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 & Shamik Sengupta
November 30, 2016

Table of Contents

Introduction

CT Games will make a virtual reality game using an android smartphone and Unity for programming the game. This means creating a virtual 3D environment that simulates interactions for an immersive and realistic experience with the use of an appropriate headset.

This project aims to showcase all of VR-step's capabilities using Unity to create a Virtual Reality 3D Pacman game with a twist. The game involves a lot of movement, so the VR-Step plugin for Unity is essential in creating a truly immersive experience. Rather than use an external camera or joystick, the game's movement will rely on real-time pedometry. VR-Step 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 game will end when Pacman has either collected all the balls (including those that allow him to eat ghosts) and has returned to the center of the maze or Pacman has lost all lives. There will be interactive map features where using certain items, like springs, will fling players to different parts of the map or even place them on the walls or roof for a different perspective. Mazes will be generated randomly and players spawn in random locations. After a certain amount of time if Pacman has not found any of the ghosts, the ghosts' outlines will be shown to Pacman. However, if Pacman were to find a tracking item, the outlines of the ghosts' will always be shown for a certain duration of time. Pac Man will be able to pick up powerups that allows him to eat ghosts. There will be a lot of control given to the user in terms of game settings where users can select their model color, map election, the area of a maze, the number of lives granted to Pacman, and if the time limit for the game.

Summary of Stakeholders' Interviews

The stakeholders interviewed were Eelke Folmer, Helen Medrano, and Alexander Wittman. Eelke Folmer is an advisor for the project as he is familiar with creating virtual reality games in Unity and is also the creator of VR-Step. Helen Medrano is a software developer for CT Games. The questions were chosen according to the individual's knowledge about virtual reality and programming.

1. How should the VR-step feature be introduced to the user? A tutorial/video in the beginning or should it just be in the description section for the game?

    a. Eelke Folmer: the video for the newer version of Gravity Pull should be readily used to showcase the feature.
    b. Helen Medrano: Perhaps implement a “tutorial mode” that can be switched on or off. In this mode tips and instructions for gameplay appear sporadically to aid the user.
2. What if the user does not *want* to use the VR-step plugin? How should the user then navigate in the game?
    a. Folmer: Users may choose to use a joystick, auto walk, or head tilt.
    b. Helen Medrano: Because users will be expected to constantly be moving in this context it may be appropriate to add an auto walk option. This way users do not feel forced to buy a joystick.
3. Because the game will allow for multiplayer, how much should interaction between players should be allowed? For example, should players be able to add “friends” and chat with them?
    a. Folmer: If it makes sense for the game then it would be a good feature.
    b. Helen Medrano: This should be easy to implement. It could be useful to just provide a chat room for each maze instead of requiring users to create accounts just so they can have “friends” to chat with.
4. How much control should players have over game settings (ex: item type, item amount, level obstacles, number of lives, course difficulty).
    a. Folmer: The game designer(s) should decide over this. The real Pac-Man does not give users control.
    b. Helen Medrano: It may be difficult to give a user this much control because of multiplayer. That is, all users probably won’t agree on how many lives the Pac-Man player should have. Also, course difficulty is hard to gauge. It could be possible to have the user select the *preferred* settings and go from here.
5. Should players be able to level up within a round of gameplay through point collection, kills, and item collection?
    a. Folmer: Is this some sort of RPG game now?
    b. Helen Medrano: This would require players have a steady account and we might just want to implement anonymous logins. That means a user will be able to just open up the game and type in a username and start playing. For players that do want to have a steady account, the win/loss ratio could be saved for the user to view.
6. Should users be able to modify abilities based on picked up items?
    a. Folmer: This may not fit the Pac-Man game as it is real time and users are constantly moving. Would there be time for users to pick things up?

    b. Helen Medrano: In a fast paced game such as this, it could be done in the style of Mario Cart where items randomly appear and enhance the player's ability for a limited amount of time.
7. In what areas does VR the VR experience fall short of the normal TV/monitor gaming experience?
    a. Folmer: VR is more immersive, but can be physically intense so the game should last for 5-10 minutes.
    b. Helen Medrano: VR requires better performance in order to not ruin user experience - especially in a multiplayer setting. Any type of lagging will affect immersion in the game.
8. What complications can arise when creating an interactive map in Unity?
    a. Folmer: This may not be an issue in the future. How is the map interactive (picking up items or moving walls)?
    b. Helen Medrano: The difficulty in implementing a map could be the design. Should the map take over the whole screen or part of a screen? Will it be transparent and visible enough for the user to keep playing or will the user need to stop in place?
9. Should users have a choice to make an account within the game or should they be allowed to be anonymous? What would entice a user to create an account for the game?
    a. Folmer: Implementing player management may not add a lot of value and will make the game harder to program. Might be beneficial to only support two players: one is pacman, the other controls all ghosts.
    b. Helen Medrano: If users want to keep track of something like the win/loss ratio then they may be enticed to create an account. Otherwise, users can just choose a username and begin playing.
10. For a MVP, should we launch with more than one map or should it just have one to show functionality?
    a. Folmer: The real Pac-Man has multiple maps and it shouldn't be hard to create 4-6 maps.
    b. Helen Medrano: Multiple diverse maps should be designed in order to create a more fun experience.
11. H - VR-step calculates a virtual velocity that simulates natural walking. The game requires a lot of walking, so should acceleration be scaled for movement in the game or would this ultimately make the game less immersive and would this induce simulation sickness? Should the users be warned about the high level of walking or would that be unnecessary?
    a. Folmer: Scaling movement would mean moving less precisely. This would be an issue in narrow hallways. This also increases the risk of simulation sickness.

    b. Helen Medrano: The game may become less immersive as the user may start to feel they are gliding rather than walking.
12. What would be the challenges in attempting to implement head tilt detection as a form of navigation to be able to walk in different directions?
    a. Folmer: Head tilt is easy to implement, but if there is a lot of looking around in the game it may generate unwanted movement in a given direction.
    b. Helen Medrano: If a user chooses to move their head frantically there may be significant latency that ruins the virtual reality experience. Maybe just have users be able to do hard left or right turns instead of tracking the head tilt.

Alexander Wittman is a student at the University of Nevada, Reno and is a part of the target demographic, which is young people interested in and already proficient with technology. Alexander is not familiar with programming, but is interested in virtual reality.

1. Are you familiar with VR Technology and products? Rate your familiarity on a scale from 1 to 10.
    a. Alexander Wittmann: Yes. I have used some VR technology before. Familiarity rating: 5.
2. Do you have a way to play VR? Why or why not?
    a. Alexander Wittmann: No. I don't own a VR headset because the ones I could get are expensive.
3. Compared to monitors, why is VR better?
    a. Alexander Wittmann: VR is much more immersive. This attribute is seen all over. In theme parks they use VR on roller coasters to make the rides more immersive as well as develop attractions that are specifically for VR. You can play more immersive games while just sitting in your house.
4. Compared to monitors, why is VR worse?
    a. Alexander Wittmann: There's more of a sense of real camaraderie when playing video games on a monitor next to a friend because you are in and out of the game. This builds better real life relationships.
5. How can VR improve current games that you enjoy?
    a. Alexander Wittmann: VR can add new aspects to the game. It would make me experience all of my games in a more immersive way.
6. What do you look for in a VR Experience?
    a. Alexander Wittmann: I want to see how the features immerse me more than games played on monitors. The more realistic experience, the better.
7. Would you like to play VR Games with friends?
    a. Alexander Wittmann: Yes. All my favorite games are multiplayer and it would be fun to experience them in VR.

8. Do you have health concerns when it comes to VR?
    a. Alexander Wittmann: The VR experience takes people out of reality and as such can affect mental health. There are also physical dangers associated with using VR such as not seeing obstacles around you in the really world. It's also bad to have a screen so close to your eyes.
9. With VR growing, what advancements do you think will be made in the near future?
    a. Alexander Wittmann: They will probably continue to make the experience more and more immersive where the player's body is the controller. More games will be made initially for VR whereas before, games were made for monitors and then used on VR headsets.
10. What limitations should there be to a VR game, if any?
    a. Alexander Wittmann: Alex: Games and headsets should have health and safety warnings because of the risks involved with playing.
11. How do you think VR Games should handle flight simulation?
    a. Alexander Wittmann: There shouldn't be a special way to handle it in a game where flying is just one action because there would be too much moving around. However, in a game that is all about flying, it might be a good idea to create a position in which people can fly such as laying down on one's stomach with both arms forward.
12. Do you like the method of running in place to move a character forward or would you prefer to run on a gaming treadmill?
    a. Alex Wittmann: I would prefer a gaming treadmill because I think that running slowly in place is not as realistic as using a treadmill.

High Level Business Requirements

1. Societal Impact Requirements
    a. Revitalize the multiplayer game interest in society.
2. Monetary Requirements
    a. No monetary requirements as of now.
3. Player Reach
    a. We aim to reach 100 players within the first month of our game being released to the app store.

Technical Requirements

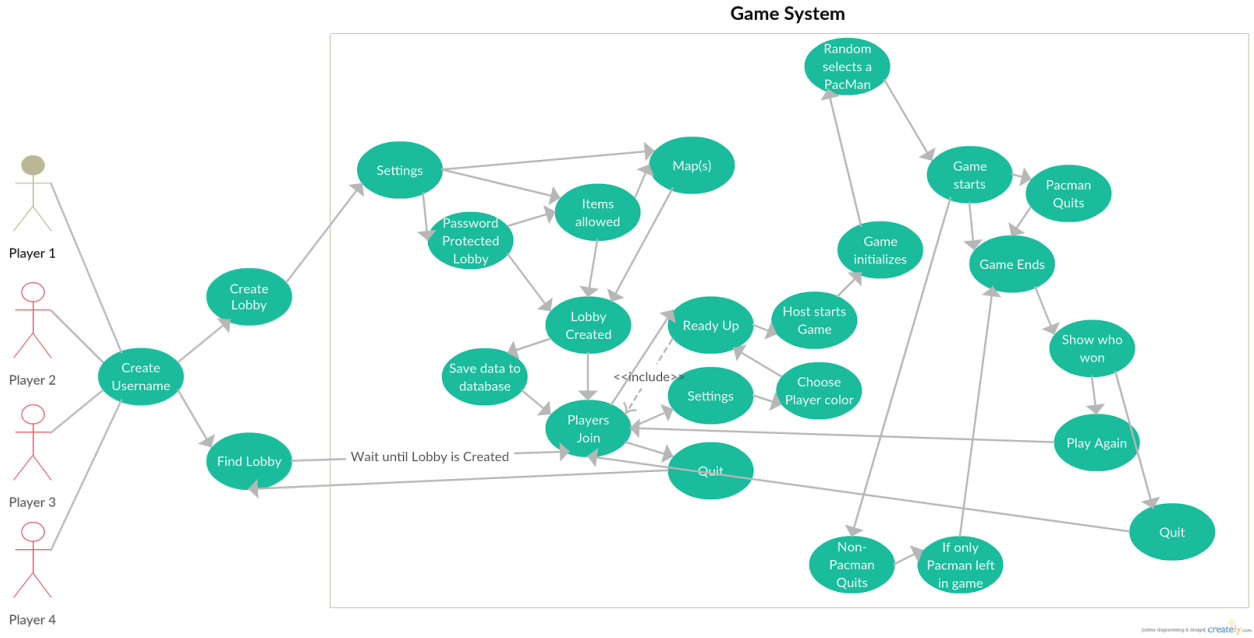
Functional Requirements

1. The game supports multiplayer.
    a. Username
    b. Social Media Credentials (optional for sharing to facebook)
2. The user can choose to play to play against Artificial Intelligence (AI) or other human players.
3. The game ends when either time has run out or the user playing as Pac-Man has collected all pac dots and has returned to the center of the maze. Or when Pac-Man loses all his lives
4. The game has a help document embedded that contains rules for the game, including how to win (FAQ)
5. If the user has made an account within the game, they can view their user stats
    a. Win/loss ratio

Non-functional Requirements

1. Learning curve for the average expected user should be less than 5 minutes.
2. Code should be efficient and strive towards maintainability to handle future changes gracefully
3. Fixed number of users game can support for smooth gameplay
4. Platform: the game should run smoothly in an android environment
5. Reduce the risk of simulation sickness by having an optimal frame rate. It is suggested that the rate should be around 75 frames per second [Make Use Of, website]
6. Handle memory efficiently to reduce the chance of memory leaks.
7. Pause functionality - only available in “story mode”.
8. Each game has a set time limit
9. Coding & documentation standards as listed on the official Unity3D website

Use Case Modeling



Use Case ID	Use Case Name	Use Case Description
UC-01	Create Username	When creating a username, it could be anything that the player wants. Since the names will be temporary, the usernames will be deleted from the database as the player exits the game or goes back to the create username screen.
UC-02	Create Lobby	When a player creates a lobby, this will establish the initial connection to the server. Once the lobby is created, the connection to the

		server will be fully established.
UC-03	Create Lobby Settings	When a lobby is in the stage of creation, the host will have the opportunity to change different settings for the game. Different variations of the game will include password protected lobbies, the different items that will be allowed, and the maps they want to play on.
UC-04	Password protected lobby	If the host wishes only to play with a specified group of people, he/she will be able to set a password to the lobby. When users try and enter the lobby, they will be prompted to enter the password the host has specified.
UC-05	Items allowed	If the host wishes only to play with a specified group of people, he/she will be able to set a password to the lobby. When users try and enter the lobby, they will be prompted to enter the password the host has specified.
UC-06	Choose Map(s)	The amount of maps can be specified when creating the lobby. This will constitute the number of games that will be played. The maps will be pulled from a database and

		randomly assigned to each game played.
UC-07	Lobby Created	Once the host is satisfied with all the settings, the lobby will finish its connection to the server. Once the connection is complete, it will open the port allowing other players to join the lobby.
UC-08	Save data to database	When the lobby is created, this will show the status of the game being played (i.e. In-Game, ¾ Players). This will be beneficial to the user when looking for an open game.
UC-09	Find Lobby	A player is able to find a lobby that is available. This will establish a link to the lobby that was created. Doing so, the player will be able to interact with the other players.
UC-10	Players Join	When the players are joining, their username shows up in the lobby alongside with their player color. The color and username helps distinguish player from player.
UC-11	Ready Up	Players will have the option to Ready Up, indicating to the host that they are ready to play the game. They also

		have the option to un-Ready Up just in case they forgot a setting or are waiting for more players.
UC-12	Player settings	In the Player settings, they have the option of changing their player color. As long as the color has not been taken already, their player color will change to their specification.
UC-13	Player Quit from lobby	If a player decides to quit the lobby, they will be taken back to the Find Lobby screen. When a player quits, everyone else's Ready Up will be "un-Ready up'd", meaning that they will have to Ready Up again.
UC-14	Host starts the game	When everyone is Ready Up'd, the host will have the option to start the game. This will do the initialization of the game to set up for the players entrance on the map.
UC-15	Game initializes	When the game initializes, it will pull the host preferences settings and set up the game variables to match the settings. This will, in-turn, be the loading screen of the game.

UC-16	Random selects a pacman player	Along with initializing the settings of the game, it will also randomly select one of the players to be “it”, or Pac-Man, for the round. The other players will then get defaulted to ghosts.
UC-17	Game Starts	Players get loaded into the game, making sure they they have the username they chose over their head and that they are the color they chose (this applies to ghosts only). The ghosts will spawn in the center of the map and Pacman will spawn randomly in the map, this will ensure no unfair spawns.
UC-18	Non-Pacman player quits	During the duration of the game, if a ghost quits the game will continue unless that there are no ghosts left in the game after quitting, which will end the game. This will make sure that everyone has a fair game.
UC-19	Pacman quits	If Pacman quits, then the game will end. This will default to the ghosts winning.
UC-20	Show who won	Once the game ends, a dialog showing who won will appear on each player’s screen. This will indicate that the game is

		over and they will be given options to play again or quit.
UC-21	Play Again	If the majority vote of the players go to Play Again, all the players will be returned to the Lobby where they have to Ready Up again.
UC-22	Quit	If the majority wish to Quit, then all the players will be returned to the Find Lobby screen. Here they will have the option to go back and change their username or just quit playing.

Use Case: Ready Up	
Use Case ID	UC-11
Actor	Player
Precondition(s)	1. Player must be in the lobby
Flow of Events	1. Player pressed the Ready Up button 2. Circle next to player lights up to green
Postcondition(s)	1. Player is ready

Use Case: Player quits the lobby	
Use Case ID	UC-13
Actor	Player
Precondition(s)	1. Player must be in the lobby
Flow of Events	1. Player presses the quit button 2. Player is removed from the lobby list

	3. Lobby waits until the required amount of players is filled 4. Player is taken back to the find a lobby screen
Postcondition(s)	1. Player is in the find a lobby screen

Use Case: Pacman Quits	
Use Case ID	UC-19
Actor	Player(s)
Precondition(s)	1. The user is in game
Flow of Events	1. The user presses the button on the side of the headset, bringing up the options menu in game. 2. The user presses the quit button on the menu. 3. If the player is pacman, the game ends. 4. The game brings up the End Game screen of who won. 5. The ghosts default to win 6. The options to play again or quit will be shown
Postcondition(s)	1. The pacman player is removed from the game.

Requirement Traceability Matrix

Function Requirements	Design Specification	Test Cases
Players can connect	Players will establish a handshake with the main server	13.i, 13.iii
Players will be	Users will be presented with three forms, the input will	1.i - 1.iii

given a login page with an anonymous user option, and a login for registered users	be validated upon sending to server.	
Players land on the main page	Users will have two buttons one labeled as “Play” and the other labeled as “Settings”. These buttons, when pressed redirect the user to the corresponding page by rendering its content	2.i.1 - 2.i.3
Players can view settings	Upon rendering settings page, the user is presented with with different options delimited in dropdown menus. Upon clicking on setting, the user selects icons or checkboxes to check and uncheck options. When an option is checked / unchecked, the change is saved to the users object.	2.ii.1 - 2.ii.3
Players can select a lobby	Upon validating the user and rendering the lobby action, the user waits in the lobby until a game is arranged. They are queued into rooms until the room is filled	2.iii.1 - 2.iii.5
Players can indicate they are ready	When a user is matched with a room, they are immediately prompted to show they are ready. This will show a button on the room status screen that allows the user to indicate they are ready	2.i.1, 2.i.3, 3.i.1-3.i.4
Players can quit lobby	On the room status screen, a user can cancel their queue at any time, including when a ready signal is given. This button will immediately cancel a request to join a room and will reallocate their position to another player.	2.i.1, 2.i.3, 3.ii.1-3.ii.5
Players can change settings in lobby	While waiting for a game, the player has access to their model color while queuing. This can be selected by tapping the color that they choose. The setting saves to the user object upon selection	3.iii.1.a - 3.iii.1.c
Upon all players accepting, the game initializes	When all players have sent their ready signals, the server will commence the initialization process. This will entail: loading the map that the players queued for, the ghost models, the pacman model, the items and power up models, the UI for rendering objects along with correctly placing them on the map. On top of that,	4.i - 4.iii, 4.iv.1 - 4.iv.4

	the server will send out synchronization handshake requests to the proper parties through a 3-way handshake.	
Server generates map for users	Upon the proper rendering of objects, the server begins the process of correctly placing the models on the map as well as generating the UI. Users will be correctly placed and the connection will be secured through a UDP transfer of data with a request for acknowledgement done in periodical increments	5.i - 5.v
One player gets chosen to be pac man	Before the game map is rendered, one player, chosen at random, plays as pacman. This will render their model correctly and rener the other players as ghosts. The ghosts are spawn in the center while Pac Man is spawned in a random corner of the map.	6.i - 6.vi
Players can move along the map	Using the VR step plugin, along with the input from the user's phone through the google cardboard hardware, players have the ability to change their positions in the map and have their view changed according to what is in their field of view.	7.i.1 - 7.i.3, 7.i.3.a, 7.ii
Ghosts can catch Pac Man	When a ghost and Pac Man enter a predetermined area of space, they activate a collision check. If Pac Man has no power up, then he loses the collision, loses a life and is sent to a random corner of the map. The Ghost receives points	7.v.1-7.v.3
Players can pick up items	Pac man has the ability to pick up powerups and items that help him. One powerup, upon pickup changes the ghost model to another that allows him to collide with Ghosts and win. When a ghost is hit by a powered up pacman, he is sent to the center of the map. The Pac Man gains points	7.vi.1-7.vi-3, 7.vi.4.a-7.vi.4.c ,7.vi.5.a, 7.vi.6
The Game can end	When Pac Man runs out of lives, or the time runs out, the game ends and all players and processes associated with the game come to an end	8.i - 8.iii
The game displays who won and the statistics that go along with the game	When the game enters in the end game screen, it will calculate all the various statistics that were gathered in game. These statistics include; time elapsed, ending condition, which team won, points organized in rankings,	9.i - 9.iii

Give users an easy way to re-play	Users will be given an easy way to requeue into the lobby through the ending screen. This will be handled through a button that triggers the server to begin the requeue process	10.i.1 - 10.i-3
Give users a safe way to quit game	A user can quit the game in any numerous ways. First, they can leave correctly through the U.I. this will prompt them for a confirmation and log them out before leaving the application and closing down all processes. Second, a user can leave by leaving the game's sandbox, this will hold the user's session in limbo. If a user is in a game, then they will still be receiving packets until a timeout is reached. Otherwise, the users session will be held until they close the application.	11.i - 11.iii

Test Cases:

1. Create Username / Login
    i. Can connect to social media
    ii. Can validate a username and password
    iii. Upon creating permanent account, username and password are validated for
        1. Uniqueness
        2. Length
        3. Lack of profanity
2. Create Lobby
    i. Buttons
        1. Can be pressed
        2. Renders correct view
        3. Are visible
    ii. Settings area
        1. Password protected lobby
            a. User is validated upon entering
            b. Redirects to login if no username and password
            c. Redirects to login if username and password aren’t validated
        2. Items wanted
            a. Items to select from aren’t nil
            b. Handles no selected items
            c. Handles any number of items
            d. Saves changes in items to user object
        3. Maps

a. Map must be selected
b. Gives validation error on no map selected
c. Maps exist and can be reached
d. Saves users changes to object

iii. Find Lobby area
1. Queues user on button press
2. Handles no users
3. Handles no open lobbies
4. Shows user other connected players
5. Handles mix of LAN users and online users if needed

3. Players join
i. Ready up
1. Handles ready accept
2. Handles no ready accept
3. Handles no users ready accept
4. Handles all users ready accept

ii. Quit
1. Handles no users pressing quit
2. Handles all users pressing quit
3. Handles one user pressing quit
4. On quit, dequeues player and open seat to another user
5. Redirects user back to main screen

iii. Settings
1. Color select
a. Has default value
b. Has values to select from
c. Saves choice for later as player default

4. Host starts
i. Loads map
ii. Loads models
iii. Loads UI
iv. Establishes connection
1. Handles user not connecting
2. Handles no users connecting
3. Handles all users connecting
4. Establishes 3-way handshake

5. Game initializes and starts
i. Loads items on map
ii. Loads player on map

iii. Doesn't spawn items in walls
iv. Doesn't spawn players in walls

6. Game random selects a Pac-Man, the other players are ghosts
    i. Player chosen to be Pac-Man is random
    ii. Only one Pac-Man
    iii. Ghosts spawn in center of map
    iv. Pac-Man spawns in a random corner
    v. Players have their username above their heads
    vi. Players have their chosen color
7. Game Starts
    i. Handles movement along the map
        1. Pac-Man can't enter center
        2. Ghosts can reenter center
        3. Ghosts have to leave center after a time period
            a. If they don't, they get pushed out
    ii. Players can't pass through walls
    iii. Players can pick up items
    iv. Players can traverse obstacles
    v. Ghosts can hit Pac-Man
        1. Pac-Man loses 1 life
        2. Ghost who collided gets points
        3. Pac-Man is replaced to a random corner
    vi. Pac-Man can pick up powerups
        1. Pac-Man can pick up
        2. Handles changing colors
        3. Handles locating ghosts
        4. Handles new Pac-Man-ghost collision
            a. On collision, ghost dissapears
            b. Pac-Man gets points
            c. Ghost get sent back to center
        5. Powerup lasts a set time
            a. Powerup handles any time period
        6. Ghosts can see time remaining on powerup
8. Game Ends
    i. Players stop moving
    ii. Any queued points are tallied
    iii. Handles queued processes
9. Show who won
    i. Handles Ties

    ii. Handles correct comparisons
    iii. Shows all statistics for the game
10. Play Again
    i. Handles one player going back
    ii. Handles all players going back
    iii. Handles no player going back.
11. Quit
    i. Handles user leaving
    ii. Handles a user leaving application without closing
    iii. Handles user leaving mid game through timeouts
12. Player leaving prematurely / disconnecting
    i. Checks for timeout
    ii. Checks for quit pressed
    iii. Handles finding new player in time
    iv. Handles reassigning Pac-Man
    v. Handles one user leaving
    vi. Handles all users leaving
13. Networking
    i. Establish 3 way handshake
    ii. Handles lost packet
    iii. Handles synchronous communication
    iv. Handles loss of communication
    v. Handles full lobbies
    vi. Handles full queue
    vii. Handles ip change

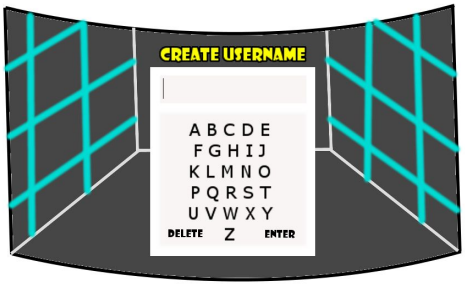
Potential Legal Issues

Because this game is a Pac-Man clone with a few twists, we must be careful how we go about design. It is generally okay to develop a game with similar mechanics/rules to another game as gameplay is *usually* not patented, but the presentation and name of the game can become an issue. For example, representing a character as a yellow circle with a triangle cut out and giving it a different name resembles the Pac-Man character too much. This goes for the ghosts characters as well. Although, there is less of a risk when a game is not distributed for profit. Developing a free game that clones another is still not legal, but there is less of a chance for legal problems to arise.

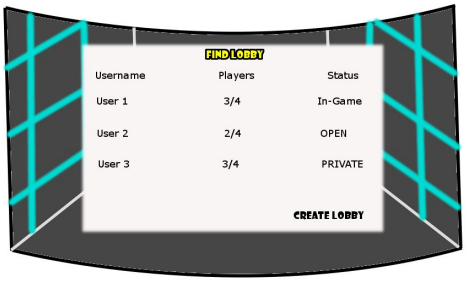
Copyright protection for Pac-Man applies to the source code, artwork, and sounds. Thus, creating a maze game where a user eats dots while being chased by enemies is not protected.

Initial Snapshots

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.



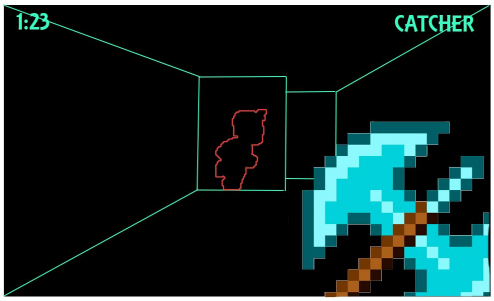
After creating a username, the ‘Find Lobby’ menu will be prompted. The user can see the status of other games in the area, along with how many players are in that game. There is also a ‘Create Lobby’ button on this screen,

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.

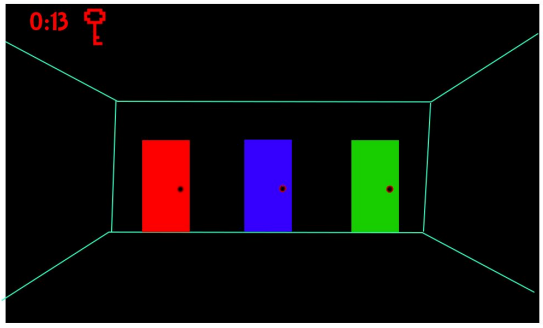
so that the user can open a lobby for other players to join.

[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.

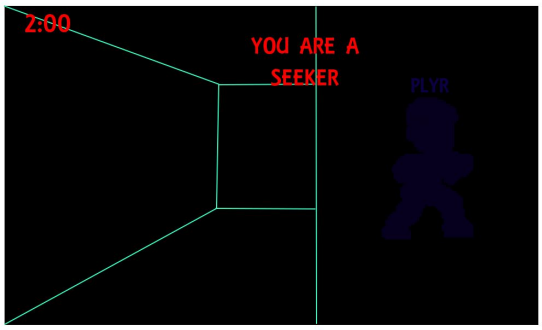
[Player screen] When a player chooses 'Ready-Up', the button changes to 'Un-Ready Up' which gives the player the option to change last minute settings before starting the game. When 'Un-Ready Up' gets selected and the green light goes out next to the player, the button changes back to being 'Ready-Up'.



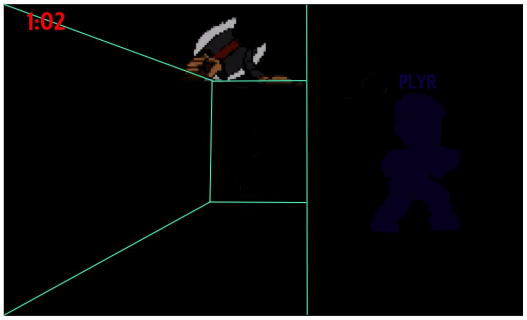
As the 'Catcher', the player is a Cyber Minotaur, the objective is to stop the other players from their objective, to find their key to escape. The 'Catcher' will be able to see the outline of the players through the walls to be able to strategically stop their path.



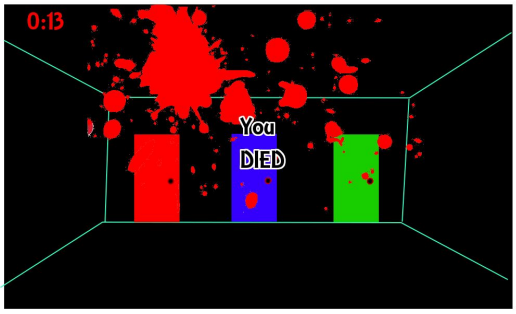
When a player gets their key, they have to go back to their door to escape. They won't be allowed to go through other player's doors or pick up other player's keys.



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.

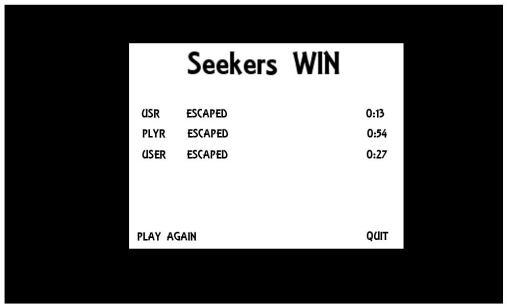


When the ‘Catcher’ is close, the player will be able to see the top of the Catcher over the wall. This will be the maximum help the Seekers get to avoid the Minotaur.



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.

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’.



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.

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.

List of References

1. Make Use Of (website). Retrieved from

   http://www.makeuseof.com/tag/get-started-making-virtual-reality-games-unity-5-free/

   This resource provides 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/

   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. E - CNET (website). Retrieved from https://www.cnet.com/special-reports/vr101/

   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.

   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

   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. E - CNET (website). Retrieved from https://www.cnet.com/news/i-used-my-eyes-to-control-virtual-reality-and-someday-you-can-too/

   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

1. Erika Manning contributed to the following sections: Vocabulary, Interview Questions, Reference Section, Interview Q&A, and High Level Business Requirements.
2. Helen Medrano contributed to the following sections: Vocabulary, Interview Questions, Reference Section, Interview Q&A, Introduction.
3. Austin Turner contributed to the following sections: Introduction and Requirement Traceability Matrix.
4. Mitchell Reyes contributed to the following sections: Snapshots and Use Case Diagram.