Iliana Marion (ijavier@pdx.edu)
CS523

Shapeland

A simulation project proposal

Background:

“Shapeland” is a fictional theme park created by the youtube channel Defunctland in his video “Disney's FastPass: A Complicated History” (https://www.youtube.com/watch?v=9yjZpBq1XBE). The purpose of Shapeland is to simulate a theme park with various activities and guest types to see the effect different forms of line queueing (mainly the use and different implementations of a “Fast Pass” system) have on ride wait times (i.e. Fast pass vs Standby line times, How many rides each type of guest was able to ride in a day, how many times guests found a queue time too long and left the line, the total amount of time in a day guests spent in lines, and so on). In addition to the core variable of which Fast Pass system is in use, there are a variety of attributes about the park and guests which can be changed by the user. These attributes include ride popularity, ride service rate, guest activity preferences, different guests’ tolerance for various wait times, and so.

In the video, Mr. Defunctland says he would need to hire an industrial engineer to create a complex simulation to fully understand the ramifications of different Fast Pass implementations. For this project, we will be acting as this industrial engineer, and our program will be written in Rust.

Project Parameters and Vision:

For this simulation, we will simplify Mr. Defunctland’s project by only implementing 1 version of the Fast Pass system- the original paper Fast Pass system. With luck, the user will be able to modify most parameters described in the simulation based on a configuration file or command line input.

When a guest is not currently in line, on a ride, or doing an activity, they will “flip a coin” to determine what to do next- ride a ride, or do an activity- with the coin being weighted based on the guest’s subtype’s preference.
If the guest chooses to do an activity, a random number between 15 and 45 will determine how long it takes them to complete said activity. Their state will return to “idle” after this amount of time.

Iliana Marion (ijavier@pdx.edu)
CS523

If the guest chooses to ride a ride, another coin flip will be initiated to determine which ride they will try to ride, weighted based on the ride's popularity. When the ride is chosen, the guest will then "look" at the ride's current wait time. If the wait time acceptable (per guest subtype preference), they will join the standby line and be "In Line" for the posted amount of time.
If the wait time is too long for the guest's subtype's preference, they will see if Fast Pass is available for the ride. If it is, they will take a paper fast pass with a given return time, and will return to 'idle' state. However, in deciding to do an activity or ride, they will now make sure it ends before the posted return time. At the return time, they will return the ride and join the Fast Pass line. Guests will be "in line" state until the service rate of the ride catches up to their placement in the line.
If Fast Pass is not available, the guest will balk, and return to 'idle' state to make a new choice.

Rides will keep track of the amount of people in their Standby and Fast Pass lines and determine the wait time based on the amount of people and the ride's service rate.

At the end of the program (i.e. 1 day of Shapeland time), the statistics for rides, wait times, and guest outcomes will be outputted to the user.

Object design:

Rides:
Ride objects have the following attributes:

- Service rate: The amount of guests a ride can accommodate per hour
- Popularity: How popular a ride is to the guest population (1-10)

Only the most popular 3 rides will have Fast Pass available

Guests:
Guests will include multiple subtypes (final number and names tbd)
Each subtype will have difference preference values for the following:

- Start time: time that the guest will arrive at the park
- Stay duration: the amount of time in hours the guest would like to stay at the park for
- Ride preference: Weighted preference of the guest for rides over doing activities
- Balking point: the amount of time a guest is willing to stand in a standby line (15 minutes --> 3 hours).
- State: Current state of the guest (idle, in line, on ride, doing activity)
- Wait time: total time guest has spent in the "in line" state
- Rides ridden: struct of all of the rides in shapeland and how many times the guest has ridden them
- Activities: Amount of time and the number of activities the guest has done

Park:
Global state of the park that users can change:

- Operational hours: Defaults: the park will open at 8am and close at 6pm
- Amount of rides and their parameters

Iliana Marion (ijavier@pdx.edu)
CS523

- Total number of guests and subtypes within the park for the day

Limitations:

- Rides are assumed to be at 100% operational capacity for the entire duration of the simulation (no downtime, no delays)
- Time not covered by the above parameters (Guests walking to and from destinations, breaks, miscellaneous distractions, etc) is not included in this simulation (aka the Shapeland imagineers have invented instantaneous teleportation and need fulfillment. Hooray!)
- Balking point will be the same for a subtype across all rides (a ride being more popular does not mean the guest is willing to wait longer)

Concerns:

- Even with this simplified version of the Shapeland simulation, I am concerned with my ability to complete the project with all of the given parameters. To start, the amount of guests and rides will be limited.
- Would like a more interactive way to display the final statistics and / or something fun to show the user as the simulation runs throughout the “day”. Don’t know if I can accomplish that that kind of animation.

Git repo: https://github.com/ijavier-psu/Shapeland