Computer Science and Engineering
University of Nevada, Reno

CT Games

Team 15: Helen Medrano, Erika Manning,
Austin Turner, Mitchell Reyes
Instructors:
Eelke Folmer & Devrin Lee
External advisors:
Eelke Folmer (Game Design) & Shamik Sengupta (Networking)

December 19, 2016

Table of Contents

Abstract

By Erika Manning

Virtual Reality technology is quickly expanding across world markets. Today, VR technology is used in fields such as medical, education, psychology, science and engineering, gaming and more. CT Games aims to create an immersive multiplayer VR game that gives the user a fresh gaming experience that they can share with their friends, and revisit feelings that games gave them as kids.

CT Games' Curse of the Glitchataur™ pits 2-3 players against one in an exciting race to collect keys and escape before the clock runs out, or the solo player, the Minotaur, attacks. Cures of the Glitchataur™ uses VR Step technology to immerse the players in the chase and in-game features such as random item generation and an interactive map keep the players guessing every second. Players must communicate via networked headsets and work together to navigate through the maze, collect keys, outsmart the Minotaur, and find their way out.

Introduction

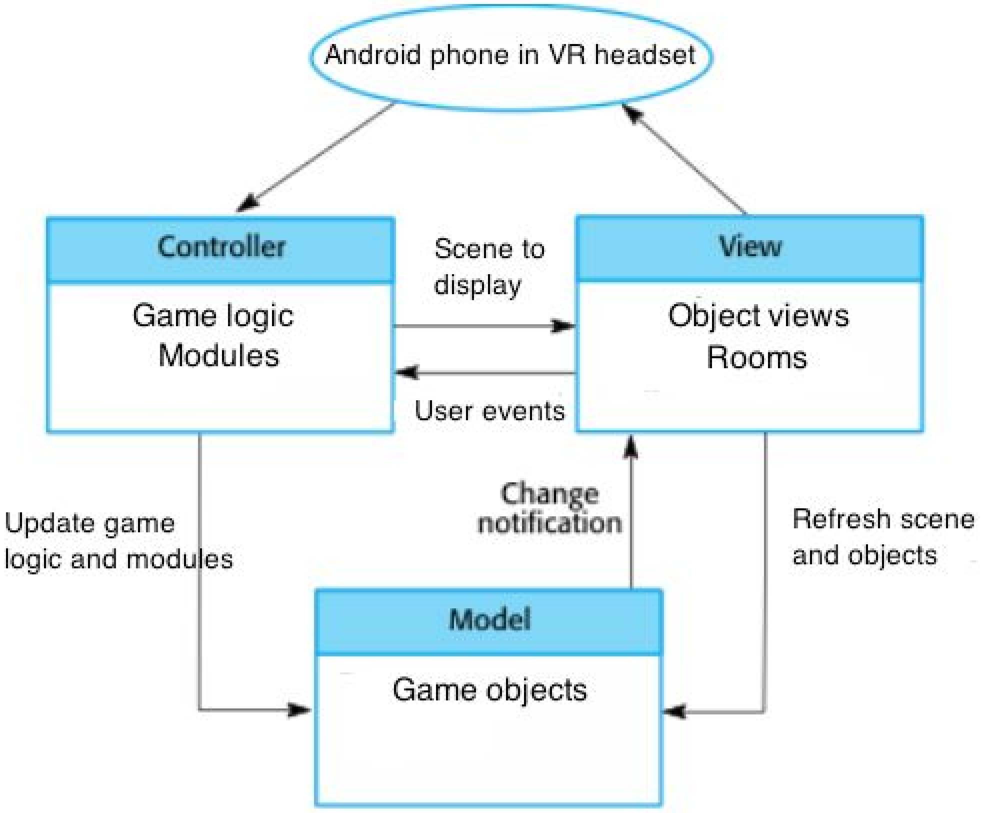
By Austin Turner and Helen Medrano

CT Games is building a Virtual Reality, multiplayer game that focuses on bringing a fully immersive, first person experience to players. The team will be focusing the platform to Android users implementing the VR-Step plugin for player interaction. VR-Step is a unity plugin that allows virtual locomotion through sensing walking in place. The translation of walking in the real world is translated to the virtual environment as real walking is approximated well. VR-Step provides a more simple and immersive experience without any external, bulky equipment. It only relies on the smartphone's inertial sensors to pick up on real-time pedometry. VR-Step also reduces the chances of motion sickness. It creates a hands-free experience and users can still stay in place.

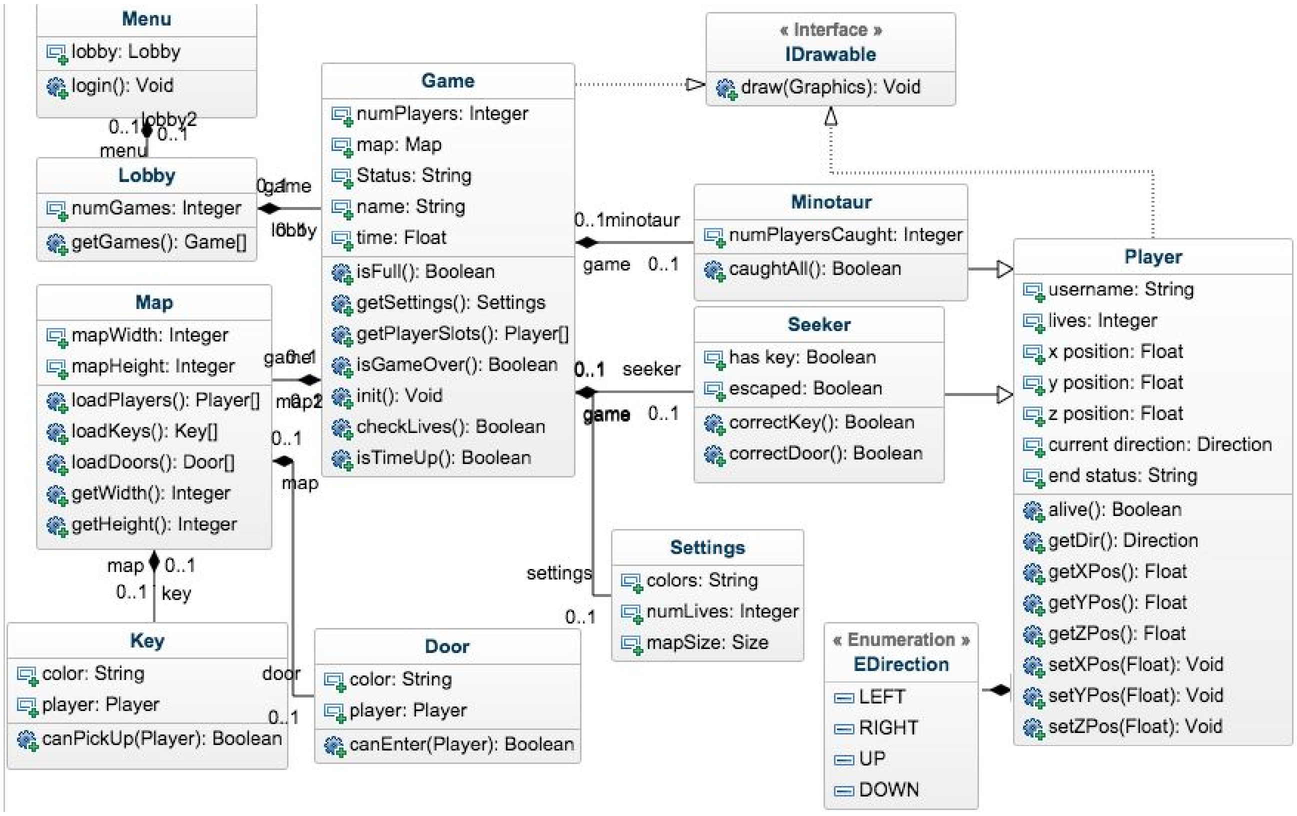
The main selling point of the project, as well as its competitive advantage, is that it implements the capability for multiple players to play the game together in a classic 3v1 style. Since the last report that was published, the gameplay and UI for the game was changed. Instead of a pacman theme, CT Games decided to avoid trademarking conflicts by moving the theme to a dungeon explorer with three players playing against one player who is a Minotaur. The goal of the game is for the human players to explore the map and find keys that are hidden and bring them to the door. On top of that, another player is the Minotaur whose main goal is to find the other players and keep them from finding the keys and bringing them to the door. Players are naturally smaller and quicker than the Minotaur, so hiding and running away are always options. But, when carrying a key, it is so large that they move slower and can easily be caught by the Minotaur. If the players are able to all bring their respective keys to the door before the time runs out, they win. Despite this large change to the gameplay, the overall structure and requirements for completion remain generally the same. CT Games is still going to need a map and player modules to generate for the game to render. Also, the UI flow will be generally the same with some of the options changing slightly. Lastly, the overall structure of hosting the application and enabling networking for our players has not changed at all. With that being said, CT Games has begun thinking about how they want to start tackling the project. Namely, structuring out the workflows, the tools for completion as well as the different parts of the project that are expected of each member to complete.

High-level Design

- **Architectural pattern: Model-View-Controller**



- **Class Diagram**



Classes

1. Player: A player in the game has a username provided when logging in on the menu screen, as well as number of lives, position, direction, and a status at the end of the game.
    a. alive(): Checks if the player is alive
    b. getDir(): returns the Direction of the player
    c. getXPos(): returns player’s x position
    d. getYPos(): returns player’s y position
    e. getZPos(): returns player’s z position
    f. setXPos(float): set player’s x position
    g. setYPos(float): set player’s y position
    h. setZPos(float): set player’s z position
2. EDirection: direction of the player (up, down, left, right).
3. Minotaur: A player that is to catch the Seekers in the game.
    a. caughtAll(): returns true if all players have been tagged by the Minotaur
4. Seeker: A player that is to escape from the maze.
    a. correctKey(): returns true if the correct key is in player’s possession.

   b. correctDoor(): returns true if player has gone to the correct door in the maze to escape.

5. Menu: where the user must login to then go to the lobby
   a. login(): user provides a username and may login if the username has not been taken by players in the games available in the lobby.
6. Lobby: shows all available games in the area.
   a. getGames(): returns all the games in the area along with information about the game. This information includes the number of players, size of the map, and whether or not the game is full.
7. Game: item in the lobby.
   a. isFull(): returns true if there are 4 players already in playing.
   b. getSettings(): returns settings for the game including: color choices for the player, the number of lives a Seeker gets in that particular game, and the size/area of the map.
   c. getPlayerSlots(): returns the players' usernames that are currently in the game.
   d. isGameOver(): returns true if the game is over.
   e. init(): initialize the game by loading everything on the map and starting the clock.
   f. checkLives(): returns true if at least one player is still alive.
   g. isTimeUp(): returns true if the game's time has run out meaning the minotaur has won.
8. Settings: game settings chosen by the admin user of the game in the lobby along with a user browsing including the color player would like to be, number of lives Seekers get in the game, and the size/area of the map.
9. Map: The maze in the game.
   a. loadPlayers(): randomly spawn the players in the map.
   b. loadKeys(): randomly spawn the keys in the map.
   c. loadDoors(): randomly spawn the doors in the map.
   d. getWidth(): return map's width as a float.
   e. getHeight(): return map's height as a float.
10. Key: Item in the map. There is one specific key per player.
    a. canPickUp(Player): returns true if the key belongs to the Seeker.
11. Door:
    a. canEnter(Player): returns true if the door is for the Seeker.
12. IDrawable: interface.
    a. draw(Graphics): this draws the game and map.

Detailed Design

By Mitchell Reyes

Figure 1: Create Lobby button is pressed by the player

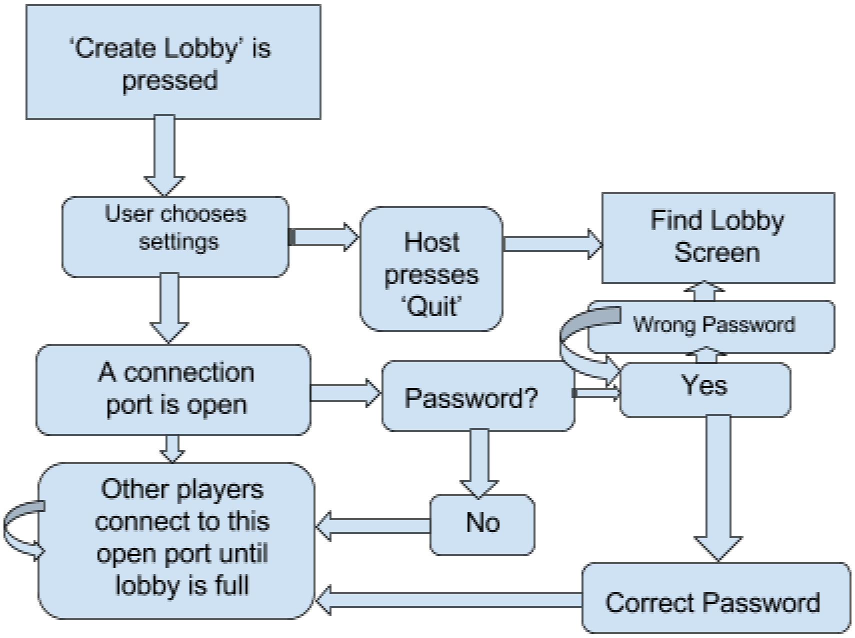

'Create Lobby' is pressed
User chooses settings
Host presses 'Quit'
Find Lobby Screen
Wrong Password
A connection port is open
Password?
Yes
Other players connect to this open port until lobby is full
No
Correct Password

Figure 2: The host starts the game

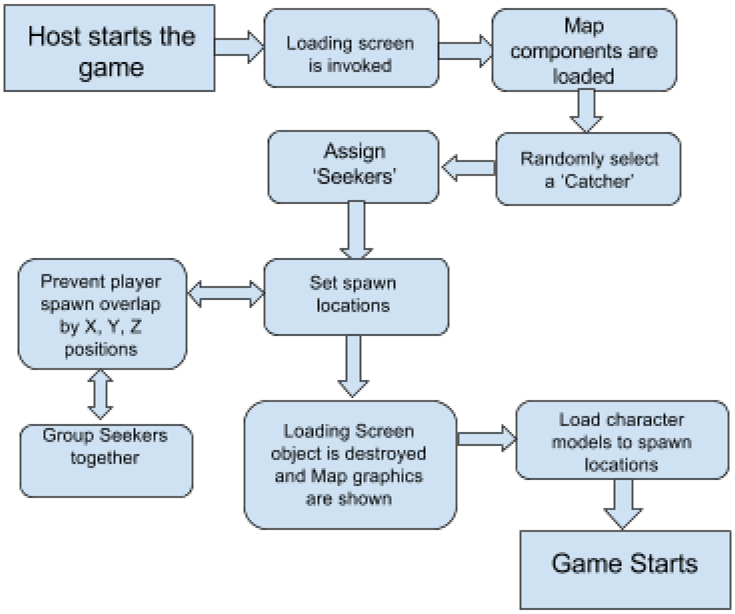

Host starts the game
Loading screen is invoked
Map components are loaded
Assign 'Seekers'
Randomly select a 'Catcher'
Prevent player spawn overlap by X, Y, Z positions
Set spawn locations
Group Seekers together
Loading Screen object is destroyed and Map graphics are shown
Load character models to spawn locations
Game Starts

Figure 3: Player picks up their specified key

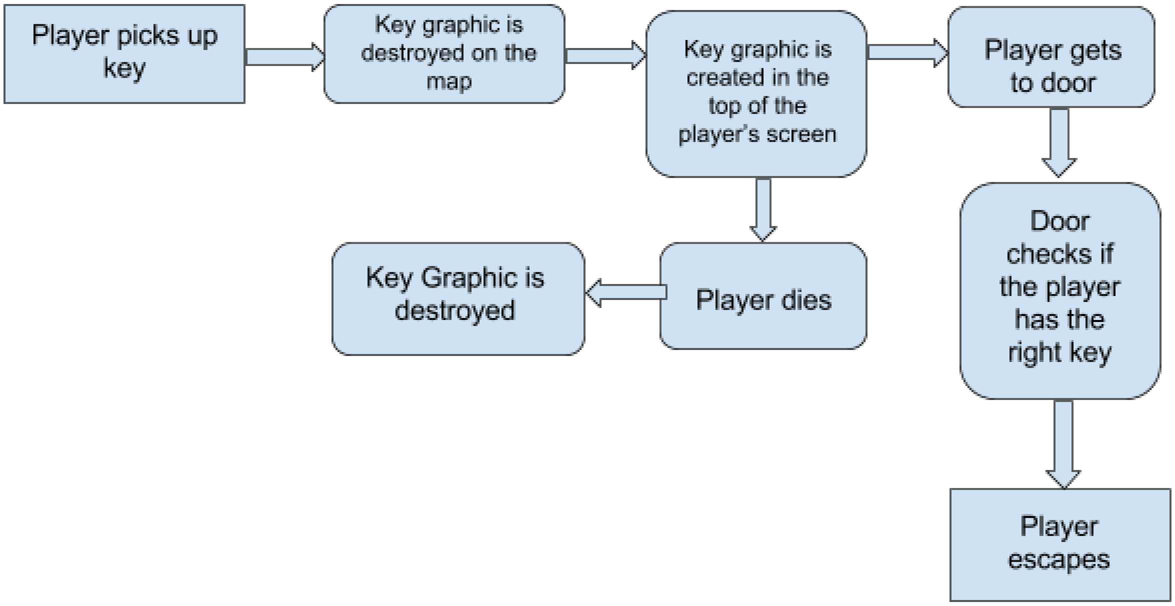
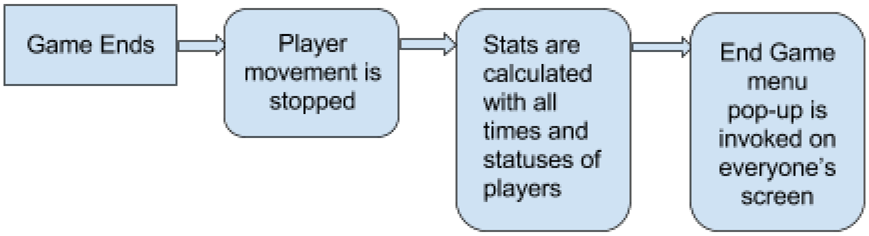

Player picks up key
Key graphic is destroyed on the map
Key graphic is created in the top of the player's screen
Player gets to door
Door checks if the player has the right key
Player escapes
Player dies
Key Graphic is destroyed

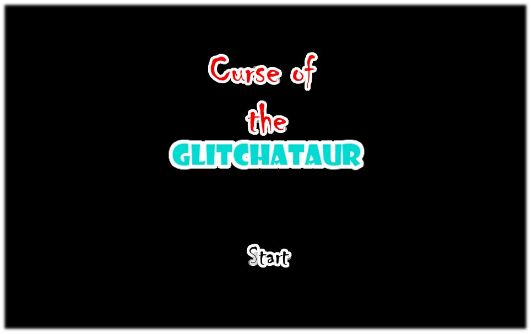
Figure 4: The game ends when the players escape or the time runs out


Game Ends
Player movement is stopped
Stats are calculated with all times and statuses of players
End Game menu pop-up is invoked on everyone's screen

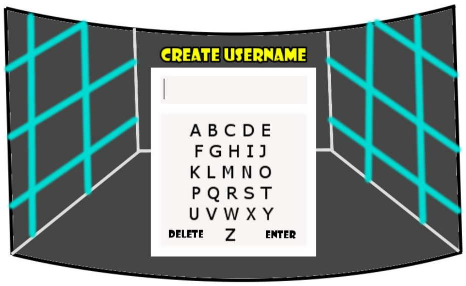
User Interface Design

By Mitchell Reyes

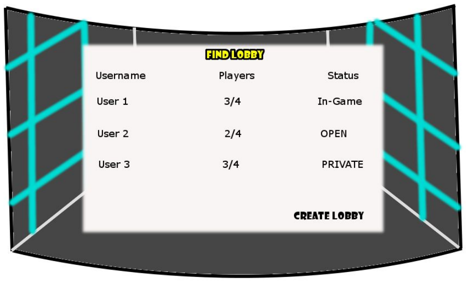
provide at least nine (**<T3>**) or at least twelve (**<T4>**) snapshots of the user interface, with accompanying descriptions. In these snapshots, the main user interface components with details (e.g., panels, toolbars, menus, menu items, buttons, textboxes, etc.) should be presented, and the format used in output results, reports and/or statistics should be shown.



When the player puts the VR headset on, they are prompted with the title screen. On the title screen, they can choose to press start or take the headset off.

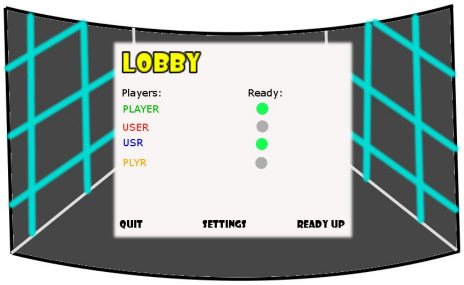


When a user presses ‘Start’ on the main screen, they are prompted to create a username. The username will be at maximum of 8 characters long to reduce the time spent on this screen. Usernames will be erased after exiting the game to decrease data on the servers.


FIND LOBBY

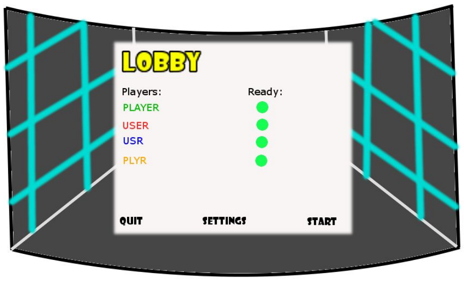
Username	Players	Status
User 1	3/4	In-Game
User 2	2/4	OPEN
User 3	3/4	PRIVATE

After creating a username, the ‘Find Lobby’ menu will be prompted. The user can see the status of other games in the area, along with how many players are in that game. There is also a ‘Create Lobby’ button on this screen, so that the user can open a lobby for other players to join.

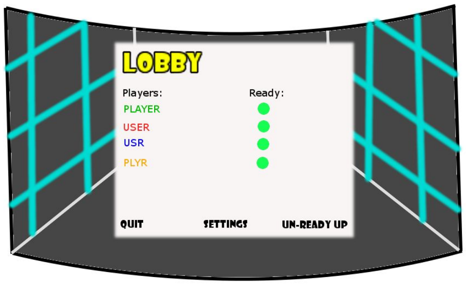

LOBBY

Players:	Ready:
PLAYER	🟢
USER	●
USR	🟢
PLYR	●

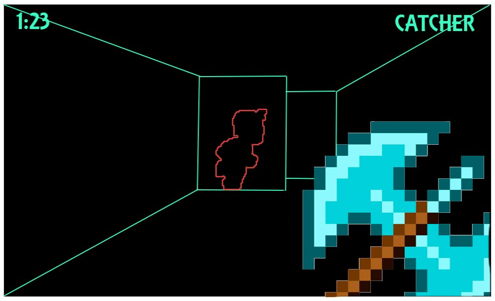
In the lobby, players have the options to ‘Ready Up’, ‘Quit’, or change their player settings. By clicking ‘Ready Up’, a green light will appear next to their name, indicating to the host that they are ready to play the game.



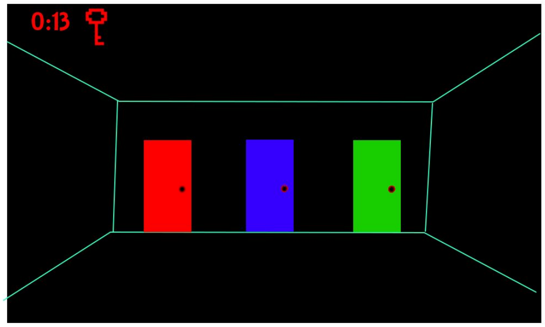
[Host screen] When all players in the lobby have indicated that they are ready, the host's 'Start' button will become enabled, allowing him/her to start the game. If the host selects the 'Quit' option, the lobby will disband and everyone in the lobby will be prompted back to the 'Find Lobby' menu.



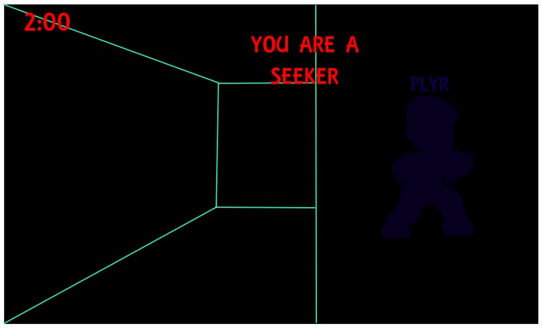
[Player screen] When a player chooses 'Ready-Up', the button changes to 'Un-Ready Up' which gives the player the option to change last minute settings before starting the game. When 'Un-Ready Up' gets selected and the green light goes out next to the player, the button changes back to being 'Ready-Up'.



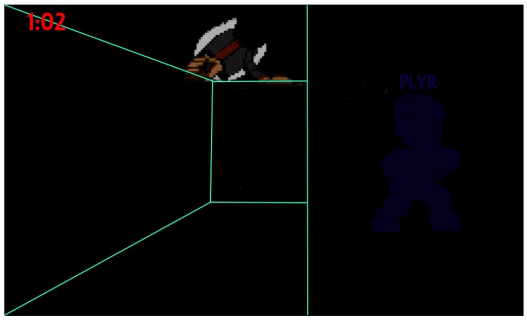
As the 'Catcher', the player is a Cyber Minotaur, the objective is to stop the other players from their objective, to find their key to escape. The 'Catcher' will be able to see the outline of the players through the walls to be able to strategically stop their path.



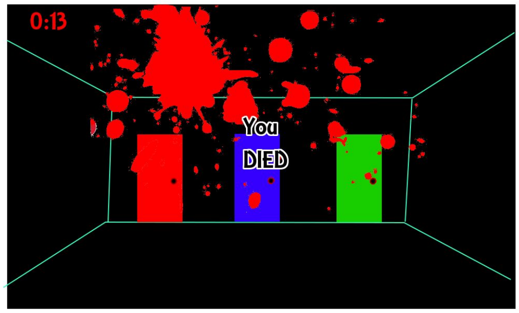
When a player gets their key, they have to go back to their door to escape. They won't be allowed to go through other player's doors or pick up other player's keys.



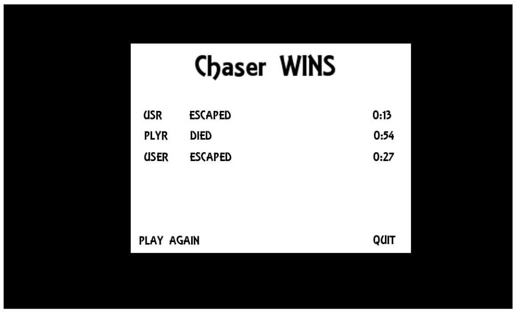
When a player enters the game, the role is randomized to be ‘Seeker’ or ‘Catcher’. If a player becomes a ‘Seeker’, they are on a team of 3 to find their keys to escape.



When the ‘Catcher’ is close, the player will be able to see the top of the Catcher over the wall. This will be the maximum help the Seekers get to avoid the Minotaur.



If a player gets caught, they are brutally killed by the minotaur and prompted to spectate their teammates. When a player dies, the key gets removed from the game so that the other players do not get confused when picking it up.

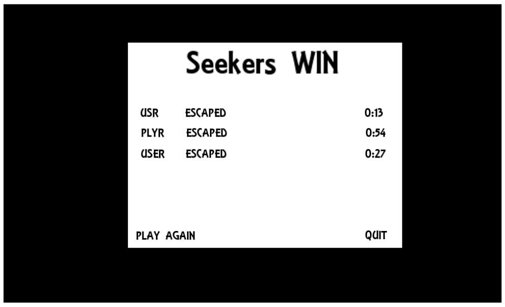


Chaser WINS

USR	ESCAPED	0:13
PLYR	DIED	0:54
USER	ESCAPED	0:27

PLAY AGAIN QUIT

When the Chaser wins, everyone is prompted with a win screen. This menu shows the statistics of each player and gives the options to ‘Play Again’ or ‘Quit’.



Seekers WIN

USR	ESCAPED	0:13
PLYR	ESCAPED	0:54
USER	ESCAPED	0:27

When the Seekers win, everyone is prompted with a menu showing the statistics of each player. The only way for the Seekers to win is for every player to escape.

Annotated References

By Erika Manning and Helen Medrano

1. Make Use Of (website). Retrieved from http://www.makeuseof.com/tag/get-started-making-virtual-reality-games-unity-5-free/
    a. This resource gives a brief walkthrough for setting up Unity and programming with C#. There are also guidelines for how to design a VR game such that performance and user experience is optimized. It provides suggestions leaning towards simplicity in the graphics department in order to reduce uncomfortable effects such as eye strain and motion sickness. The website also gives tips on UI design so that users are not uncomfortable looking at objects and can clearly identify everything in the environment.

2. Unimersiv (website). Retrieved from https://unimersiv.com/how-to-make-a-great-virtual-reality-experience/
    a. This article addresses the intricacies of natural human vision in order to understand how virtual reality can be truly immersive. This source discusses peripheral vision, proprioception, and general aspects to prioritize when designing a virtual reality experience. For example, immersion should be given more importance than creating intricate gameplay. This is what keeps the user interested. Along with immersion, it is important for users to feel that they may appropriately interact with the

environment with a friendly user interface.

3. CNET (website). Retrieved from https://www.cnet.com/special-reports/vr101/
   a. A basic overview of Virtual Reality as it is currently in use and how it will be used in the future. This webpage lists the five best options for Virtual Reality use with links to articles with more information about these options. It also notes the different uses for Virtual Reality now and in the future. This page will be a good go to source for us when looking into different options for devices that can be used. It will serve as a directory for us to look up different information on new developments for VR and current projects that are being undertaken.
4. Folmer, Eelke. “VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments.” CHI’16.

   a. This is the the paper Eelke Folmer, the creator of VR-Step, submitted for an ACM conference. VR-Step is a plugin for Unity that makes walking in place an input to a virtual reality environment. This allows for the use of more immersive experiences when using a smartphone, such as an android, to run a virtual reality program. VR-Step uses the sensors in a smartphone and uses real-time pedometry to simulate natural walking.

5. TechRadar (website). Retrieved from http://www.techradar.com/news/gaming/beyond-oculus-the-future-of-virtual-reality-gaming-1255974
   a. This page goes over smartphone virtual reality gaming. It details improvements that have been made to smartphone software to allow for virtual reality capability. The article outlines the biggest challenge of Virtual Reality products which is just to get people to use them. It underlines the significance of smartphone Virtual Reality games, stating that because of price and accessibility, they are the pathway that people need towards integrating Virtual Reality products into their lives.

6. CNET (website). Retrieved from https://www.cnet.com/news/i-used-my-eyes-to-control-virtual-reality-and-someday-you-can-too/
   a. This articles speaks about a new goal of virtual reality: eye movement to game translation. To make Virtual Reality even more realistic, developers are creating technology that allows the headset or phone to sense the

player's eye movement so that the screen will move according to the players eye movement to simulate looking around in reality. When we create this game, the first version won't be the last. Because of this, we want to make sure we know what advancements are in the near future and prepare our software to be able to handle those changes.

Contributions

Erika Manning created the Cover Page and Table of contents, wrote the Abstract section, and contributed to the Annotated References section, and the Glossary.

Austin Turner did the Introduction, and the system level design flow chart.

Mitchell Reyes created the User Interface and the description for each snapshot. He also created flow charts for the Detailed Design section.

Helen Medrano contributed to the High Level Design section by creating and detailing the class diagram. Helen also updated worked on the Annotated References section, and the Glossary.

Glossary

1. **Virtual Reality:** A virtual 3D environment that simulates interactions for an immersive and realistic experience. This is facilitated through software along with the appropriate hardware/headsets.

2. **Social Virtual Reality:** experience in virtual reality space where users can interact with each other

3. **Head Mounted Display:** hardware that allows for virtual reality experience. In the context of the project, this will be a headset strapped to the head.

4. **Head Tracking:** sensors detect the user's head movement in order to translate this to the virtual reality environment. Thus, the head position in real life will match that being seen through the head mounted display.

5. **Field of View:** The viewing angle in a visual field.

6. **Presence:** The perception by a user that they actually in the virtual reality environment. The realistic feeling that comes from virtual reality.

7. **Plug-In:** a software component that adds functionality to an existing application or piece of software.

8. **Unity:** a tool used to develop video games for multiple platforms such as the following: Android, Apple TV, BlackBerry 10, iOS, Linux, Ninetndo 3DS, OS X, PlayStation 4, PlayStation Vita, Unity Web Player (Facebook included), Wii, Wii U, WIndows Phone 8, WIndows, Xbox 360, and Xbox One.

9. **Judder:** shaking perceived in the virtual environment.

10. **Latency:** This is characterized by the delay between a real life movement and when the visuals adjust to the appropriate scene to sync up with the movement.

11. **Simulation Sickness:** similar to motion sickness while immersed in a virtual reality environment.

12. **Interpupillary Distance:** the distance between the center of the pupils. This is important when designing viewing headsets so that the pupils are properly aligned.

13. **Aspect Ratio:** the ratio of the width to height of an image being viewed.

14. **Haptics:** The feeling of touching something in the virtual reality world.

15. **Immersion:** The user’s reaction to the realism of being in a virtual reality world.

16. **Refresh Rate:** How fast the images are being updated. For example, the number of frames being viewed per second.

17. **VR-Step:** a plugin for Unity that takes walking-in-place as input to a virtual reality environment. This reduces simulation sickness.

18. **Motion Tracking:** recording movements such that they sync up to what is being perceived in the virtual reality environment.

19. **Avatar:** The visual representation of a user in a cyber environment.

20. **Artificial Intelligence:** a simulated entity that appears to have human intelligence such as decision making.

21. **Proprioception:** a sense of the positions of body parts relative to each other as well as awareness of the body's motion.