scene: Scene)	Changes the world's current scene.
startMinigame(game: int)	This operation is used to launch a specific minigame that the user can play.
Relationships	
A world is composed of different entities, such as NPCs and the Player. They are capable of displaying information from the scoreboard class and can utilize a Settings object to handle user preferences.	
UML Extensions	
N/A	

Element Name	Description
Minigame	A class for handling the minigames that users can play. Minigames are initiated as a result of the Player class interacting with members of the NPC class.
Attributes	
currentScore: int	The user's current score in the ongoing minigame.
highScore: int	The highest score achieved in a specific minigame by the user.
scene: Scene	The Unity object that contains the assets relevant to the user's current minigame.
Operations	
updateScoreboard()	This operation is called to update the scoreboard with the latest high scores, if the user achieves a new high score.
Relationships	
Minigames are contained within and are initiated by the game's world. Once the user achieves a new high score, the Minigame class interacts with the Scoreboard class to update its information.	
UML Extensions	
N/A	

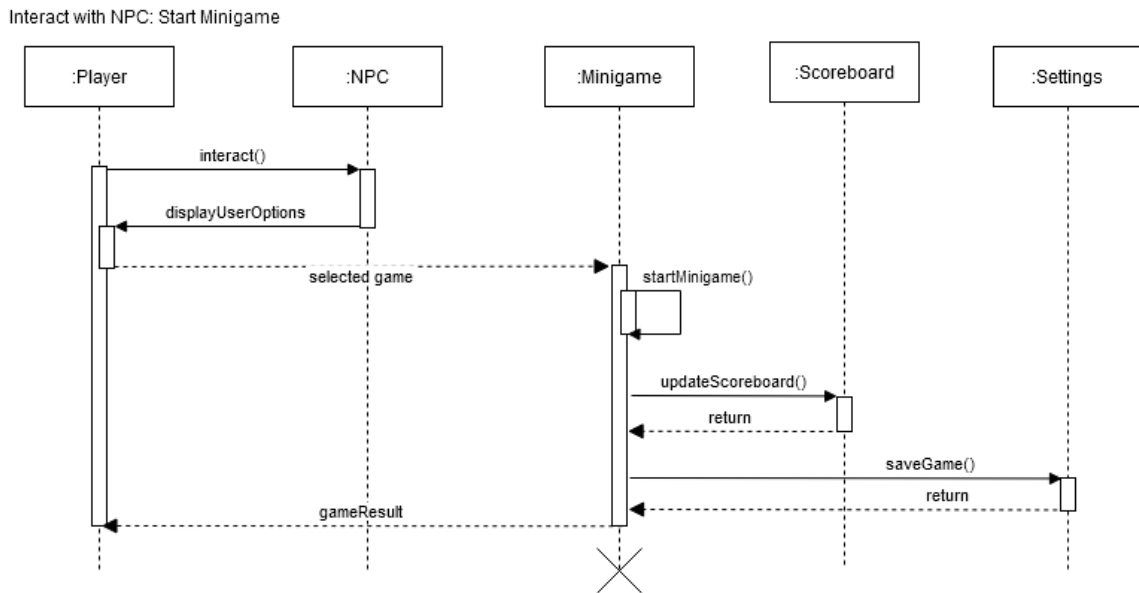
Element Name	Description
Scoreboard	A class for displaying the user's session high scores for the game's three minigames.
Attributes	
speedGameHighscore: int	The session high score for the typing race minigame.
vocabGameHighscore: int	The session high score for the vocabulary minigame.
geoGameHighscore: int	The session high score for the geography minigame.
ui: UIComponent	The user interface component for displaying the scoreboard's information on-screen.
Operations	
openScoreboard()	Displays the scoreboard's user interface on-screen.
closeScoreboard()	Removes the scoreboard's user interface from the screen.
Relationships	
Objects of the Minigame class update a Scoreboard's information whenever a new high score is achieved.	
UML Extensions	
N/A	

Element Name	Description
Settings	Records and manages the volume levels of the game
Attributes	
audio: audioController	This attribute manages and plays audio clips.
Operations	
setVolume(volume: int)	Adjusts the volume levels for certain parts of the game's audio.
closeSettings()	Removes the setting options from the screen.
openSettings()	Displays the game's settings options on-screen.
Relationships	
The World class utilizes the Settings class to manage the game's volume. It is also capable of opening and closing the settings options on-screen.	
UML Extensions	
N/A	

Sequence Diagram 1, Figure 4.1:

The following two diagrams depict two possibilities of what could happen when the player selects a minigame. The process of doing this requires the player to first talk to a townspeople NPC. The NPC will display dialogue that displays the name of one of the following games: “Vocabulary game”, “Geography game”, “Typing race”. The player can then select “yes” or “no” for if they want to play the game or not. The first diagram here shows what will happen when one of the games is selected from the NPC. After the minigame is created, the minigame object will then launch the minigame of the correct game type and continue looping the game until it is over. Once the game is over, the scoreboard will update through the scoreboard object if a high score was achieved. This sequence ends by returning the game result to the player so they can see it.

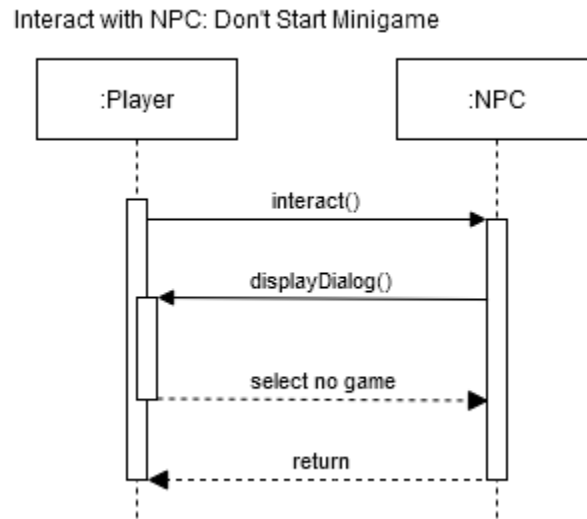
Figure 4.1: Sequence Diagram - Start and play game



Sequence Diagram 2, Figure 4.2:

This diagram shows the sequence of actions that occur when in the previous scenario the player chooses “no” from the NPC’s dialogue. When this happens the NPC will simply recognize that no game was selected and the dialogue will end. The player will then be free to walk around the town once again and interact with something else.

Figure 4.2: Sequence Diagram - Start and cancel game selection

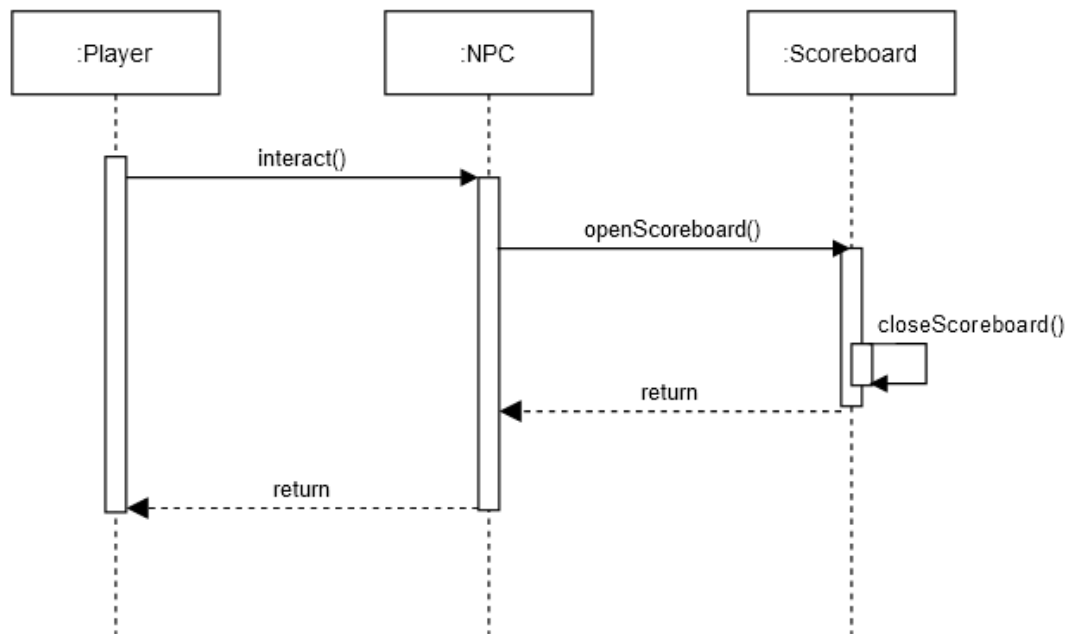


Sequence Diagram 3, Figure 4.3:

The following diagram shows the sequence of events that occur when the player wants to view the scoreboard. The player will start by finding the scoreboard NPC in the town and interacting with it. This works exactly like it does with a regular townsperson NPC, but with the scoreboard it will call the scoreboard object and the scoreboard scene will open. The player can view the scene as long as they like until a key is pressed and the scoreboard closes. Once this happens the regular town scene will be viewed again and the player can walk around normally.

Figure 4.3: Sequence Diagram - Display Scoreboard

Open Scoreboard

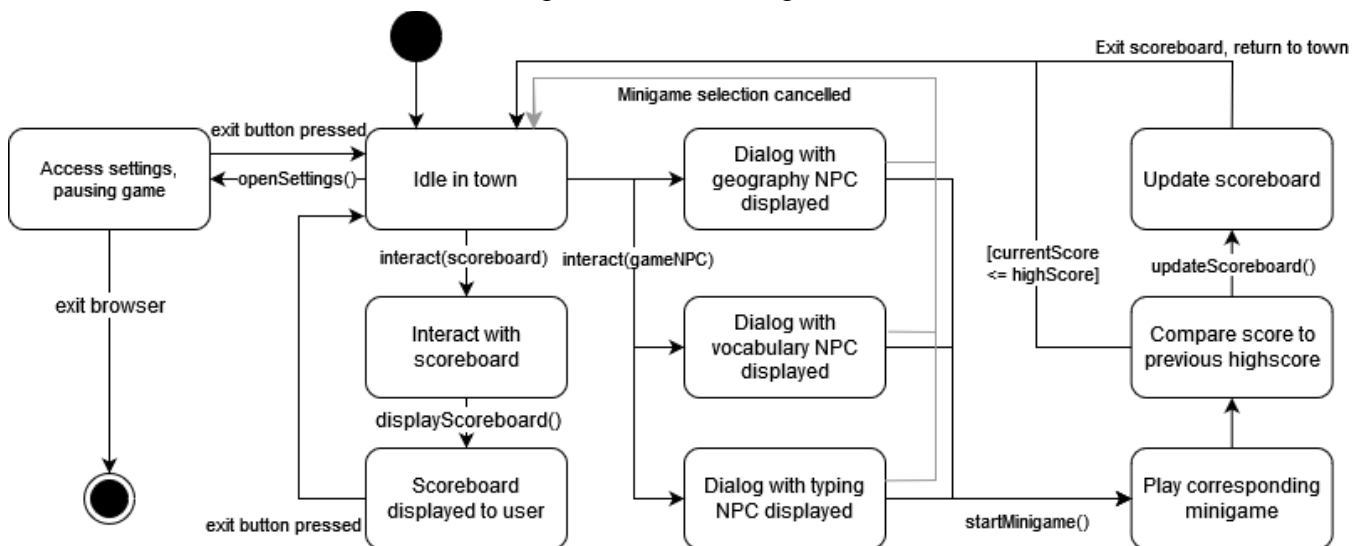


State Diagram:

This diagram shows the different possible states that Geotyper can enter while in use. The state diagram begins at the solid black circle, which is when the user enters Geotyper. The user is able to initiate transfer to different states via various in game actions, which correspond to function calls. These transitions are depicted by solid arrows, with the corresponding function shown on the label. At any point, after accessing the settings menu, the user can exit the game. This is represented by the black circle enclosed by another circle on our diagram.

After the initial state, users are able to interact with a scoreboard object to open it, or interact with an in town NPC to play their associated minigame. After completion of the minigame, their score is compared to their highscores, and the scoreboard will be updated if necessary. If the scoreboard is opened, users can exit it by selecting its exit button to return to town. Lastly, users can exit the settings menu by pressing escape, where their game will pause. They can return to town by selecting this menu's exit button, or they can close their browser, exiting the game.

Figure 4.4: State Diagram



5 Prototype

The prototype will demonstrate the town setting, player control, NPC interaction, a settings menu, and scoreboard interaction. The player (currently a white circle) will be able to move around a placeholder town and collide with three NPCs representing each minigame. The player will also be able to collide with the scoreboard. Interacting with the NPCs by hitting 'e' on the keyboard will bring up a UI component that will ask the player if they want to play a specific minigame or exit the conversation. If the user accepts that prompt, then the corresponding minigame will be started. Next, if the player chooses to interact with the scoreboard, a UI component will appear displaying the user's high scores from each game. The player can exit this menu. Finally, if the player hits the 'esc' key, a UI component will appear containing the settings menu. The user will be able to modify the game's output volume from this menu.

4. 5.1 How to Run Prototype

To run the prototype, any web browser is required along with a stable internet connection. The user's browser should support HTML5, but this should be default to any web browser. The game is hosted on itch.io, a website built for hosting indie games. As long as the user can access this site, the prototype will run. There are no operating system restraints, the game will run on all major operating systems.

Link to page: <https://infinite-cut.itch.io/geotyper>

5.2 Sample Scenarios

Figure 5.1: Game selection prompt



Fig. 5.1) Represents the user pressing the key 'e' when interacting with the NPC standing for the vocabulary minigame. In the future, this interaction will launch the minigame.

Figure 5.2: Session Scoreboard

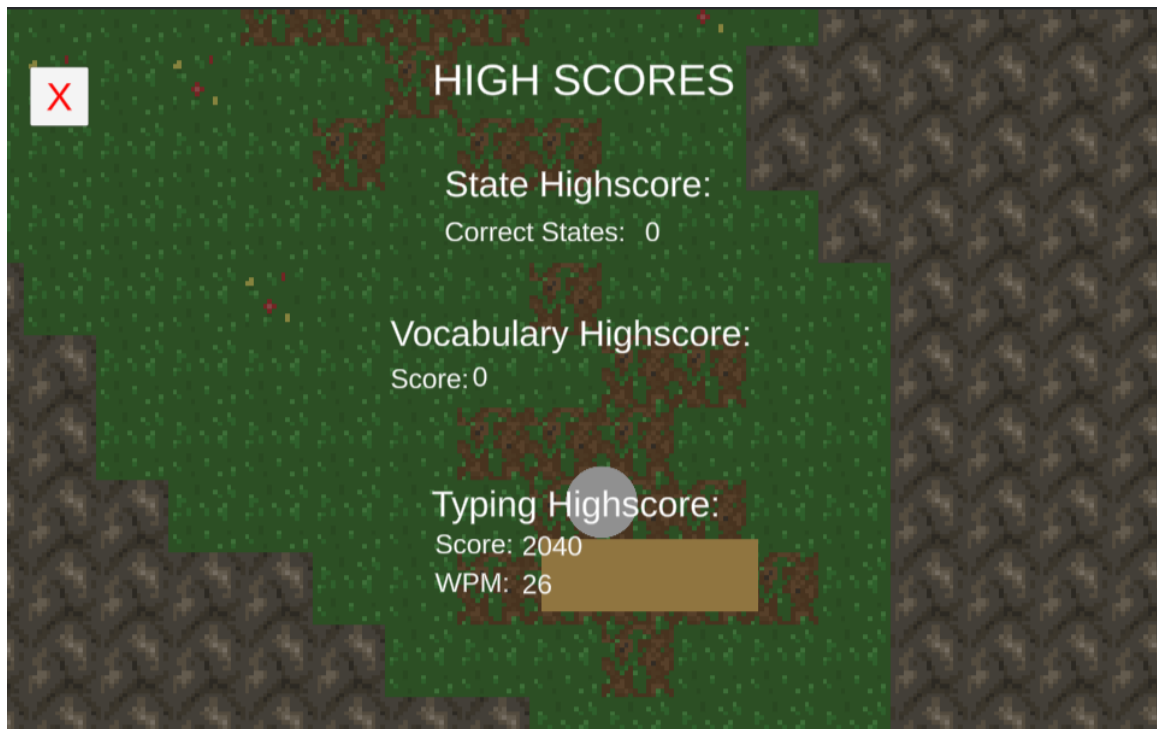


Fig 5.2.) Represents the user interacting with the scoreboard by pressing the key 'e'. In the future, the user's highscores will be displayed here.

Figure 5.3: Settings Menu

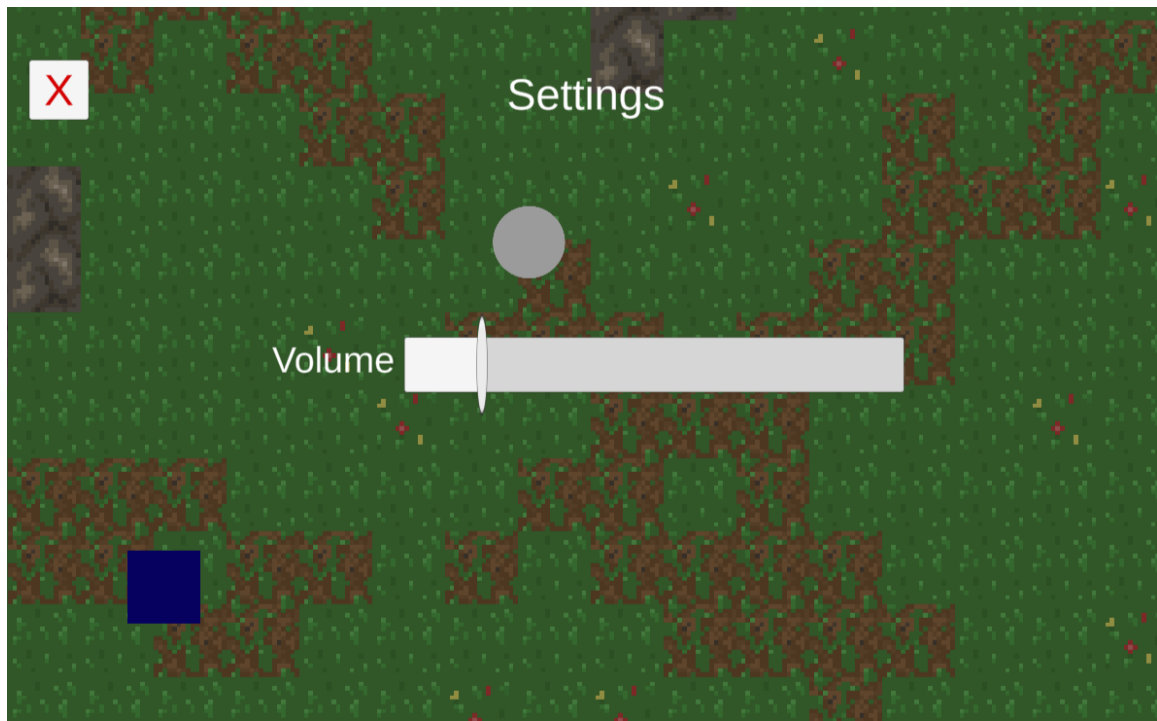


Fig 5.3.) Represents the user opening the settings menu by pressing the 'esc' key. In the future, the user will be able to change various settings here.

Figure 5.4: Geography Minigame



Fig 5.4.) In-game screenshot of the geography practice minigame. The user is being shown a state to guess, the time elapsed, how many states they have left to guess, and their accuracy so far.

Figure 5.5: Typing Minigame



Fig 5.5.) In-game screenshot of the typing practice minigame. The user is being shown a word to type, their time remaining, their current wpm and their score. The next letter to type is highlighted in blue.

Figure 5.6: Vocabulary Minigame

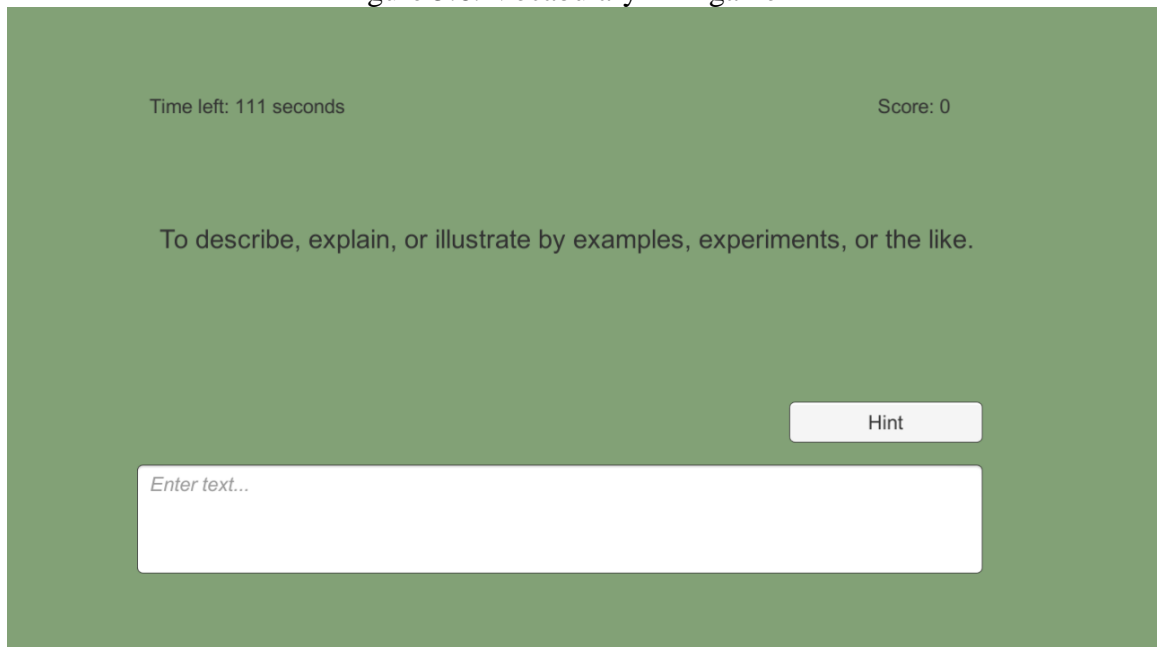


Fig 5.6.) In-game screenshot of the vocabulary practice minigame. The user is being shown definition, their time remaining, and their score. They are prompted to type in the corresponding word, and can also get a hint if they are stuck.

6 References

- [1] Sullivan, J. (n.d.). Software Engineering Product. Retrieved from <https://joshua119.github.io/SoftwareEngineeringProject/>.
 - [2] D. Thakore and S. Biswas, "Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks," Proceedings of IEEE Military Communication, Atlantic City, October 2005.
 - [3] History and Social Science Framework. Massachusetts Department of Elementary and Secondary Education, 2018, <https://www.doe.mass.edu/frameworks/hss/2018-12.pdf>
 - [4] English Language Arts and Literacy. Massachusetts Department of Elementary and Secondary Education, 2017, <https://www.doe.mass.edu/frameworks/ela/2017-06.pdf>
- Sources for in-game assets:**
- [5] Typing Minigame's Background: https://unsplash.com/photos/ZPVisr0s_hQ
 - [6] Geography Minigame Images (Wikimedia): [https://commons.wikimedia.org/wiki/File:California_in_United_States_\(US48\).svg#file](https://commons.wikimedia.org/wiki/File:California_in_United_States_(US48).svg#file)
 - [7] Background Music: <https://freesound.org/people/adeluc4/sounds/125347/>

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.