Grand Canyon University

Project 5 - Learn From Mistakes and Successes

Justin Dietrich and Ryan Scott

CST-415: AI in Games and Simulations Lecture & Lab

Dr. Ricardo Citro

March 9, 2022

A brief description of the game or simulation (one paragraph)

1. Our game will be from a top-down view, set in an environment that has paths, walls, and other types of cover or obstacles. Project 5 features (alongside the Bellman equation) attacking enemies and the first pass on basic User Interface for the player. The player (now equipped with either a gun or knife) will move through randomly generated grid-like levels, while fighting off a variety of different enemies (now including some enemies that learn) that are placed throughout the level with weapons of their own.

How are the concepts listed above relevant and its purpose? (one paragraph)

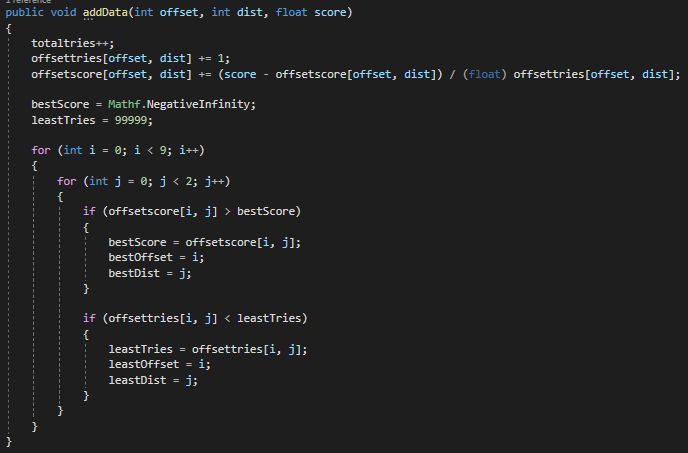
1. We use the concept of temporal difference learning and adaptive dynamic programming by having enemies learn. There will be a few enemies that, when the program runs, have a goal but no knowledge on how to accomplish it. They will learn what to do through trial and error, exploring what happens when certain actions are taken, replicating actions that yielded good results, and avoiding actions that yielded bad results. This also gives the Player a sense of increasing difficulty as their enemy becomes more intelligent as time goes on.

Which search method will be used? (one paragraph and bullet points outline)

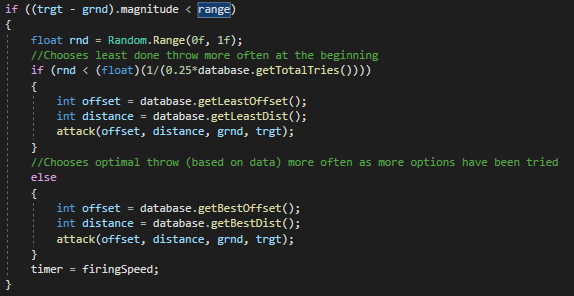
1. Enemies can now throw grenades, but do not know what the best spot to throw one is. We have a database that starts with no info at runtime. As enemies throw grenades, they will choose what direction to lead it depending on how the player is moving as well as how far to lead it. The possible options for direction are the direction of player movement along with 7 other directions (spaced 45 degrees apart, making the options 8 directions around the player). There can be a small lead, large lead, or no lead at all. Upon throwing a grenade, the options chosen are remembered. When the grenade explodes, it returns a score that shows how well placed the grenade was. The score is recorded in a table (2D array) in the database. The enemy could choose either to experiment with the least tried option or choose the best option. The chance of choosing the best option increases and the chance of choosing to experiment decreases as more tests are done.

Github: <https://github.com/AsePlayer/CST-415>

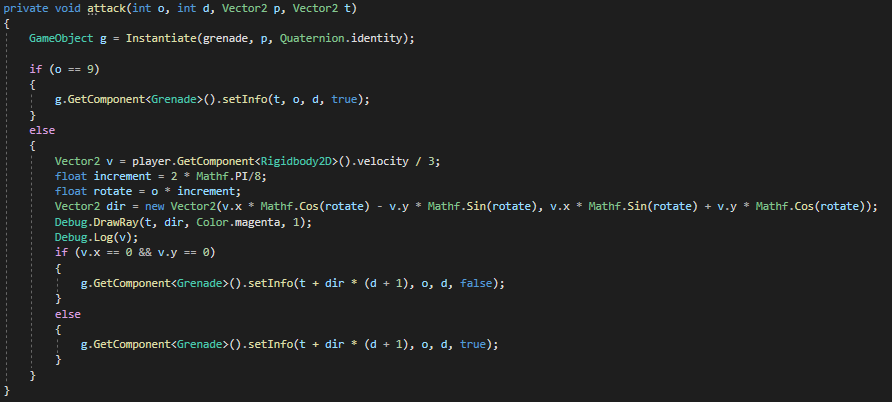
Example Scripts



When adding data to the database, it keeps track of how many tries and its average score for each possible option. It also finds a new least tried option or best score option every time a score is added, so the function can easily return it when called.



When a grenade is thrown, the enemy decides whether to experiment or choose the optimal choice using random numbers and (1/0.25\*database.getTotalTries()).



When a grenade is created, its data on where it was thrown is recorded to put in the database once a score is calculated.

How will you overcome unforeseen obstacles during implementation? What is your 'plan B'?

1. If it doesn’t work, we would at least just get it so that the enemy randomly experiments and doesn’t record data or make informed decisions. It would really only be half of what is required, but it’s better than no plan B.

How is the project aligned with the current topic objectives?

1. The project shows that we can create a program that can make decisions based on the consequences of previous actions.

| Temporal Difference | Database starts empty at runtime and is filled as tests are done. |
| --- | --- |
| Adaptive Dynamic Programming | The enemy will start to make informed decisions on what to do as data is recorded. Data may vary depending on the player’s actions. |
| Action Utility Functions | Scores are calculated for how well placed each thrown grenade was. Based on player hp and friendly fire mostly. |

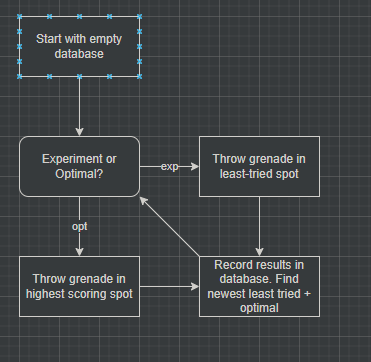
What will appear on the screen: animation, user interactions, information dashboards, UI elements, etc.

1. Project 5 now has, along with what has been available previously, custom-made grenade sprites that have been expertly crafted in MS Paint. It also features the first sprite that changes, with the grenade having a base and exploding sprite (also professionally illustrated in MS Paint).

List the platform and software tools you plan on using

1. Stuff we will use:  
   Unity (with C# scripts)  
   Adobe Photoshop  
   Adobe Illustrator  
   MS Paint  
   Audacity

Flowchart for Adaptive Learning:



Screenshots below:

