

Alec Bielanos

IGME 202 Section 2

Final Project Documentation

Description of World:

To put it in a phrase, I would most accurately describe it as “The Subway of Doom”. The scene is set completely underground in a maze of subway tracks lying along the tunnels. Inside the gloomy tunnels, you will find two trains that endlessly move through the area, changing their direction once they’ve reached the rubble at either end, searching for a way out that they will never find. Also inhabiting the tunnels are a rogue clan of creatures that constantly move in a pack, seeking out any surviving beings to devour. The monsters have come to know the trains, seeing them not as a threat, just as a common occurrence of the world they live in. As they wander endlessly, they will not pursue the trains, but stay out of the way as they have learned to coexist.

Steering Behaviors:

Flock implementation: The monsters all utilize separation, alignment, and cohesion as they move throughout the world with each other in a pack.

Steering Behaviors: I went with a hybrid of seeking and wandering on this project. Located throughout the level are a series of waypoints. The monsters will seek a waypoint and upon arriving at that point, a new adjacent point will be chosen at random for them to seek. It could be down a new tunnel or even back the way they just came from. This keeps their movement completely unpredictable. As with other flockers, obstacle avoidance is also implemented so that the monsters do not collide with the trains.

Advanced Steering Behavior 1: My first advanced steering behavior is simple path following seen in the subway trains that move throughout the level. Each train uses an array of points (labeled PATH_1 and PATH_2 in my code respectively) to seek along a path. As the train car passes within a specified distance of the current point, the index is incremented and the train will begin to seek the next point.

Advanced Steering Behavior 2: The second advanced steering behavior that I chose to implement was leader following. Each subway train in my level consists of three individual cars. The first one follows the path as described above while the second uses leader following to seek a point behind the leader. This effect is repeated in the third car which follows the second car. At the end of the track, when the train must start moving the other way, the third car becomes the leader and the second and first cars switch to follow mode.

Characters' response to other characters:

The flocking monsters will avoid the subway trains if they happen to be wandering in the same area as them so as to not bump into them. The trains do not change their behavior based on what the monsters are doing, only the other way around. I chose to make them behave this way because it would make more sense that a living being would actually react to an inanimate object in order to keep itself alive whereas only complex machines would be able to detect and avoid a collision.

Characters' response to the environment:

There aren't any other entities in the environment, moving or otherwise, so the characters do not react much with the environment. It is worth noting, however, that the entire level takes place "underground" or rather within the insides of game objects. It is necessary for the monsters and trains to stay inside the boundaries represented by the walls of the tunnels. Although they do this fairly well, this area is not without bugs. Occasionally, the monsters will "cut corners" when their waypoint changes and they have not yet reached the intersection that the waypoint was at.

Other notes:

You may press 'C' to toggle camera views in the program

Camera 1 – First person moveable view via First Person Controller

Camera 2 – Top down view of the entire subway system

Camera 3 – A camera which follows one of the monsters in the flock (smooth follow did not work, so I scripted a last minute implementation myself)

I tried a lot of new code implementations that we didn't really cover in any of my semesters of Game Design here. Since everything was added one after the other and coded until a point where it worked, the code may be very messy and/or be cluttered with unnecessary lines. I have done my best to comment all of my code to help it be easily understood, but there was simply not enough time to look for more efficient implementations. After this project is completed, however, I will continue to look at this code and actively search for ways to improve it. I wholeheartedly believe, however, that this was my best attempt at the scenario I described in the first paragraph of this document given the amount of time I had and my current knowledge of coding.

Resources:

Subway Tunnels and Track – Subway Level Kit by Christian Palmroos

<https://www.assetstore.unity3d.com/en/#!/content/18334>

Subway Trains – Subway Cars by Unity Technologies (Edited by me in Maya)

<https://www.assetstore.unity3d.com/en/#!/content/2609>

Flocking Monsters – Monster 3 by BÜMSTRÜM

<https://www.assetstore.unity3d.com/en/#!/content/48933>

Smooth Follow Camera Script – Taken from in-class exercises

Alternative Camera Script – Scripted by myself in response to smooth follow breaking 1 hour before I submitted this assignment