

Milestone 2: Interim Report

In this report, we will explain the progress that we went through after game idea pitch and formal proposal. Before we started the implementation, we considered all the feedback that we got and made arrangements accordingly. In this milestone, we focused on the fundamental game dynamics rather than the game design.

1. Task Progress: What have we implemented so far?

a. Level Scene:

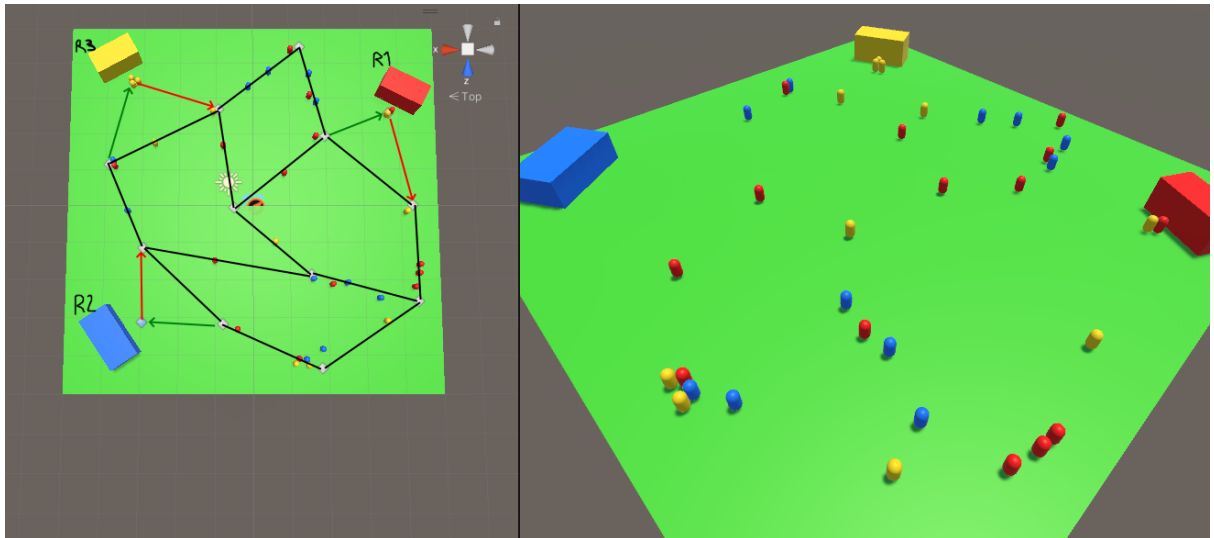
We started with a basic level setup which consists of capsules as visitors and rectangular prisms as rides to be able to implement the game dynamics. We gave 3 different colors to the rectangular prisms to represent different rides with different features and 3 different colored capsules to represent different visitor profiles with unique constraints. For simplicity we haven't assigned any profile to a color but for now visitors are more likely to prefer to go to matching color rides with theirs.

b. Drag and Drop:

We have implemented the Drag & Drop system using mouse clicking as the input method. We have defined the objects as "Interactable" and assigned them appropriate tags. This allows us to freely drag and drop objects as desired. Adding corresponding animations will ease the interaction.

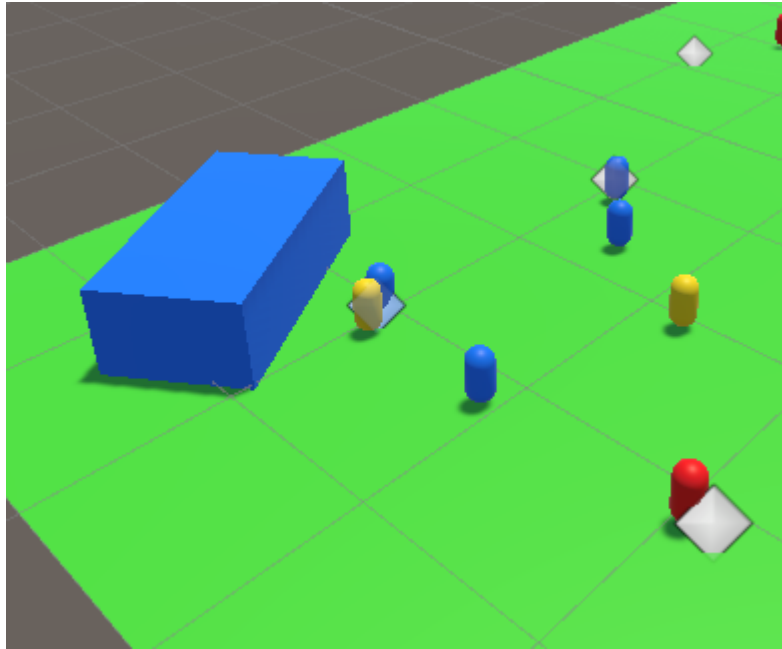
c. Crowd Simulation: Waypoints Map & NavMeshAgents

We have started implementing the crowd simulation with a basic graph-based system with waypoint nodes and ride nodes. A basic AIController script together with a NavMeshAgent component controls the navigation of the visitors. We can have as many visitors as we want that can stroll around the environment, wait in the line and take a ride.



As a default behavior, a visitor traverses along the nodes of the graph in a randomized way. We have implemented this as NavMeshAgent destinations which change whenever they reach a new node along the map and added some factor of randomization by sampling a point near the node coordinates for more natural crowd behavior. And when the visitors reach a ride, they make the probabilistic decision to enter or not, where the probability is based on their current satisfaction level, their visitor type and the type of the ride. If they decide to enter, they check for the availability of the ride.

Currently each visitor that enters a ride, spends time inside for the run time of the ride which is indicated in the ride variables. For the future versions, we would be changing this to a system where each ride takes people from the queue and let people enter and exit in batches (indicated by ride capacity) for each run of the ride, which would be more realistic and more akin to a real-world theme park ride.



While we use the default unity NavMeshAgent with an accompanying AIController script for the basic crowd simulation, we have done research on how we can make it more advanced. Behavior profiles of the visitors will be expanded when we add events to the game in the future versions. We will simulate the crowd Reynolds flocking algorithm together with NavMeshAgent. Our visitors will follow three main rules which are turn towards the average heading of the group, turn towards the average center of the group and avoid those around you. Thus, the crowd of visitors behave like particles in a fluid.

2. Challenges

In this milestone, we didn't face many challenges. The progress was generally smooth. The only thing we spent time to solve is our drag and drop functionality on the NavMeshAgents. At the beginning of the game it was working. However, after a couple of spawns of visitors the drag and drop functionality seemed to stop working for some of the visitors. We worked on the code, debugged and changed the code but it still didn't seem to work. After more trials we solved this issue by removing Rigidbody from the object that also has NavMeshAgent.

3. What have we improved after the feedback?

Majority of the feedback that we got were about the complexity/overwhelming level of the game. Most of the people mentioned that managing the amusement park and regulating the visitors at the same time can be overwhelming for the player. We decided to keep the amusement park management part not too confusing and make it intuitive as much as possible, such as if you are adding an upgrade to a ride to earn more money and taking risk then you can expect to have increased difficulty in controlling the visitors which lead to accidents. Otherwise, players can also choose to play safe to not make a big profit.

We also liked the suggestions about adding fun updates to amusement parks which make the game more interesting, for example adding functionalities to bring a politician or celebrity.