



**University of Arkansas – CSCE Department
Capstone II – Final Report– Spring 2020**

The Maze Master

**Nick Davis, Maharshi Dubasi, Jose Garcia, Dale Hagen, Jaylen Kimbrough,
and Alec Ray**

Abstract

Plenty of genres of video games have advanced exponentially within the last decade in regards to story, graphics, and interpersonal connectivity. However, one of the genres left behind are maze-type games. Games like *Doom* have been removed as a game that's meant to be emulated and replaced with *GTA V*. Games are moving from a closed and controlled environment to an open-world and less predictable environment. This means that games with mazes are now overlooked and are often less designed and under-developed. All of the games found on the app-store relating to maze games are either dull and boring or puzzle games.

The objective of our game is to create a game that changes the perspective on maze games. Typically these games are single-player and doesn't get the player involved. Using VR technology, it will become easy to instantly involve the player in the game they're experiencing. This approach will tie in a newly emerging technology along with interesting multiplayer gameplay. This will change the perception that maze-games are boring and trend toward a newer style of video game design.

1.0 Problem

Most maze games available on the market have a tendency to be repetitive and rehash what has already been done. This includes the horror genre of maze and the classic 2D maze.

A large number of mazes have the main character proceed and rely on their speed or wits to outrun the clock or an enemy, the horror genre of maze. An example of this that has been done well is the game *Dark Deception* by Glowstick Entertainment[1], which utilizes a small series of premade mazes and creatures to hunt you in them. Mazes like this leave the player with no option other than stealth to escape their pursuers. While this can be entertaining at first, it can leave the user bored as they have no reason to return to reconquer completed mazes. The end result is a game whose lifetime ends after a 1 or 2 playthroughs.

The other type of maze that most people are probably familiar with is the classic 2D maze. There exist many iterations of this maze utilizing either the introduction of digitization, such as *aMAZE 2* by Blender Games[3], to the use of an elevated overview to start before delving into the maze below as in *Maze of Gaea* by TiStar Game[2]. This type of maze game tends to rely on its visual

aspects to draw in players which results in its lifespan being dependent on the variety it can provide and how pleasing the aesthetic is to the player.

2.0 Objective

The objective of this project is to create a 3D maze runner game and give the user a dynamic and immersive experience they might not get to have with other maze games that are on the market today.

3.0 Background

3.1 Key Concepts

There are several technologies related to the project, the first one we will be using is called Unity. Specifically, we will be using Unity3D for all of the modeling for the maps along with the design for the minotaur chasing you. Unity has built in software that makes it easy to edit, and a nice game engine. Essentially it's an all-in-one editor that helps us with creating NPCs to chase you, and also to edit graphics that might be used in the game. It is able to use different languages, but mainly we will be using C# for the coding side of the game.

Another technology that will need to be used in this is the Oculus VR headset. This will be the main platform on which the user will be able to play the game on. The headset will allow the player to fully immerse themselves into the game, making it feel as if they were inside the maze running for their lives. The headset comes along with two handheld controllers that will be used to help the user navigate through the maze.

The next technology we will be likely using is Github. Github is a place to publish code and to also make it easier for everyone to work on in a group project. With working in a group, it's hard for everyone to have the same exact code to be working on at the same time. With Github, the coders are able to update every line of code they add, so that way everyone in the group will have the most current code. This will help cut down on bugs being produced, and also developers not having to worry they are working on code that might have been previously developed already.

The last technology we will be using is an app called Trello. Trello is designed for developers like us that will be taking on tasks of creating projects. It's an easy interface that allows us to create cards of different tasks that we will need to be working on. The page is split up into different sections, allowing us to move the cards into the sections specifically. The sections will likely be split up into in development, testing, and complete. Once our team creates all the cards, it allows us to assign cards to different users, giving them different tasks to complete. Overall it's an easy way for a team to be familiar with what needs to be done, and what has been done in the project.

3.2 Related Work

Other researchers and developers have accomplished many things in this project area. Some things that we want to address from the others is that most maze games grab your attention for

only a few play throughs and then you might not feel too interested in playing the game again[1]. The games feel slightly repetitive and aren't very engaging with the users[2]. Maze games generally have a basic map design and don't really make the user feel like they are in a maze[3]. A couple of different things that we are going to do to make ours different is first, we will place the player in an actual maze. Rather than placing the player with an aerial view like the majority of other games. Second is for us to make it engaging. We will accomplish this by implementing true lighting in the maps, and a minotaur chasing them through several different levels. Eventually, we would like to make the map dynamically adapt to the players actions, making the maps dynamic to each player and giving them a unique experience.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

For this project, we want to make the game more involved and able to create an interaction between player and game. This is mostly seen with the monster NPC, as it will be moving within the maze and trying to catch the player, which will cause a game over. The game will also be using sound location to tell you whether the monster will be close or far away from you. The other pieces are the items that players receive. The items, a torch or a sword, will allow an interaction between player and environment, which, depending on the item chosen, can either help or harm the player in its own way. With that, we will also need to create models for the player, walls, and NPC Monster. This way the player can tell what an object is.

The project will also need to be able to have a multiplayer functionality to where players can interact with other players in a competitive manner. These interactions will be in competitive racing game mode with the players seeing who can make it to the end of the maze first. We also want to make this game VR compatible instead of the basic 3D environment on television. With this, we can have the player jump in and experience the feel of the game while looking into the game.

We'll also set up a simple UI for this to allow people to jump in and be able to have fun without worrying about how controls work. This project will only have two items in game, the torch and sword, which will give the player an option for either protection or more viewing range. The last thing we will be implementing is multiple game modes for this game. This game will have a regular maze mode where the entire maze is generated in the beginning and you will have to get to the end of it before the monster gets to you. There will also be a mirror mode, where the walls are replaced with mirrors and can lead to more difficulty in movement and location. The last mode is insanity mode, where we will implement a dynamically changing maze that generates a new block every time you step into a new area.

4.2 Architecture

This project will have a maze like functionality such as the examples seen below. The environment will be considered dark due to the game being a more horror genre than a regular maze game. The monster will be around in the game but it will pop up at certain points in the maze. You can see this from the graphic that was made below, showing the minotaur in the background. The player will have a hand that will be holding either a torch or a sword, which will either give defense to the player(sword), or give more field of view to the player(torch). The walls will look more deteriorating than what is seen in the image and will have slightly more detail on them. The view of the maze will be first person in a Doom-like fashion, where everything will be around you instead of it being top down like most maze game formats. The maze itself will consist of a lot of turns and halls that will confuse the player on where the monster is and also where the exit will be as well. There will also be a sound notification system on the maze so you can hear how close the minotaur is and be able to pick which direction you will have to take to not be face-to-face with the creature chasing you.

There will also be two extra game modes in this game that the player can use to interact with the maze. The Mirror mode is a game mode that replaces all the walls with mirrors and have the monster appear in the mirror, but you can't tell whether or not the monster is actually there or if it's the actual creature. There will also be an Insanity mode, where the maze dynamically generates a new path each time you move up a space. This means that if you turn left 3 times in the maze, you won't end up in the same room you started at.

We will also be looking at a multiplayer functionality for the game that does a time based competition for the players interacting with each other. These characters will be racing with each other to see who will be the first to reach the end of the maze or the last one surviving the maze. As stated earlier, we have a figure that we made to show how the setting of the maze is going to be set up and what the environment will more or less look like by the final product(first beta).



Figure 1 shows the rendering of the maze game (Work in Progress.)

4.3 Risks

Risk	Risk Reduction
Making a poor decision about which platform/engine we choose	We know the functionality that we want in our game, so we have to do research and choose a game engine to develop in that can handle all of our capabilities, and not cut our game off short.
Spending too much time on the little details such as the art of the game	We all want to play a game that is visually pleasing, has a good soundtrack, etc. Art, for example, is an important feature of games, but we have to make sure that our components work and our game is completed first because the little details can take up a lot of time. We have to build the skeleton of the game then start adding all the details to make it attractive.
Making an addicting game that can harm the health of players	Game addiction is a big issue today. We have to do research to see what the current non addicting games are doing versus what the addicting games are doing to avoid developing a game that can be detrimental to a user's health.
Not overscoping a prototype so that we can finish our project	Games involve a lot of time. We have to properly prototype our project, so that we can finish it on time with all its components working. We have to realize that we aren't making the next "Call of Duty" or "Fifa" so we have to work accordingly since we only have 1 semester to finish it.
Defining proper mechanics that a user can easily understand	Rules are part of every game. Sometimes, they can be very confusing which can be frustrating on the user. We have to make sure to set proper mechanics that don't have loopholes, make those mechanics clear to the user, and test them out to ensure that everything works as expected.
Not testing our game before submitting	Testing is an important part of the game development process. As we add scripts, change functionality, add detailing, etc, we have to test everything to make sure our game is working properly. We plan we have everyone on the team as well as outsiders test our game to be able to catch bugs before we submit our game.
Making a game that users will enjoy	We want a game that people enjoy. We have decided to make a maze runner game because there aren't many on the market and the ones that are there aren't good since they are very repetitive. We have to do research to make a game that is fun for users to play.
Having members uncomfortable in the technologies that this game will use	Depending on the game engine that we use, we have to learn how to use its features, libraries, etc. For example, Unity uses C# for its scripts, so we all have to learn that. We also have to know how to use Unity in general such as testing, manipulating values, adding art, etc. We have to learn all this,

	and get all team members comfortable in the engine that we choose to be able to make progress.
--	--

4.4 Tasks

1. Research game engines and choose one that supports VR game development
 - Must be compatible with Oculus VR headset
 - Must allow team development
 - Must be in budget (under \$50)
2. Get familiar with game engine. Watch youtube videos, do follow along tutorials, read game engine documentation, look at similar projects, etc. At this stage, we want to dive into the game engine to see how things work.
3. Develop a Game Design Document (GDD) that lists the game mechanics, interface and heads up display, levels and missions, narrative, art such as our color scheme, game's tone, mood, imagery, etc. Things to think about below
 - What controls will the player use to operate the game?
 - What can the player do in the game?
 - What are under the hood mechanics?
 - What is the story of the game and what is the end goal?
4. Create tasks on Trello that have to be done, and assign to each of the team members. For example, in assigning scripts, one team member will focus on creating the maze while another will focus on the health of the character.
5. Set up a GitHub repository for the team to be able to contribute to the project. We will all need to set up a branch and merge to the master branch when all of our components are working.
6. Implement the tasks that were created in Task 3. Continue to reach out to resources to be able to complete the project.
 - Make sure to test when everytime code is added/modified
 - Upload code to GitHub
 - When everything "completed," merge all code to master branch
7. Test our game using the Oculus VR
 - Have everyone in our group play, test, and find issues
 - Bring in people that aren't in our group to find bugs and/or give us suggestions about flaws in our game
8. Fix bugs and work on the flaws of our game
9. Work on the little details of the project
 - Use software such as ProBuilder Tools or Blender to create the art for our game
 - Download and import audio into our game to make it entertaining

10. Again, test and make sure all the components are working properly like they were before the addition of art, audio, etc.
11. Write the final document for the game, and submit it.

4.5 Schedule

Tasks	Dates
1. Research	<ul style="list-style-type: none"> ● January 13th - 20th <ul style="list-style-type: none"> ○ One Week
2. Gain Familiarity with Game Engine and tools	<ul style="list-style-type: none"> ● January 21st - February 4th <ul style="list-style-type: none"> ○ Two Weeks
3. Develop a Game Design Document (GDD)	<ul style="list-style-type: none"> ● February 5th - 12th <ul style="list-style-type: none"> ○ One Week
4. Create and Assign Tasks	<ul style="list-style-type: none"> ● February 13th - 20th <ul style="list-style-type: none"> ○ One Week
5. Setup a GitHub Repository	<ul style="list-style-type: none"> ● February 13th
6. Implement the Tasks	<ul style="list-style-type: none"> ● February 21st - March 21st <ul style="list-style-type: none"> ○ One Month
7. Testing	<ul style="list-style-type: none"> ● March 22nd - March 25th <ul style="list-style-type: none"> ○ Three Days
8. Bug Testing and Fixing	<ul style="list-style-type: none"> ● Will be done throughout tasks 6-10
9. Polish the Game	<ul style="list-style-type: none"> ● March 26th - April 8th <ul style="list-style-type: none"> ○ Two Weeks
10. Final Testing	<ul style="list-style-type: none"> ● April 9th - April 12th <ul style="list-style-type: none"> ○ Three Days
11. Write the Final Document	<ul style="list-style-type: none"> ● April 13th - April 20th <ul style="list-style-type: none"> ○ One Week

4.6 Deliverables

- Game Design Document: Lists the games major mechanics, interface and heads up display, levels and missions, narrative, and art and design elements.

- Website Code
- Game Code: The source code used to build the game, the language will depend upon the final game engine.
- Finished VR Maze game
- Final Report

5.0 Future Work

Throughout the semester, we made a lot of progress on The Maze Master. Due to COVID-19, however, our teamwork dynamic slowed down especially since we didn't have the technology at home to continue working on our project. Despite that, we were able to have a functional one player game. Players spawn at the beginning of the maze and have to find their way out to win while facing many challenges. In the future, we plan to add more features such as AI enemies because we currently have enemy prototypes. This adds more challenge to the player, and also allows players to use weapons that the team developed for a player's survival. Not only that, we plan to dive out of the storyline mode and allow for online play. This allows players to not get bored with the game, and compete with their friends or others. The groundwork for this feature has been set up as we have been able to connect to a server. We are using Photon Unity Networking 2 (PUN 2) since it allows for synchronous game environment and matchmaking. The implementation is still in progress, but it is a key feature we plan to incorporate. Another thing we plan to work on are little details in our visuals. We switched rendering pipelines towards the end of the semester for better visuals: fog, lighting, shader-graphs, and iridescent colors. The main feature visuals were fixed such as the maze walls, but other visuals were left untouched due to time. There is also a plan to expand the storyline and expand out from the maze setting to keep players interested.

6.0 Key Personnel

Nick Davis is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses such as Graphics Programming, Software Engineering, and is currently taking algorithms. He has experience with Unity2D, GitHub, and C#. He will be responsible for implementing game mechanics, testing, and polishing the game into a finished product.

Maharshi Dubasi is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Mobile Programming, Graphic Design, Software Engineering, and Algorithms. He has experience with Unity3D, GitHub, Blender, and C#. He is involved with developing the game mechanics, testing, and ensuring communication happens across the team to ensure a successful final product.

Jose Garcia is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed relevant courses such as Mobile Programming, Software Engineering, Algorithms, and Artificial Intelligence. He has experience with Unity, GitHub, and C#. He will be responsible for implementing game mechanics, testing, and polishing the game into a finished product.

Dale Hagen is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Software Engineering and Algorithms. He has experience with Unity2D, GitHub, and C#. He will be responsible for implementing game mechanics, testing, and polishing the game into a finished product.

Jaylen Kimbrough is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Software Engineering and Computer Graphics. He has experience working with Unity2D, Blender, and GitHub. He will be responsible for implementing game mechanics, testing, and polishing the game into a finished product.

Alec Ray is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Software Engineering and Computer Graphics. He has experience with GitHub, Unity3D, and C#. He will be responsible for implementing game mechanics, testing, and polishing the game into a finished product.

a. **Facilities and Equipment**

Oculus VR Headset - This headset is required to play our game. It is a virtual reality headset that is developed by Oculus VR, a division of Facebook. It allows players to enter the maze as if they were there, and hand controllers are used to navigate around the maze.

7.0 References

[1] “AMAZE 2 on Steam.” *AMAZE 2 on Steam*, Blender Games, 21 Apr. 2017,

store.steampowered.com/app/628750/aMAZE_2/.

[2] “Dark Deception on Steam.” *Dark Deception on Steam*, Glowstick Entertainment, 26 Sept.

2018, store.steampowered.com/app/332950/Dark_Deception/.

[3] “Maze of Gaea (Real Maze VR Simulation) on Steam.” *Maze of Gaea (Real Maze VR*

Simulation) on Steam, TiStar Game, 10 Oct. 2017,

store.steampowered.com/app/727900/Maze_of_GaeaReal_Maze_VR_Simulation/.