MIMICKING HUMAN-LIKE BATTLE BEHAVIOR OF ENEMIES IN A GAME

A Project

Presented to the

Faculty of

California State Polytechnic University, Pomona

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

In

Computer Science

By

Wan Suk Lim

2022

SIGNATURE PAGE

PROJECT:	MIMICKING HUMAN-LIKE BATTLE BEHAVIOR OF ENEMIESIN A GAME
AUTHOR:	WAN SUK LIM
DATE SUBMITTED:	Fall 2022
	Department of Computer Science

Dr. Markus Eger
Project Committee Chair _____
Assistant Professor of Computer Science

Dr. John Korah _____
Assistant Professor of Computer Science

ABSTRACT

In most role-playing games, non-player character enemies do not show intelligent behavior. Non-player character enemies usually stand still or move around simply and wait for a human player to approach to do anything. Usually, non-player character enemies attack the player until they die. Human players would find the games more fun when those non-player character enemies show more human-like intelligent behavior. In this paper, I designed and developed logical behavior for the non-player character enemies in a game. This human-like battle behavior by non-player character enemies can bring more fun for human players to play a game.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In most role-playing games, enemies usually stand still and wait for a player to approach to do anything. Usually, enemies attack the player until they die. There's not much human-like battle behavior by enemies for the player to think that enemies have some kind of artificial intelligence. Humans have certain battlefield behavior by default to survive and win the battle.

If the enemies show more human-like battle behavior in a game, the player might enjoy the game more. Humans might play the game more if they feel like they are playing with other humans through the human-like behavior of enemies. Therefore, it will be interesting to make enemies show human-like battlefield behavior in a game.

CHAPTER 2

LITERATURE SURVEY

The article [1], GDC 2005 Proceeding: Handling Complexity in the Halo 2 AI (Damian 2015), talks about the behavior tree with priorities for NPC (Non-Player Character) to choose what to do at each moment. The goal of the article is for NPC to appear to have common sense in terms of behavior while managing complexity. For example, if a player gets off a vehicle to walk away, the ally NPC would also open the door of the vehicle and get off the vehicle to go with the player.

The game, Halo 2, is a real-time FPS and NPCs have to make decisions fast in real-time. The behavior has to be coherent which means the NPC should start, stop and change actions at appropriate times. And NPC should avoid the problem of dithering - the quick flipping back and forth between two or more decisions. An NPC should be transparent where the player can reasonably guess what the NPC is doing. Also, the player has to be able to understand what's going on in the game. If too many things happen at the same time, it's difficult to understand what's going on.

The paper [2], Finite State Machines for Building Believable Non-Playable Character in the Game of Khalid ibn Al-Walid (Fathoni, Hakkun, Nurhadi 2020), talks about different Finite State Machines for different kinds of NPCs. An NPC whose job is mainly talking has 2 states - idle and talk. The player can trigger the NPC to talk from the idle state. An enemy NPC which can attack the player has a different FSM with different states such as

search player, move to player, and attack. FSM is a typical way for games to make NPCs behave differently to make the player believe that NPCs have intelligence.

The paper[3], Three States and a Plan: The A.I. of F.E.A.R. (Orkin 2006), talks about an interesting way to use FSM and A*. Their FSM has 3 states - Goto, Animate, and UseSmartObject. UseSmartObject is a specialized data-driven version of the Animate state. Instead of telling it which animation to play, the animation is chosen through a SmartObject in the game database. So it's possible to consider UseSmartObject to be the same as Animate. Then, it's 2-state FSM, Goto and Animate.

F.E.A.R. implemented planning which is a process of searching for a sequence of actions to satisfy a goal. They implemented planning similar to STRIPS planning which consists of goals and actions where goals describe desired state of the world to reach and actions are defined in terms of preconditions and effects.

CHAPTER 3

RESEARCH GOAL

Research goal would be to mimic human-like battle behavior by enemies in a war game. Usually in wars, no soldier fights enemies alone unless a soldier has a specific role like a sniper. Soldiers are usually in groups and the groups have common goals such as defending certain areas or attacking certain areas. Sometimes army groups retreat from certain areas if they realize they can't win the battle. We can mimic such dynamics in a game.

Enemies can be grouped in 3 or more members. If one member finds the player to attack, the whole group will go together to attack the player together. If only one or two members are alive in a group, that enemy would run away from the player to find other enemy groups. When running away starts, the enemy would make a sound of screaming "Help!" to find help from other enemies. If the running-away enemy finds another group of enemies, they go together to the original location of the running-away enemy to find the player to attack together.

When an enemy or enemy group encounters the player, they can assess the strength of the player by the difference of entity numbers. Then, if the enemy realizes their strength is weaker than the player's strength, one or two enemies would run around to gather more other enemies to engage in the fight. Or the whole enemy group would run away from the player as they think they would all be killed by the player.

CHAPTER 4

METHODOLOGY

I used Unity and wrote my own code to have a game environment as seen in Figure 1 and Figure 2. Graphical assets [4] are free to use. Human plays one player character to find monsters and kill like a usual role-playing game. It's real-time based. I placed groups of monsters 1 - 10 per group. I designed a way for each enemy to check how many other enemy monsters are nearby. If there is no other enemy character than just 1, run away from the player and remember the location where running away started. The enemy should make a text and a sound of screaming "Help!" to gather other enemies as that's what normally human soldiers would do at a battle.

Figure 1 Graphical Environment

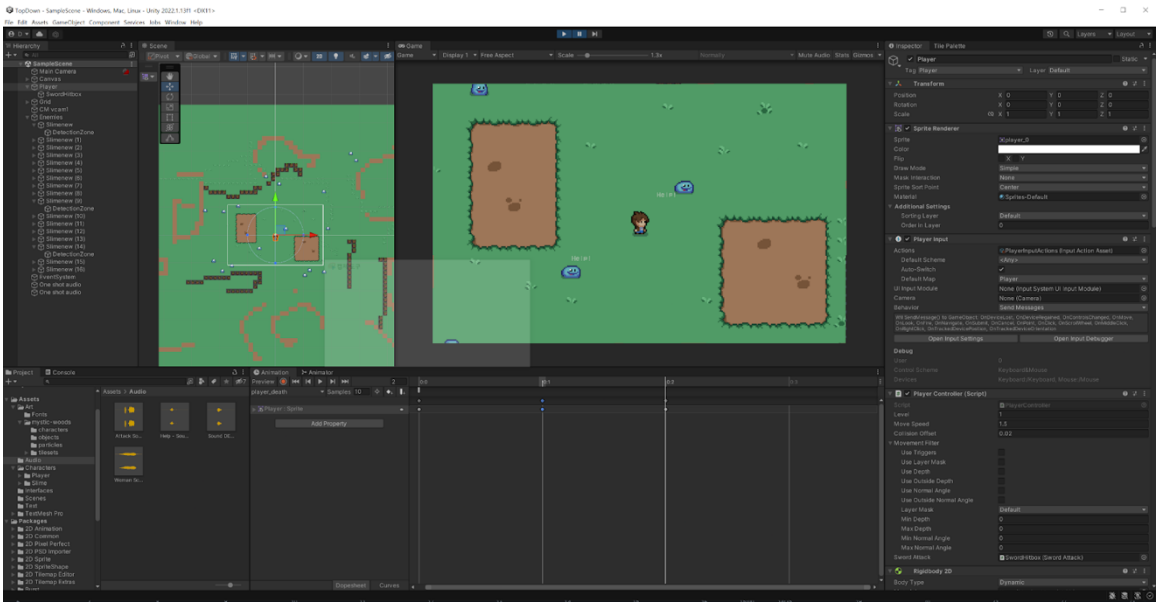

Figure 2 Unity

If the lone survivor enemy finds another group of monsters, join the group and make the group go to the location where the lone survivor ran away from. The enemy group should make a text and a sound of screaming “Attack!” to gather other enemies as that’s what normally human soldiers would do at a battle. If the lone survivor doesn’t find another group of monsters, keep running away from the player. Play the game as a player to see if desired behavior is working with monsters.

The enemy should gauge the player’s strength constantly and compare it to the strength of the enemy group engaged in a fight. If the strength of the enemy group is significantly lower than the player’s strength measured in difference in numbers, one enemy would run around to gather enough enemies to kill the player.

CHAPTER 5

EVALUATION OF RESULTS

To evaluate the human-like rational battle behavior, a user has to use the main player character and interact with monsters to see the reaction of monsters. The user would think the monster is more like a human when the monster can make intelligent decisions of when to fight and when to run away.

The player can see if an enemy monster is left alone within close distance with the player. If the enemy monster is attacking the human player without other enemy monsters nearby with significantly lower strength compared to the player, the enemy monster fails to show the human-like rational battle behavior. If the enemy monster realizes that there is no other enemy monster nearby attacking the player together and starts to run away from the player, it shows the human-like rational battle behavior. And if the enemy monster finds another group of enemy monsters and they start to go back to the original location where the lone survivor came from, it shows successful human-like battle behavior.

The user can see if the enemy makes appropriate text and sounds like “Help!” while running away from the player and “Attack!” when the enemy goes toward the player to attack. The user can see if the enemy assesses the strength of the enemy group and strength of the player and make an appropriate decision to fight together or run around to gather more enemies to match the strength of the player.

More specifically, these are each item to check for testing each component.

1. Find the enemy group that has 3 or more monsters around.
2. Move the player to the enemy group and check if the monster group attacks the player or runs away.
3. If the monster group runs away, see if they find other monster groups and come back to the original location together
4. Kill most of the monsters in a group and see if one or two monsters are left and that monster runs away.
5. When the monster starts to run away, see if the monster makes text and sound “Help”
6. When the monster starts to fight the player, see if the monster makes text and sound “Attack”
7. See if that running away monster finds other monster groups and comes back to the original location with the monster group

Figure 3 Enemies running away from the player. One enemy on the right top side of the player knows there are not enough enemies around to kill the player successfully, so the enemy makes the sound and text “Help!” and runs away from the player character.

Figure 4 Enemies attacking the player. Four enemies on the left side of the player character know that they have enough strength (more than 3 enemies) to successfully kill the player, so they go towards the player with the sound and the text “Attack!”

Figure 5 Enemies running away from the player again. Out of four enemies on the left side of the player from the previous situation, two of them got killed by the player. The remaining 2 enemies realize they are not strong enough to kill the player successfully, so they decide to run away with the sound and text “Help!”

CHAPTER 6

CONCLUSION AND FUTURE WORK

It feels a lot more interesting to play the game against this enemy behavior that mimics human battle intelligence than the typical standing and attacking enemies. Many successful online games are based on the multiple players cooperating or fighting each other. If we can add more intelligent behavior to the enemies so that it becomes hard to distinguish if an enemy is an AI or human, the games don't need to rely on concurrent numbers of human players for enjoyment of the player.

There can be different types of enemies with different attack types and strategies for more intelligent battle behavior such as one enemy luring the player towards location where many other strong enemies are waiting to kill the player. Or an enemy can choose to seem very weak by intentionally getting rid of items an enemy is wearing to lure the player, but then suddenly use powerful skills to kill the player. Such deceitful strategies are highly intelligent battle behavior humans use to trick other humans for their advantage and can be achieved by AI in the future work

REFERENCES

[1] Damian Isla Blogger March 11, 2005. (2005, March 11). *GDC 2005 proceeding: Handling complexity in the halo 2 AI*. Game Developer. Retrieved August 18, 2022, from https://www.gamedeveloper.com/programming/gdc-2005-proceeding-handling-complexity-in-the-i-halo-2-i-ai

[2] Fathoni, K., Hakkun, R. Y., & Nurhadi, H. A. (2020). Finite state machines for building believable non-playable character in the game of Khalid ibn al-Walid. *Journal of Physics: Conference Series*, *1577*(1), 012018. https://doi.org/10.1088/1742-6596/1577/1/012018

[3] Orkin, J. (n.d.). *Three states and a plan: The A.I. OF F.E.A.R.* Retrieved August 18, 2022, from https://alumni.media.mit.edu/~jorkin/gdc2006_orkin_jeff_fear.pdf

[4] Game Endeavor. Assets Retrieved August 18, 2022, from https://game-endeavor.itch.io/mystic-woods