

Team no: 15

Team Members: Kenneth Bradburn, Zach Pearson, Zach Alleman, Jacob Prichard, Keith Wiehe

Project Name: True Stealth

Project Synopsis: A stealth game that uses machine learning and improved AI to give a better experience for players.

Project Description: Stealth games will usually have the same criticism about how the game is only about waiting for guards to follow the same path. This game will address these issues by using AI and machine learning to have the enemies learn the players usual movements to force the player to consider every move they make. The player will now have to plan actual routes and find ways around guards that are more in depth than waiting for guards to just move their route. This game is supposed to give players a new way of experiencing stealth to offer a more risk vs. reward experience. The end result is meant to reinvent how games tackle stealth sections or how games based around stealth should give players more engaging ways to evade enemies by having enemies act more realistic.

Project Milestones:

1st Semester:

Map made with zones and temp assets created. Deadline: 10/30/2020

- To be done by the end of October, this will allow us to get to work on our code for getting player movement and enemy movement.

Demo with reset feature, items, stationary guards. Deadline: 11/30/2020

- The Demo should consist of a possible level 1 and debug test rooms. The test rooms

Demo with working form of AI and pathing for guards. Deadline: 12/30/2020

- This demo should have a floor of the level able to be played with a set of working guards that are able to detect the player

## 2nd Semester

AI reaching final stages. Deadline: 1/30/2021

Full realize levels or floors: 2/28/2021

Polish with Better Models: 3/15/2021

Completion and publication: 4/5/2021

## Project Budget:

Hardware: team personal computers

Software: Unity engine

Estimated Cost:

Needed:	Cost:	Totals:
Unity Teams Advanced (8 months)	\$23 per month	\$184
Unity Asset Store (animations)		\$60
Publishing to Steam (optional)		\$100
	Total:	\$344

## Preliminary Project Design:

Using the Unity game engine, our project will be built as an Executable file. It should work with an OS that allows for a .exe file to be run, but we recommend that you use windows to run our project. We will run into a couple design constraints as we proceed through with our work. We have to consider that most users won't have access to higher end computer parts, so we need to test run our software on a lower end computer to guarantee it works. We will also be constrained by the amount of time we have to develop the project. We have considered making levels and potentially more modes for our project, but those will have to fall in line with our schedule. We might have to make a few sacrifices to get what modes or design details we want in the final product.

Our game will start with a player getting a short tutorial with the mechanics and learning how we have pushed to make our game more realistic. We want to change the user experience to focus solely on stealth and our team believes that AI and machine learning will be the best way to achieve that goal. Having enemy characters learn how the player moves and all their past action stored in memory will give the players a higher level of challenge and engagement.

To achieve this our goal, we will need to have a way for all the player actions to be stored for the AI to learn against. This is where the idea of a “reset” ability came from. If the player is ever caught or in a situation, they can't escape they will have the ability to reset the board. This reset provides a chance for the enemy AI to learn from the player's previous moves and now forces the player to come up with a new plan or engage with the enemies in a new way.

For instance, refer to Figure 2, a player would start by entering zone 1 and then would be caught in zone 5. The player would then have to reset and in that process, the game would set zone 1 and zone 5 as higher priority zones for guards to patrol. This in turn would disincentive a player from running the same route again. This would engage them into testing out zone 2 and seeing where they could go from there.

These ideas are not spelled out for the player to read either. It changes in the way they play. When a person plays a game, they don't want to be told they are doing it wrong. How many of us have been playing a game, whether it be a physical or board or video game and as we enjoy ourselves, a third party comes along to tell us we are playing wrong. It's an unpleasant situation to deal with, but if the messaging was done subtly then we could teach the player how to get better at the game without being forced to tell the player in a text box. Suppose you were playing kick ball in a muddy field and while you were having fun, you kept slipping when you tried to kick the ball. Sure, you could keep trying on the muddy field, but you would keep slipping. Now if you look over to your right you see a nice blacktop playground perfect for you and your friends to play on and after playing a round you notice no one slipped when they went to kick the ball. It's an inherent style of trial and error learning that will reward the players more because it's something they were able to solve themselves without a third party telling them how to do it at every turn of the game.

To give the players a decent challenge the enemies will also have to be smarter. In technical terms of games, enemies pathing and disregard for obvious signs of intruders is a lazy way to employ stealth mechanics. A guard shouldn't come across a dead body and stand over it for 5 seconds wondering what just happened to their fallen comrade then proceed on with their day. This should be a cause for concern and the guards should flood the area and look for who could be hiding. A guard shouldn't be distracted by the same bottle thrown in a corner from a random location for the tenth plus time.

By employing AI in these situations, guards will wise up and search the area for common spots they have caught the player before. They should alert all guards around them to help look or if the bottle trick has been used one too many times, look for the source.

An idea we have really wanted to see come into realization is the tools or bottle usage ability. A common idea in some games is that a thrown bottle should distract a guard, but it's unrealistic how using the item over and over should keep working. If a player were to use a bottle too much, then we should have them be punished by having the guards look for the source of the bottle rather than look at where the bottle landed. This could give the player a new way of using it by smashing the bottle near them and if the guards have been trained to search in the opposite direction of the bottle from too many uses they could look in different locations giving the player a couple extra seconds to hid or to try and get guards out of a specific zone.

Our project is meant to be a top down third person game and the player will be able to see most of the map available to them or as shown, a zone. The final ideas for how the games are meant to be played will be fleshed out after we evaluate our tools in Unity. One of our more prominent ideas has been that you are sent to steal certain items on the other end of the map. There isn't a clear win state as much as there is an item in that area you need to take and then get back to a starting zone to escape.

Figure 1

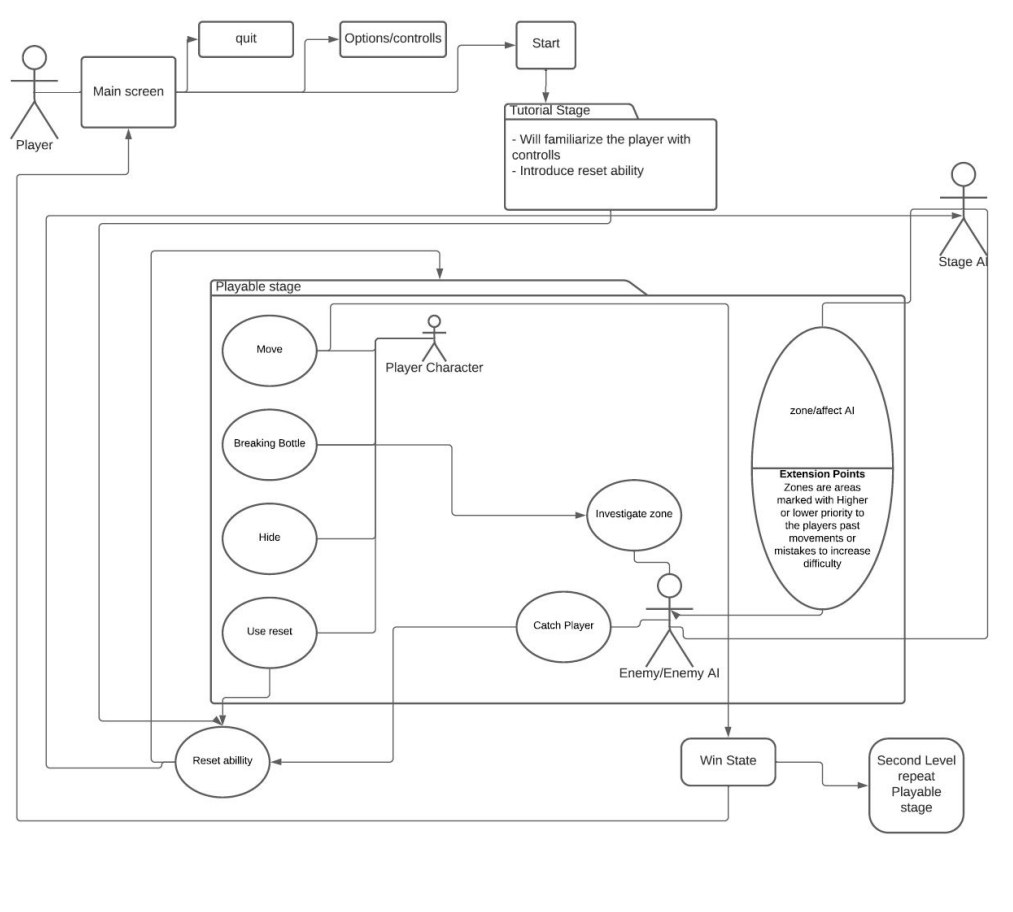


Figure 2:

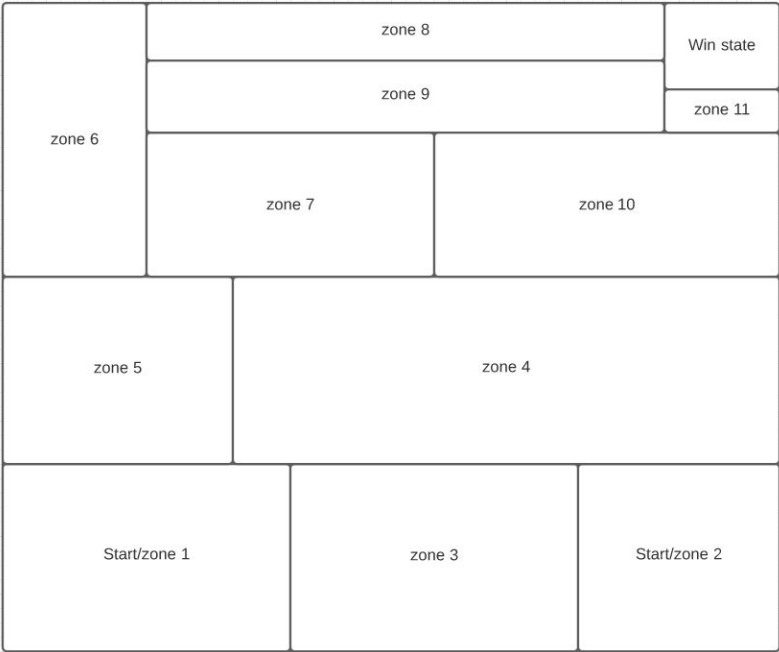
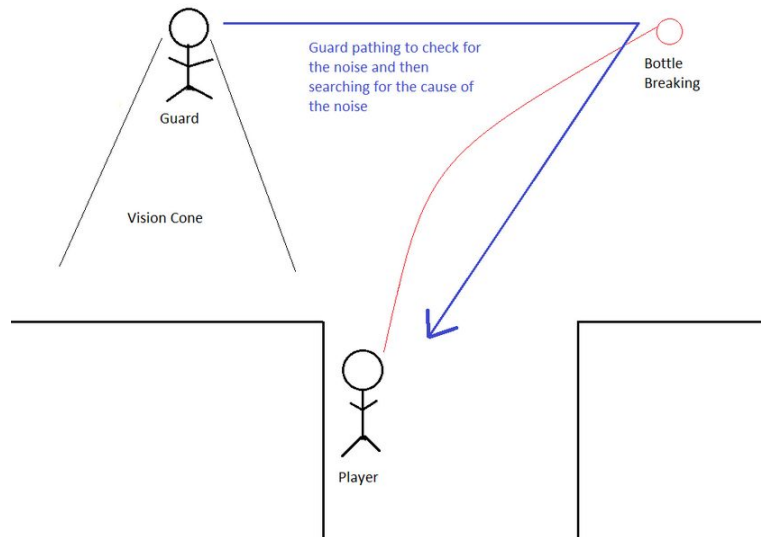


Figure 3:



#### Ethical Issues:

There is only one ethical issue this project is liable to come across. When creating a game, unless the project team wishes to make all assets themselves, many assets are purchased or acquired from open sources. This includes rigged skeletons, models, and snippets of code. To be ethical our team will need to properly acquire any assets we do not make ourselves. This means either properly documenting and sourcing our assets that are open source or paying for assets that are not free. There will be no pirating of assets in the making of this project.

#### Intellectual Property Issues:

The only intellectual issues that this project may encounter are the same issues that may be encountered ethically. That is the improper use or citation of other people's work. This can include, but is not limited to, rigged skeletons, models, code, story, or other defining features of a previously made game. By always citing our source for open-source material, and paying for any non-free assets we can avoid this intellectual issue. The project team will not use anyone else's code and label it as our own.

#### Change Log:

Gantt Chart has been updated to reflect current timing.

Name is still True Stealth but subject to change in the coming weeks.

Added Preliminary Project Design, Ethical Issues, and Intellectual Property Issues.

Gantt Chart:

