Theme Park Queuing Simulation - a Multi Agent Based Model

Proag Tanvee, 2422814
University of Mauritius

Abstract

Agent-based modeling enables the analysis of simulating complex systems with interacting individuals. This paper demonstrates the application of agent-based modeling, using AnyLogic, to analyze visitor behavior in a theme park setting. The model captures the decision-making processes of diverse visitor types, including regular, VIP and impatient individuals, as they navigate the park's attractions. The simulation explores the emergent dynamics of queue formation, abandonment and ride selection, seeking to observe and understand theme park operations.

1. Introduction

1.1 Theme Park Queuing

A theme park is a large entertainment space where people with different preferences can explore imaginative worlds through rides and attractions. Theme parks are major economic contributors, earning billions yearly worldwide. Visitors aim to enjoy as many rides as possible but waiting in line which is a necessity due to ride capacity limits, might be equivalent to time wasted. Long waits can lead to dissatisfaction, so efficient queues are critical for visitor happiness. Park operations also benefit from an efficient queuing model. This simulation models queuing dynamics at a theme park (loosely based on Disneyland California) using a multi-agent-based system, where visitors exhibit distinct behaviors. The aim is to analyze the crowd management processes of the theme park environment by representing the stochastic processes of visitor arrivals, service times of rides and resource allocation. The queuing behaviour is analysed under operational scenarios. Then, factors that influence queue lengths, waiting times and system performance are observed.

1.2 Data Calibration

The Disneyland California Adventure queue time [https://queue-times.com/parks/16] data used in this model is based on live statistics from the well-known park in the United States. These statistics are continuously updated and include real-time queue durations, ride performance metrics, average crowd levels, crowd forecasts and historical attendance records. For the simulation, the live queue times and the peak hours (March 2025) are used; The maximum queue time determines the threshold at which visitors decide to leave a queue while peak-hour schedules define visitor arrival rates. However, the data is not directly replicated in the model as Disneyland California Adventure has many rides as compared to the model simulation, which would excessively increase complexity if fully mirrored. Instead, the ratios and relationships within the data (e.g., relative wait times between rides) are preserved. For trademark neutrality and to keep the model abstract, ride names are altered — for example, Disneyland$^{\text{TM}}$'s *Astro Orbitor* is referred to as *Space Twist* in the model.

1.3 Conceptual Model

This multi-agent system models visitor behavior in a theme park with five rides, featuring three distinct types of agents: regular visitors, VIP visitors and impatient visitors. Regular visitors follow standard queuing rules while VIP visitors possess priority access to bypass queues entirely. Impatient visitors, distinguished by their low tolerance for delays, dynamically abandon queues if wait times exceed their thresholds.

Each ride operates with a fixed capacity (maximum simultaneous visitors), a delay time (duration of the ride experience) and a first-in-first-out (FIFO) queue system. Queues are initially formed in the order of arrival but VIP visitors disrupt this order by moving directly to the front. After completing a ride, visitors proceed to another attraction to join a new queue for their next experience or exit the park.

The model incorporates a patience threshold which is a dynamic limit that determines when visitors abandon their current queue. Most visitors tolerate moderate wait times but impatient agents act based on whether their current wait surpasses a ride's average expected wait time. If so, they leave the queue (provided they are not already at the front) and seek the shortest alternative queue. Regular visitors may also switch queues but do so less frequently due to higher patience levels. These decisions are driven by real-time comparisons of queue lengths and individual tolerance for delays.

The interaction between VIP interruptions, impatient abandonments and FIFO ordering creates dynamic fluctuations in queue lengths and wait times. For instance, VIP priority is expected to shorten wait times for privileged visitors but indirectly increase delays for others while impatient agents might redistribute themselves to shorter queues might introduce unpredictability in ride demand. The resulting emergent complexity reflects real theme parks where visitors' varied behaviors influence the park's daily operations.

Computational Model

2. Modeling

2.1 The world

The simulation runs for 14 hours (8:00 AM to 10:00 PM) with each time unit representing one minute. The park layout includes five attractions positioned within an enclosed area. Visitors enter through an arrival gate and exit via the same gate. Each attraction features a curved zigzag queue with a maximum capacity where visitors line up. Each attraction include a capacity (the number of visitors the ride can accommodate at once), a delay time (the duration visitors spend experiencing the ride) and a timeout (the maximum time visitors can wait in the queue before leaving). While in the queue, visitors remain in the designated waiting area. After the ride ends (the delay length of time), they either depart the park or join another queue based on probability.

2.2 Arrival rate

The visitor arrival rate follows a predefined schedule spanning a 14-hour operational period (8:00 AM to 8:00 PM), repeating daily. Arrivals peak between 9:00 AM and 11:00 AM, followed by a gradual decline and a sharper reduction after 4:00 PM. Figure 1 illustrates the gradient of arrival rate fluctuations throughout the day.

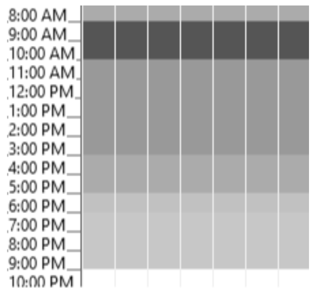

Figure 1: Arrival rate - Schedule

3. Agents

Visitors in the park are modeled as autonomous agents, simulating human-like decision-making. All agents inherit from a Base Agent class which defines core attributes such as `priority` (determining queue access rights) with a default value of 1 and `patience` (time tolerance before abandoning a queue) with a default value of infinity. The three agents types are Regular Visitor, VIP Visitor and Queue Hopper Visitor.

3.1 Regular Visitors

The `Regular Visitor` Agent type extends the Base Agent.

1. **Arrival Rate** - Schedule
   Regular visitors follow the schedule that defines the arrival rates. From 8:00 to 22:00, it gradually changes from 1.5 to 3.0 to 1.8 to 1.5 to 1.0.

2. **Priority** - 1
   Regular visitors have no privileges and retain the base priority value of 1.

3. **Patience** - 1.0
   Regular visitors exhibit high patience (triangular distribution with parameters minimum of 0.9, maximum of 1.1 and mode of 1.0) as they are willing to endure long wait times to ensure they experience attractions even at the cost of time. They prioritize ride access over time efficiency.

3.2 VIP Visitors

The `VIP Visitor` Agent type extends the Base Agent. VIPs are privileged visitors who override FIFO queues by skipping ahead of all other agent types.

1. **Arrival Rate** - 0.1*Schedule
   VIP visitors follow the same arrival schedule as regular visitors but at 10% of the rate (0.1 $\times$ schedule rate) to reflect their significantly lower frequency within the park population.

2. **Priority** - 100
   VIP visitors are granted elevated priority status so that they can bypass queues and move directly to the front to reflect their privileged access.

3. **Patience** - 0.8
   VIP visitors exhibit moderate patience (triangular distribution with parameters minimum of 0.7, maximum of 0.9 and mode of 0.8) as they are willing to endure long wait times to ensure they experience attractions. However, they are not as patient as regular visitors.

3.3 Queue Hopper Visitors

The `Queue Hopper` Agent type extends the BASE AGENT. Queue Hoppers exhibit abnormal behavior by abandoning queues prematurely when their patience threshold is exceeded even if it means forfeiting the ride experience.

1. **Arrival Rate** - 0.2*Schedule
   Queue Hoppers follows the same arrival schedule as regular visitors but at 20% of the rate (0.2 $\times$ schedule rate) to reflect their significantly lower frequency within the park population.

2. **Priority** - 1
   Queue Hoppers have no privileges and retain the base priority value of 1.

3. **Patience** - 0.3
   Queue Hoppers exhibit very low patience (triangular distribution with parameters minimum of 0.2, maximum of 0.5 and mode of 0.3) as they they prioritize time efficiency over ride completion, frequently abandoning queues to seek shorter lines or exit the system entirely.

Basically, Regular visitors follow FIFO rules, VIPs prioritize their own interests by cutting the queue and Queue Hoppers dynamically adjust their behavior based on patience thresholds. They have distinct goals (e.g., minimizing wait time, maximizing enjoyment). They have a localized focus as their actions are driven by personal objectives rather than system-wide optimization.

3.4 Coherence

The agents interact indirectly through their impact on the shared environment i.e. the queue state. VIPs prioritize self-interest over collective efficiency while Queue hoppers act independently with no coordination to balance workloads. Their behaviors influence each other's experiences even without direct communication. VIP visitors disrupt the FIFO order for Regular visitors by cutting the line, increasing wait times for others. Queue hoppers leaving due to impatience reduce queue length, indirectly affecting others' wait times. If hoppers leave frequently, Regular visitors might experience shorter waits, creating a self-regulating system. However, VIP dominance could destabilize the queue, causing cascading impatience i.e. Regulars consequently becoming hoppers.

4. Agent-Based Modeling

4.1 Model flow

[Refer to the AnyLogic model] The flow begins with three sources: the `VisitorSource` which generates regular visitors; the `VisitorVIPSource` which introduces VIP visitors; and the `QueueHopperSource` which represents impatient visitors with low tolerance for delays. These visitors are directed to the `selectRide` decision point, where they choose one of five available rides—*Space Twist*, *Haunted House*, *Ferris Wheel*, *Swing* or *Photobooth* depending on the queue size. Once a ride is selected, visitors move to the corresponding ride location (e.g., `moveToSpaceTwist`, `moveToHauntedHouse`, etc.) and join its queue (e.g., `queueSpaceTwist`, `queueHauntedHouse`, etc.). Each queue operates on a first-in-first-out system but VIP visitors bypass this order and go directly to the front. Impatient visitors evaluate their wait times using a dynamic timeout mechanism (`TimeOut`) and may abandon the queue if their patience threshold is exceeded. If they leave a queue, they seek an alternative ride with shorter wait time. Visitors who remain in the queue eventually reach the front and begin their ride experience (e.g., `SpaceTwistStart`, `FerrisWheelStart`). The rides have fixed delay times (e.g., `delaySpaceTwist`, `delaySwing`) representing their duration. After completing a ride, visitors proceed to the next decision point (`selectOutput`) where they decide whether to visit another attraction or leave the park. Those who choose to exit move to the departure area (`moveToDeparture`) and leave the system via the sink block.

4.2 Defining the complex system

Leverage points - Small changes can disproportionately shift system behavior. For example, if the arrival rate of VIP visitors is increased by 5%, it could trigger drastic outcomes as more Regular visitors might reach their patience threshold.

Path Dependence - Every time a visitor chooses which ride to go to, it is based on the length of its queue. VIP line-cutting at a ride early in the day could form congestion.

Non-Linearity and Dynamics - Doubling VIP arrivals may quadruple Regular visitor wait times. So, it is a non-linear relationship. Queue Hoppers abandoning a ride reduces its queue length, making it suddenly attractive again. This implies that optimizing one parameter (e.g., ride capacity) may worsen others (e.g., overall fairness).

Robustness - Queue Hopper redistribution balances demand. However, ride delays or VIP surges cause cascading failures.

Diversity and Heterogeneity - Regular, VIP and Queue Hoppers compete for resources, creating realistic crowd dynamics. Triangular patience distributions mimic real-world behavioral diversity.

Interconnectedness and Interactions - VIPs cutting lines increase Regular visitors' wait times → Regulars migrate to other rides → unintended congestion elsewhere.

Representation - Agents with patience thresholds and priority values capture human decision-making.

4.3 Randomness v/s Determinism

Agent-based models integrate both deterministic and stochastic elements to reflect real-world decision-making. Unlike purely equation-driven models, ABMs allow agents to follow rules while incorporating controlled randomness.

Deterministic Elements

The arrival rate of visitors is deterministic too since it follows a predefined schedule which creates predictable initial demand.

Stochastic Elements

For ride revisitation, a probabilistic rule is used to decide whether visitors re-enter the system after completing a ride: a probability of 0.25 (25% revisit rate). It is calibrated to balance ride utilization and system exit rate. This prevents overcrowding (low revisit rate) while maintaining steady ride demand.

Queue selection follows a probabilistic rule, giving shorter queues a higher probability of being selected. This simulates the tendency to minimize wait times while still allowing for some randomness in destination choice.

Emergent Synergy

Using both fixed rules (determinism) and probabilistic behaviors (randomness) reflects the dynamics of a real theme park. Queue-length-based choices create predictable spatial visitor distributions. Stochastic revisitation introduces variability in ride demand over time.

5. Dynamics and feedbacks

The model's behavior changes over time as visitors move through the theme park. Visitors enter and leave queues depending on whether they are regular, VIP or impatient. Impatient visitors, managed by the `TimeOut` port of the `Queue` block, might leave a queue if the wait is too long. All visitors eventually complete rides, which moves them through the park's system. Meanwhile, the number of people in each queue goes up and down as visitors arrive, VIPs cut in line and impatient visitors leave, creating a constantly changing queuing environment.

The model also features feedback loops. The length of the queues influences which rides visitors choose at the `selectRide` block; shorter queues are more attractive. Additionally, the VIP visitors affect the wait times of regular visitors. The behavior of impatient visitors, who leave queues when they are too long, also influences the queue lengths at other rides as they redistribute themselves. Because visitors' decisions about which ride to attend change as the wait times change, the dynamic selection of which ride to attend creates a behaviourally related feedback.

6. Results

A slider is set on the user display to let the user change the arrival rate of the visitors to display its impact on the queue lengths. `TimeMeasureStart` and `TimeMeasureEnd` blocks are added in the flow to measure time. The time taken (queue + delay) at each ride, the time taken (queue + delay) at all rides combined, the time taken by each agent in the park and the time taken by all agents combined in the park are displayed through histograms. The results are then summarised across multiple runs. The figures below show the histogram for the time taken by agents at all rides combined and the histogram for the time spent in the park by each agent type.

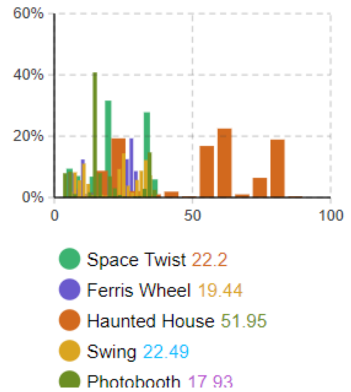

Figure 2: Time taken by agents at all rides combined

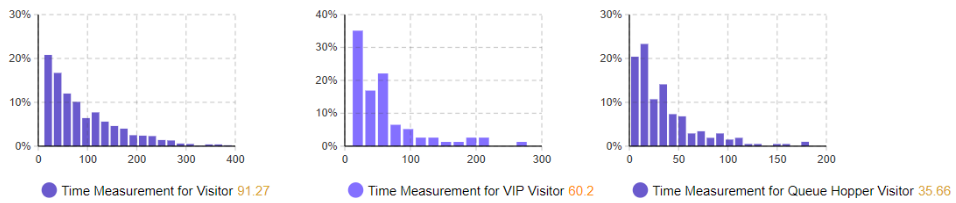

Figure 3: Time spent in the park by each agent type

It is observed that the average time taken at the Haunted House is the highest. This was to be expected as this ride has the highest delay with the smallest capacity when compared to others. Still, the queues stay balanced with the help of the `SelectOutput` block that allows visitors to be more attracted to shorter queues. It is also observed that VIP visitors spend significantly less time as compared to regular visitors in the park which was to be expected since they can cut queues to get to the front. Queue hoppers spend even less time since they abandon their queues and keep moving to the next until they potentially leave the park as they prioritize time efficiency over enjoyment of rides.

7. Evaluation

7.1 Verification

Verification increases confidence that the code correctly implements the intended model formulation. Verification was made in AnyLogic as so: The connections between the blocks were visually inspected to ensure the correct flow of agents. The injection rates of the source blocks were tested to ensure that the model injects the correct number of visitors into the simulation. Unit tests were conducted to verify that VIP visitors bypass the queue and proceed directly to the ride service. Debugging was used to trace the behavior of impatient visitors through the `TimeOut` port confirming that they abandon the queue when wait times exceed their threshold. [`traceln(agent.patience)` etc.] Finally, the animation was used to visually confirm that VIP visitors bypass the standard queues.

7.2 External validation

The model's predicted queue length distributions are compared to historical queue length data from the Disneyland California park, revealing a reasonable agreement between the two. Further validations can be done. A survey on how long visitors are can wait in line for different types of attractions can be done and the patience thresholds used in the model can be validated against the survey results.

8. Challenges

A key challenge in modeling the theme park queue was accurately representing visitor behaviors, such as priority handling and impatience, while maintaining a robust system. Initially, a `Seize` block was considered for managing queue interactions but it introduced unnecessary complexity. Furthermore, the requirement of a corresponding `Release` block added additional overhead, making it an unsuitable choice for this scenario where explicit resource allocation was not required.

Instead, a simpler approach was implemented using a `Queue` block configured to be priority-based via the agents' priority factor. VIP Visitors were assigned a higher priority, allowing them to move ahead in the queue. Visitors and Hopper Visitors had the same priority, ensuring a FIFO order. To model impatience, a patience factor was applied as a multiplication factor to the Queue block's timeout. Hopper Visitors, with a patience factor of approximately 0.3, had significantly reduced timeouts, leading them to exit the queue earlier.

Initially, `SelectOutput` blocks were considered to simulate Queue Hopper Visitor behavior, but this approach proved to be overly complex. Instead, leveraging the `Queue timeout` property provided a more direct and simple solution. Fine-tuning the agent factors was essential to ensure that Queue Hopper Visitors did not exit too frequently while maintaining fair queue dynamics for regular Visitors.

9. Conclusion

This model has allowed the observation of the behavior and decision-making of agents with distinct characteristics and behaviors. It showed how regular visitors, VIP visitors and impatient visitors interact with the park environment including their responses to queue lengths, wait times and ride selection dynamics. By simulating these interactions, the model showed the emergent complexities that arise from individual decisions such as queue disruptions caused by VIP priority access and the redistribution of demand due to impatient visitors abandoning queues. The simulation showed theme park operations such as fluctuating queue lengths and varying visitor flow patterns. This result can be used for optimizing park management strategies to improve visitor satisfaction and to balance resource allocation across attractions. However, the model's limitations such as its simplified assumptions about visitor behavior and static ride parameters suggest areas for future refinement. Expanding the model to include more realistic factors like ride popularity dynamics, spatial layout effects and additional visitor types would make it more applicable to real-world scenarios.

10. Limitations

The visitor behavior in the model is simplified as it assumes that visitors make decisions solely based on queue lengths and patience thresholds. It does not account for other factors that influence real-world visitor behavior such as preferences for specific rides, group dynamics or external stimuli like weather or promotions. Also, each ride has fixed capacity and delay times which may not reflect real-world variability. For example, ride delays could be influenced by operational issues or maintenance requirements and capacity might fluctuate due to safety protocols or staffing constraints. Additionally, there is a limited number of agent types. While the model includes regular visitors, VIP visitors and impatient visitors, it does not account for other potential visitor types (e.g., families with children or visitors with disabilities) who may also exhibit distinct behaviors. Finally, the model only simulates visitor behavior over a single day. Long-term operational trends could have been observed if it was replicated for a longer period of time.

11. Guideline for Simulation

11.1 Simulation Platform

The computational model is developed in **AnyLogic PLE 8** in Java which is an onject-oriented programming language.

11.2 Simulation Controls

1. Click the Run button to start the simulation.
2. Observe agent behavior in queues and rides during runtime.
3. Use the slider to adjust the visitor arrival rate in real-time.

11.3 Visualization Tools

- The Flow Diagram tracks the cumulative visitor count at each simulation step.
- The Histograms display real-time counts of visitors.