CS 4ZP6 Problem Statement

Yuan Gao
Su Gao
James Lee
James Zhu

Introduction:

For our Computer Science Senior Capstone project, we propose to create a “Cat-and-Mouse Game”, where players will control either a “Cat” or “Mice” character and battle each other in a maze. We plan on creating the game using the Unity Game engine, as well as various other programs to create game assets such as sound or models. This game will be multiplayer-oriented with the main “Cat vs Mice” game mode, as well as multiple other game modes such as King of the Hill or just Escaping the Maze. Teams will consist of one Cat and up to three Mice. The cat will have an advantage in speed and strength compared to the mice, which will require the mice to work as a team in order to defeat their more powerful foe.

Both teams will be able to complete tasks or battle neutral monsters in the maze to level up and gain new abilities and power-ups, in order to come on top of their opponent. Some examples of power ups and abilities include higher speed, becoming larger/stronger, and less ability cool down.

Motivation:

Growing up in the Digital Age has allowed us access to a myriad of electronic devices, and electronic gaming has become more and more mainstream. However, through the extensive library of games that we have played, we have yet to come across a fun, well-designed, stand alone Cat-and-Mouse game, where teamwork and tactical cooperation are rewarded.

While games such as Starcraft have spawned custom game genres which resemble Cat and Mouse, our proposal would bring a new, creative spin to this concept.

Challenges:

There are numerous challenges our group could face when trying to develop a game like Cat and Mouse. As none of our team members have experience with creating a multiplayer game using the Unity Engine, there are many challenges to overcome during the development process. Another is how to make this game appeal to and bring replay value for a mainstream audience. To make a multiplayer game enjoyable, we have to take in consideration of whether or not the game is fair to both teams. As the teams in this game are *asymmetrical,* meaning teams are not equal (in both individual strength and number), balancing is a concern -- as one team could easily be made too powerful. To remedy this, we will create unique level-up opportunities and benefits for each side. Another unique problem that comes with this asymmetry is that while the Cat team consists of only one player, the Mice have three. This may create a situation where if the players are well-coordinated in terms of communication, they could easily overpower the Cat on each play. Vice-versa, if the Mice are bad at communicating with each other, the Cat will dominate every match.

Thus, the procedurally-generated map (the “maze”), should also allow for *both* teams to play to their respective strengths. Designing an algorithm that will create such “balanced” maps would be a major challenge in itself. The algorithm should create a maze with no “sealed rooms” meaning every point of the map is accessible, as well as having dead ends and balanced neutral monster spawn camps, meaning that

they are not placed too close or too far from Cat or Mice spawn points. Additionally, the distribution of objects (such as power-ups) should not be overly-biased towards one team.

However, the map should also present a reasonable challenge to all of the players. As such, a variety of different terrain and environmental conditions must be possible (eg. mountainous, desert, rain, snow etc), each presenting their own tasks and obstacles which must be overcome. This should be done in a way that ensures continuity and harmony within the overall map, in addition to ensuring that it does not grant an unfair advantage to a certain team.

Objectives:

Our game should:

1) Appeal to all ages and demographics
2) Bring a creative and unique experience to the player, and provide for high replayability
3) Allows for some communication between the Mice (eg. such as voice and text-based chat, in-game quick “hotkeys”, etc)
4) Balanced and open gameplay, which is fair for both teams
5) And most importantly, let the player have a good time!

Assumptions:

We assume that individuals from all backgrounds and age groups would be interested in playing a fast-paced online multiplayer video game. The amount of video game titles and the number of unique game designs over the past few decades have resulted in a steady increase in gamers across all generations. We would therefore assume that a game of this nature will spark interest among a large section of the population.

We would also assume that most individuals own and can use a Windows PC ,which are the main targeted devices for this project. We also assume that they have the necessary files and libraries required to run the game (such as DirectX).

Our final assumption is to assume that the game that we are creating, a level up based Cat and Mouse game has not already been created by other people, and that it is original.

Constraints:

Time is an obvious constraint as we only have seven months to plan, design and implement the project entirely from scratch. Naturally, game companies will have many employees on a full-time basis designing their games, and even then they often take many years to “go gold”.

However, our game will not require as much resources, as we will also not be undertaking the numerous additional tasks that commercial game companies have to consider, such as advertising, porting it to multiple platforms, among others.

Graphics and audio is another constraint that we will have to deal with. As we are not artists, the quality of our art (such as models and sound effects), may not match that of other game titles on the market. Sound effects will have to either be downloaded from free sources, or we would have to make our own original sounds. Background music would also have to either be original or be used with permission from the creators. However, if we create most assets ourselves, and develop our own unique art and design style we can turn this barrier into a positive and distinctive aspect of our game.

Additionally, we would have to design the game and our assets in a way that does not infringe copyright laws, such as by ensuring that our characters do not look too similar to already existing characters created by other companies, and that our core game mechanics are not taken from existing titles.

Technological Aspects

As mentioned above, the Unity Game Engine will be the main *platform* during development. It has native support for both Windows and Mac based-PC's (where the majority of our target audience lies), in addition to mobile.

Additionally, the main *language* of development is **C#**, which is *imperative and object-oriented*, and thus, well-suited to a game such as ours, which will involve the rendering and manipulation of many objects (such as potions, characters, map elements, etc).

The Unity engine is easy to use, has a built in IDE that supports C# , as well as includes through documentation a series of tutorials on its website. Due to the of the popularity of Unity, it also sports a very active development community and thus, enables both novice and expert developers to familiarise themselves with and start using the engine in production very quickly.