Formal Proposal

CSC 584 Building Game AI

Debalin Das and **Kunal Kapoor** and **Utkarsh Verma**
Department of Computer Science
North Carolina State University
{ddas4, kkapoor, uverma}@ncsu.edu

Abstract

Crimsonland is a 2D (top-down view) survival game where the player is trying to stay alive while various enemies converge on him from all sides of the map. The player is placed in the middle of the map at the beginning of the game and the enemies enter from around the boundary of the map. The player has to shoot the enemies to kill them and stop them from reaching him. Killing enemies also earns the player points, and the objective is to try and maximize the points earned until death.

Our objective is to create a game like Crimsonland [1], where we want to try out, invent and implement novel enemy behaviors to make the survival game interesting. The variety of enemies will range from simple wanderers without a goal in mind to beings with swarm intelligence that try to use their numbers to corner and kill the player. Additionally, we plan to add a respawnable helper bot for the player, which will be used for assistance in dropped item collection. The goal of the project is to create various interesting types of enemy bots demonstrating different AI techniques to defeat the player, and balance the equation by creating a helper bot to assist the player.

Introduction

The main purpose of our game is to make a testbed for strategic movement techniques seen in enemy AI in games and analyze the efficiency of an assistant bot in a survival game. The game starts off with the player at the center of the map. With time, enemies emerge from the boundaries of the map and come towards the player. The player needs to fend off the enemies to save himself.

For simplicity (and also as seen in the original game), the enemies don't have the ability to shoot to harm the player. They can only come in physical contact with him to cause any damage. On the other hand, the player has a gun with which he can shoot and kill the enemies (no friendly fire). Different enemy types would inspire different strategies from the player to evade and score higher. As the enemies come in swarms and their numbers increase with time, it will become difficult for the player to keep them off himself and stay alive.

To make it more interesting, we decided to add an assistant bot with the player, which will go around the map on its own and pick up ammo deposits and ferry them back to the player. Experimentation with restrictions such as how far the assistant bot can go or how good the bot is at path-finding could yield some interesting results.

Background

While building AI for games, movement techniques, decision making and path finding are some of the most important and significant aspects which should be taken care of. We decided to focus mainly on these three things and improvise to make them look cool and fun.

Movement techniques range from simple kinematic motion to more complicated dynamic motion techniques like seek, wander, flocking and many more. We plan to keep a mix of several kinds of motion techniques to get a feel of each within the context of the game. To give you a glimpse into our thoughts, in some cases the enemies would directly seek the player and in others, they would wander around the map before coming within a certain radius of the player and then darting towards the player to completely smother him. Other collective intelligent methods like flocking and hive minds would also be implemented in interesting ways to attack the player. Swarm intelligence behaviors could allow a group of enemies to corner the player in place so that he has nowhere to move. Some flocks of enemies could follow their leader and then go berserk once the leader dies. There are tons of fascinating ideas that can be put into a genre of game like this and we plan to implement at least a few of them.

Let us not forget our favorite assistant bot though. This bot would have the simple task of going around the map, picking up item drops of ammunition and returning to the player to provide him with what he has gathered. Ammo deposits may be randomly spawned at different parts of the map. He can't go to every place as that might not be efficient enough. So, he has to decide which deposits are near to him and then proceed. Moreover, instead of going straight towards the deposits, it can compute its path dynamically (keeping in mind the enemy positions or obstacles) and then move on that computed path. Thus decision-making and dy-

namic path-finding will be some important fundamentals in our game. We also hope that over the course of our development, we would actually come to like the assistant bot enough to name it!

Problem

The enemy bots are of varying types, with each type operating on different strategies. The AI logic for seeking the player, and trying to surround him collectively (hive-minds) is tricky to get working in a natural-looking manner.

The helper bot AI for picking up the dropped items is seemingly simple, if we just have it seek each individual dropped item, then carry multiple such items back to the player. However, this may not be efficient, and a very natural looking behaviour.

Instead we plan to implement an informed search algorithm like A* or hillclimbing [2, 3] (we will weigh several algorithms during development) for finding the shortest path which covers all the items and brings the bot back to the player in the shortest time. Pre-deciding such paths, keeping in mind the obstacles to avoid on the way complicates the bot AI, but will give a much better life-like behaviour, not to mention make the bot efficient at its job. Figuring out this path in real-time will pose a problem, which we hope to overcome during our project work.

Potential Benefits

Putting all the movement techniques, collective intelligence, pathfinding and decision making into the game, we would see how each aspect plays out with others. More importantly, we would be able to analyze how each of the enemy AI techniques (and a collection of them) would be handled by the human player. Does a culmination of too many enemy types make the game less interesting? Putting it in a different way, would it be more fun if the different types of enemies came one at a time in stages? From the current generation of games, we can draw a conclusion that the answer to the previous questions is somewhere in the middle. But then again, it depends on the player himself and what his primary approach to playing games are. Building these constructs into the game would give us a chance to study the same and comment on them.

What happens when we put the assistant bot into the mix? Its job is as simple as picking up ammo and providing the same to the player. In the process, how will it affect the player's connection with this character? In a fast paced game like Crimsonland, there is not much time to pick up ammo. Thus, in Crimsonland, the player always had infinite ammo. As a test, here, we are taking that feature out of the game and providing a bot to help with the ammo picking. We want to see how this unison plays out. The bot wouldn't be completely vulnerable to contact damage, so the player would have to shoot the enemies near the bot when it is fetching ammo. How would this dependency/symbiosis with an AI character affect the fun factor of the game?

Tasks and Techniques

This is a survival game with waves of enemies converging on the player, trying to kill him. This gives us a platform to create many different enemy bots demonstrating a combination of AI techniques. Along with that, our assistant bot will showcase its own independent path-finding techniques to ferry the ammo drops back to the player. Apart from these, we will briefly touch upon the map generation and the player himself which will be controlled by the user.

Map Generation

For simplicity, our map will only contain a constant terrain and randomly generated patches of obstacles. The AI bots in the game as well as the player needs to avoid the obstacles while moving around the map. These obstacles can be put to some interesting use. An enemy AI will no doubt be fired upon constantly by the player and more often than not, the enemy will die before reaching the player. But we can program the bots in such a way, that when they have very low health, they will run for cover behind obstacles to avoid incoming fire. This will add a different kind of realism to the game. We plan to touch on some of these subtle things when we implement the game.

The player will spawn at the center of the map and the assistant will spawn near him. The enemies will spawn outside the visible area of the map and will come towards the player. In the process, they will become visible.

Player

The player is represented as a distinct shape/sprite on the map, and is under direct control of the person playing the game. The player can move around the map in all directions. Player will be responsible for moving strategically across the terrain, avoiding obstacles and using them as cover when necessary. The player has to avoid coming in contact with the enemy bots, which can lead to death. The player will also be able to shoot at enemies while moving, and will receive help with collecting ammunition from the assistant bot.

Enemy Bots

There will be several kinds of enemy bots in the game, which will differ in appearance, and behaviour. However, most enemy bots have one common goal, which is to kill the player by coming in contact with him (there will be a few simpler bots who just turn up during gameplay to wander the game environment and eventually dying). By providing a variation of enemies, we plan to test them together in a single session of the game as well as individually (we will talk about this in more detail later). We are listing a few of the many different enemy types that we have thought of below.

1. **Soldier**: These are the basic and most common enemy units, which follow a simple rule. They directly seek the player, and try to kill him by contact. They move at a constant speed throughout the map, and are simple in their AI complexity. They are unaware of other enemy bots, and thus each such bot moves independently. These bots have

normal hit-points (HP). After a soldier is generated outside the visible area of the map, it starts doing a position matching seek with the player. In the process, it tries to avoid any obstacles that might come in its way. We could figure out a pre-defined path avoiding obstacles for all of these bots, but we plan to keep the path-finding technique only for the assistant bot. In that way, we can reflect and analyze the differences between the two. So, instead of using path-finding, we will make a soldier start moving in the direction of the player; if it notices an obstacle within a certain radius of its location, it will avoid it and then move forward. It does not have any knowledge of other obstacles lying on its path unless it encounters one.

2. **Grunt**: The grunt would aimlessly wander the map searching for a cause unknown to mankind just so that it could serve as another obstacle for the player to murder. The idea behind this is that in a game with all "intelligent" AI bots, we need a few that are essentially simple and, to an extent, random in order to provide a better gameplay experience.

3. **Hermit**: These bots are move powerful than the soldier bots, as they can move much faster and have more HP, making them harder to kill. These bots do not actively seek the player but they wander around the map in pseudo-random fashion. They can only see as far as a certain radius allows, and are harmless when the player is not within their visible range. However, when the player comes within this range, these bots get immediately alerted. Upon activation, they immediately steer towards the player and move towards him for a kill. This will be implemented using a seek-steering behaviour, wherein the bot accelerates to max speed while moving towards the player. Otherwise, it will display a orientation based wandering motion.

4. **Flocker**: These are (collectively) intelligent enemy bots which spawn and move around in groups. Each such group has a leader, where all other members follow the group leader. The leader alone seeks out the player and moves towards him, while the group members keep following the leader. The leader has a pre-defined alert range, and once the player is within that range, it alerts the group followers, who then spread out and try to surround the player independently. Moreover, we plan on playing with the flocking pattern (i.e. the structure formed by the followers around the leader) and see which one looks the most interesting.

5. **Martyr**: Martyrs are small weak bots that follow their leader with the sole purpose of keeping the leader alive. They try their best to sacrifice themselves in order to keep the leader out of harms way. Once the leader is dead, however, these numerous creatures go berserk and chase after the player extremely fast for revenge. This is an interesting aspect in game enemy design which actually gives a human touch to the game. The player, for example, may understand this behavior and decide to shoot all the young-lings before targeting the leader.

6. **Blender**: These bots start off with very low HP when they spawn, but they have an interesting feature. Every blender will look at a certain radius around itself and tries to spot another blender within that area. If it finds any, then both (or more) of these blenders will come together and unite to form a bigger blender having more HP and speed. This blender will be more difficult to kill than the smaller blenders. So its imperative for the player to kill any blender whenever they spawn, so that they aren't given much time to coalesce and form larger and stronger blenders. This might give an interesting choice to the player to think and decide which enemy to kill first; allowing a blender to roam around for long may be more dangerous to the player than anything else. Furthermore, the goal of a blender could be two-fold. Knowing that to kill the player the blender must be stronger, it could have a dynamic goal matching behavior to initially find other blenders and morph into a stronger being before going after the player.

Assistant Bot

The assistant bot is a computer-controlled in-game helper for the player, who can fetch items from the map. As the game progresses and enemy bots are killed, several ammo items are dropped across the map randomly. These items can be picked up by player to gain more ammunition. The bot will take the shortest path from its starting position to the bonus items spread across the map, collect them and then return to the player. This is where we will make use of the path-finding algorithm based on an informed search algorithm like A* or greedy, for the bot. Instead of making it seek the ammo deposit directly, the bot will dynamically pre-compute a path to the same and follow that. If it comes in contact with any other characters while travelling it will change its course as required. Thus we plan on implementing a steering pipeline combined with path-finding for the assistant bot. As we said before, the functionality of the bot and the finite ammo state of the player will supposedly give an interesting outlook to the game. We look forward to analyze how immersive this construct turns out to be.

Evaluation Methods

Being a survival game, the game ends when the player dies. The different kinds of enemies in our game would give us enough scope to analyze how different types of AI behaviors affect the gameplay. To stick to the basics, a game shouldn't be insanely difficult or terribly easy. As we would start implementing the characters, there will be a lot of values that we would be needing to tinker with to figure out that exact line between too hard and too easy. Also, having thought of a variation of characters, we can test which characters would suit best with each other, i.e. instead of putting every kind of character at the same time in the map, would a certain combination of characters make the game more fun and immersive? The game should also start to get increasingly difficult as it progresses. This is an important metric directly

affecting a users interest in the game, because this increasing rate of difficulty will pull the user more into the game as it becomes more challenging.

There can be various metrics to measure the effectiveness of the AI bots. Ranging from how long could the enemies survive on the map to how each of the flocker and martyr elements were successful enough to protect their leaders, all of these different measures can be applied to individual AI characters to see whether they are being useful in the context of the game. Among other things, getting the hermit bot to work perfectly would require a lot of effort.

One significant evaluation aspect of the game would be the assistant bot. As mentioned before, we have taken a leap of faith here and ditched the infinite ammo concept generally present in such survival games. Instead, inspired from AI companions seen in The Last Of Us [4], Bioshock: Infinite [5], etc, we decided to test how would a similar companion work in the context of this game. To make it more interesting, we have also kept the assistant non-vulnerable to contact damage. How does the player in a session of the game help the bot and maintain a give-and-take relationship with it? Or does he get tired of the assistant and feels it's just more headache? In that case, we would want to increase the efficiency of the bot and make him carry more ammo to the player. The player needs to feel that he actually needs the assistant bot instead of feeling that it's just a pawn in the game. Thus, we feel we would have to dynamically evaluate every outcome of the assistant bot behavior and correspondingly change the same to make it fit for such a game.

Conclusion

This multi-enemy combat game is a good platform to showcase several AI techniques, while also being simple and fun to play. We have identified and proposed several different AI strategies for the enemy bots, and a path-finding technique to be followed by the assistant bot. We can explore different artificial intelligence algorithms and directly see their effectiveness in different game situations thereby understanding the elements of an engaging game. We intend to experiment with different strategies in order to understand their impact on game quality and hope to derive some interesting results from them. All of these will be working in tandem on the game map, and can prove to be quite challenging to get right.

Forseeing how the different AI algorithms would perform in a real game environment is difficult at this stage. The most influential feature would be to think of innovative ideas of enemy behaviors. Interesting enemy behaviors could then be implemented using several common AI techniques or by thinking of a unique solution. The comparison between these different implementation methods would be our prime takeaway from this project.

References

[1] *Crimsonland - the Game*. URL: http://www.crimsonland.com/.

[2] *A* Search*. URL: https://en.wikipedia.org/wiki/A*_search_algorithm.

[3] *Hill Climbing*. URL: https://en.wikipedia.org/wiki/Hill_climbing.

[4] *Ellie in The Last Of Us*. URL: http://thelastofus.wikia.com/wiki/Ellie.

[5] *Elizabeth in Bioshock: Infinite*. URL: http://bioshock.wikia.com/wiki/Elizabeth.