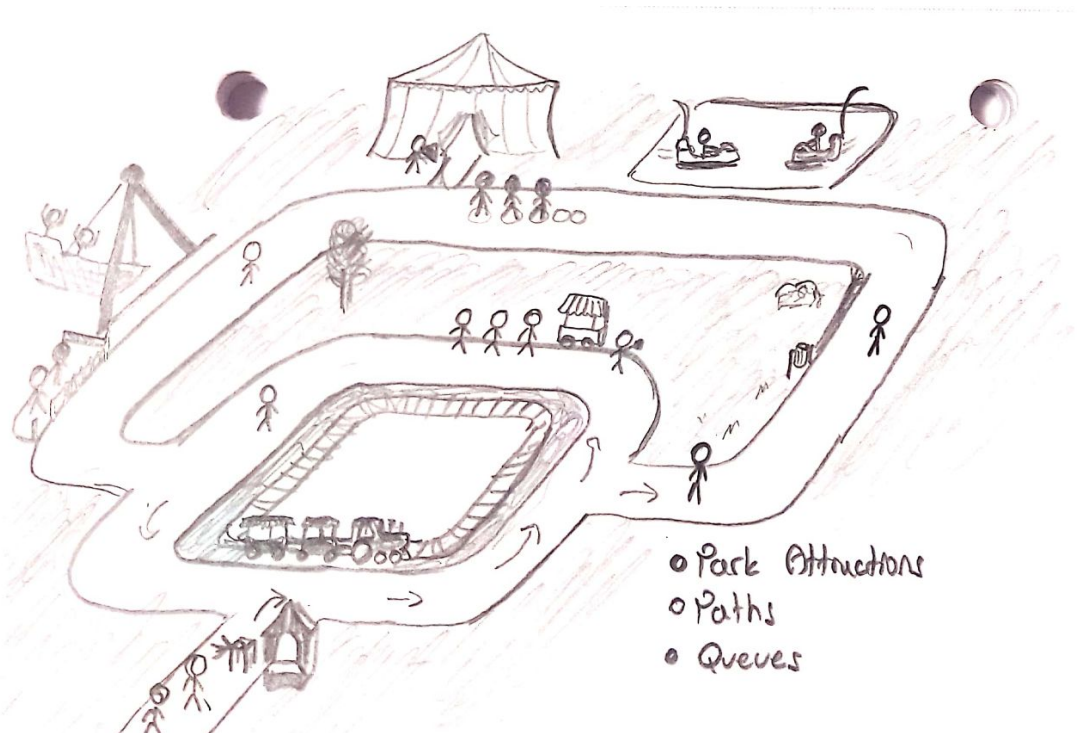


# Milestone 1: Formal Game Proposal

## 1. Game Description

For the theme “roller-coaster” we are going to develop an amusement park management game. However, our game draws inspiration from several genres such as tower defense, simulation/management and tower defense games. The player’s goal is to manage an amusement park and try to keep waves of incoming visitors satisfied. As the game progresses, players will have the opportunity to upgrade and modify the existing amusement park rides/attractions, add new ones, and also manually interact with both the visitors and the rides when problems arise in the park. While doing this the player must consider the different types of visitors and their preferences and characteristics. Satisfied customers let players earn more money and develop the park further, while failing to satisfy customers for a while ultimately results in game over.

We plan to create a 2.5D isometric game, while doing so we will be using 2d and 3d assets from various sources online. The game is mainly designed for mouse and keyboard controls and is being developed for PC using Unity. There is also the possibility of porting the game to mobile devices, which is part of the higher target (discussed further in the coming parts).



**Fig 1:** General layout of the game.



**Fig 2:** Point of view of the game

## 1.1 Gameplay

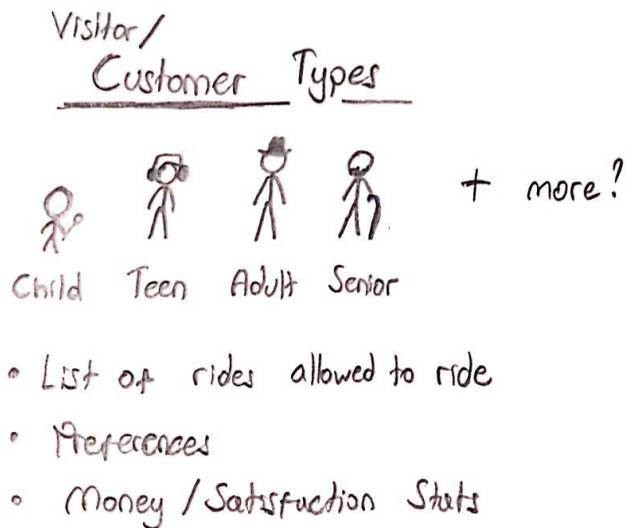
The game features a low skill floor and simple gameplay, allowing players of all experience levels to enjoy it. Players are tasked with strategically upgrading and placing new amusement park attractions to keep waves of visitors that approach satisfied. The game encourages manual interaction with the rides and visitors, allowing players to fine-tune their rides to handle the ever-increasing number of visitors. Some improvements, while increasing the visitor capacity, entertainment factor, or speed of the ride, may also increase the risk of accidents, reduce durability, and make the ride more prone to breaking.

The player can also interact with the visitors directly to achieve various goals, such as steering them away from an overcrowded ride and directing them to others or seeing a child who is not allowed to ride a certain amusement park attraction waiting in line, and making them leave the queue.

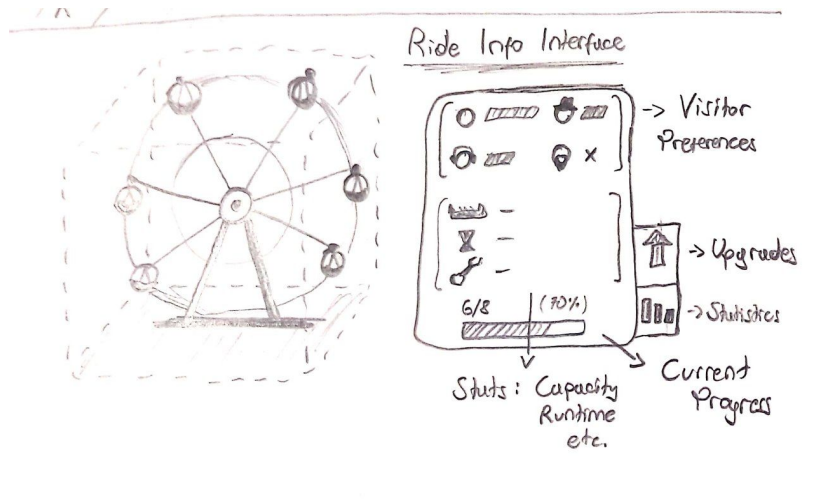
Overall, the game will offer an engaging and strategic gameplay experience during the building and upgrading phase, coupled with fun and interactive gameplay mechanics during the visitor phase.

## 1.2 Visitors and Amusement Park Attractions

In the game, visitors come in different age groups, such as *children*, *teens*, *adults*, and *seniors*, each with their own preferences and limitations when it comes to park attractions. The game also features a variety of ride types, each with different *capacities*, *run times*, and *fun factors*, that appeal differently to each visitor type. To add to the fun, upgrades and modifications are available for each ride type, allowing players to customize them to cater to the needs and preferences of their guests. By understanding these differences, players can create a more enjoyable experience for all their virtual visitors and improve the overall success of their amusement park



**Fig 3:** Initial character design.



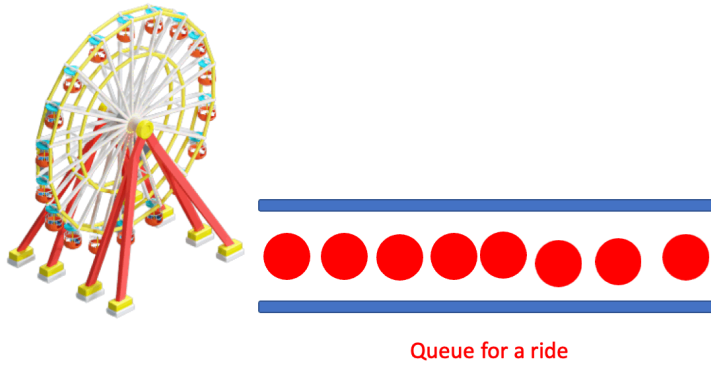
**Fig 4: Ride Information UI.**

## 2. Technical Achievement

For the technical achievement in our game we choose crowd simulation, since our game narrative consists of a crowd of visitors in the amusement park. In the game, if an accident such as a fire happens in one or more of the rides, it is expected that the crowd should act on it properly and realistically such as running away from the place of the incident, seeking help or staying in the place to help.

First goal is to implement a rule-based approach to the crowd simulation. We will define specific rules for how visitors should behave in response to an accident which will be applied to all visitors. For instance, going to the specific ride, waiting in the queue, running away from the accident location, avoiding stepping over other visitors, or for the park staff, clearing the area or calling for help.

After we successfully implement the rule-based and if we still have time to approve we can turn it into an agent-based approach to crowd simulation. For this case we will define each agent's own set of rules to determine how an agent behaves and responds to an accident or in general to its environment. Ultimate goal would be that agents can communicate and interact with each other.



### 3. "Big Idea" Bullseye

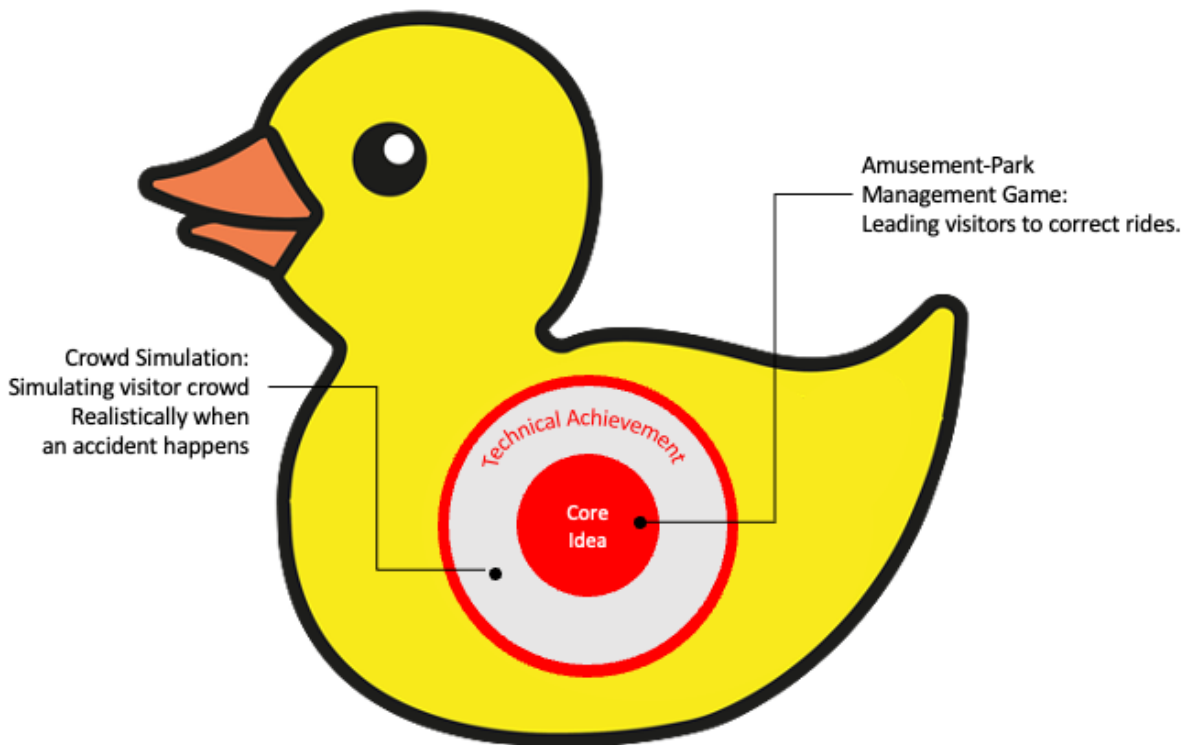


Fig 5: Bullseye representation.

## 4. Development Schedule

1. **Functional Minimum:** The first stage of game development involves creating basic playable interactions. This includes a day and night cycle, at least three types of customers (Old, Young/Child, Disabled?), and three types of amusement park toys. The players should be able to interact with characters by clicking, pulling, and pushing them. The game should also feature a basic economy, where player satisfaction increases when they play with the toys. The game should also have a suitable background music.
2. **Low Target:** The next stage of game development involves creating events such as accidents, deaths, and chaos, which require players to make meaningful decisions to keep the deaths low. This stage also involves incorporating shaders and improving the game's graphics, making it visually appealing.
3. **Desirable Target:** This stage involves adding as many toys and character types as possible, and making the toys upgradable. Upgraded toys may lead to success or failure, adding more depth to the game's economy.
4. **High Target:** This stage involves customizing the park or introducing procedural generation. Customization allows players to personalize their amusement park, while procedural generation creates an unpredictable game environment that keeps players engaged.
5. **Extras:** Adding a dialogue system and story to the game adds another layer of depth and immersion, enhancing the player's experience.

In conclusion, the game development schedule includes creating basic playable interactions, incorporating events and improving the game's graphics, adding toys and characters, introducing park customization or procedural generation, and adding a story with a dialogue system. By following this development schedule, we can create an immersive and engaging amusement park game that keeps players entertained for hours.



- In addition, the game should be playable for everyone, from casual gamers to hardcore enthusiasts. The difficulty level should be balanced to ensure that both types of players can enjoy the game without feeling overwhelmed or bored.
- Another important criterion is the level of crowd simulation. The game should have a believable level of crowd simulation, where visitors move around the park in a natural way and react to the events in the park. This adds to the immersion and realism of the game, making it more enjoyable for players.
- Finally, as a high target, the amusement park simulation should be comprehensive, with a wide range of rides, attractions, and amenities available for players to build and customize. The game should allow players to create their own amusement parks with unique layouts and designs and offer plenty of upgrades and modifications to keep things interesting.

## Artistic Style:

### Representative rides:

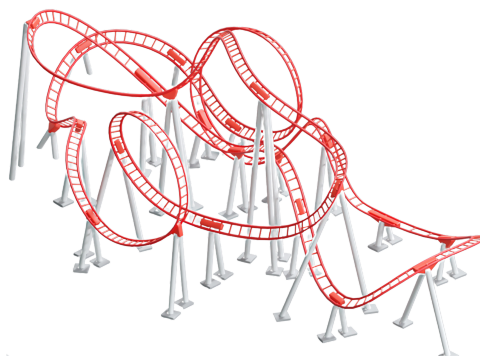
Each ride has its own features and visitor constraints.



Ride 1:

Suitable for ages 18-60

Should not have any heart disease.



Ride 2:

Suitable for age 15+.

Should not have any heart disease.



Ride 3:

Suitable for all ages.

No limitation for disease.

**Fig 6:** Ride ideas[1]



[1]<https://www.artstation.com/artwork/DAX0o9>

**Visitor Profile:**



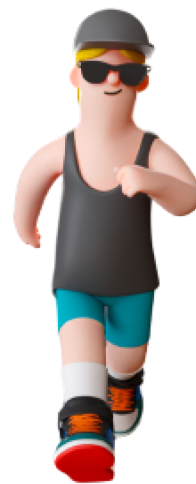
Young adult  
(18)

No health problems



Old Adult

Has health problems



Child (younger than

No health problems

**Fig 6:** Character ideas[1]

[1]<https://www.artstation.com/artwork/DAX0o9>