BroncoDrome Expected AI Behavior

Team Kitty Cat Studios

Written by: Andrew Jacobsson

Introduction

As an AI developer, I need to come up with a mockup of the AI's expected/ideal behavior as well as brainstorm potential AI strategies and design for engaging and intelligent opponents.

Artificial intelligence is a complex problem. Creating an agent that is both believable, efficient, and effective is no simple task. Given that BroncoDrome is in a 3D environment, this problem has its added challenges as there is a whole extra dimension of space to account for. Determining the ideal behavior and brainstorming strategies to make this behavior a reality, it will be significantly easier to design the AI for BroncoDrome.

Unreal Engine fortunately provides a host of tools to develop artificial intelligence through behavior trees, their environment query system (EQS), AI perception, and debugging tools. (See "Resources" section at end of document)

General Expected Behavior

The AI agents in BroncoDrome are the enemy "runners". They control go-kart-like vehicles exactly like the player with the goal of eliminating the player (as well as any other nearby runner depending on the game mode). They have the ability to drive forward, reverse, turn right, turn left, and jump. They can shoot a projectile that if it collides with another runner, will damage/destroy them.

They should avoid obstacles while simultaneously attempting to dodge any incoming projectiles by jumping or evasive tactics. They should make use of the environment and learn to pick up power-ups. They should have uniqueness or randomness to their behavior so that each AI appears to behave differently and with a greater or lesser sense of tactics/intelligence based on their difficulty level.

The AI should be of "reactive" design to the environment since it is difficult to plan in a game like this.

Pathing

The AI should intelligently and seamlessly path through its environment, modifying its path depending on its stimuli and if it notices a power up or another runner. It will attempt to avoid collisions with other runners and path around obstacles.

If it notices a runner, it will not drive straight at the runner but maybe try to circle the runner. If it notices a powerup, it will drive straight to the powerup but may modify its path if there is an approaching enemy runner and it notices they will get the powerup first.

One potential solution is to have preset "pathing points" around the map. If the runner does not currently have a path and no target, it will pick a random point on the map and generate a path to

that point, updating the path as it goes depending on stimuli and repeating the process as necessary. This solution would help prevent the bot from getting stuck in corners and potential loops in avoiding geometry. It would also help create fairly organic/believable paths and would keep the AI fairly active on the field. It would also allow certain bot difficulties/personalities to pick from a different pool of pathing points for greater variation. The negative with this approach is that it requires additional effort to plan and place these pathing points and developing the code for this approach.

Targeting/Attacking

The AI, depending on its strategy and game mode, should typically target the nearest runner (or player) that it has line-of-sight on within a relevant radius. (Meaning, even if a bot has a small line of sight across the map, they probably wouldn't take a shot at you) They should not target runners that are behind walls/obstacles. As often as they can (use a shot timer that could vary based on personality/difficulty) they will attempt to shoot at any relevant target they can. If they cannot shoot, they will path around the map until they find a new target.

Difficulty

The AI should be designed to have varying difficulty levels at bare minimum: easy, medium, and hard. The difficulty level should affect the AI's senses and how keen they are of noticing runners and power ups, as well as their shot frequency and shot accuracy. They should also have varying delay levels in response to stimuli, lagging a bit at easy or responding almost instantly at hard. A hard bot should make its shots 70 to 90% of the time as an example, whereas an easy bot should make its shots in the realm of 20-30% of the time.

Personality

To create personality in the AI, it may be necessary to create additional modified AI scripts and randomly assigning a script from an available pool to each AI on spawn. Slight changes could be made where one AI is more hesitant to engage in fights and has a lower radius of engagement and will seek to run around the border more avoiding most players, where another design could be an AI patrolling the center and immediately chasing any runner it detects nearby. Another could add a random chance to jump at any given point. These slight differences in behavior would enhance the overall gameplay and user experience creating the sensation that the AI is relatively intelligent, believable, and generally more interesting.

Resources

Unreal Engine 4 AI system:

https://docs.unrealengine.com/4.26/en-US/InteractiveExperiences/ArtificialIntelligence/