



Satellite Tycoon: Modeling Economic Competition in the Business of P-LEO Constellations

Davide Guzzetti*, **Daniel R. Tauritz****, **Rehman Qureshi[§]**, **Cody Roberts^{§§}**, **Manuel Indaco[§]**, **Lucy Bone^{§§}**, and **Emily Kimbrel[‡]**

With the advancement of communications technology, lower launch costs, and development of small satellites, the satellite internet marketplace is becoming a complex and fast-expanding sector. While modeling of individual satellite constellations and comparisons of relative technical performance have been conducted, modeling the competition between satellite internet providers and their complex strategies remains largely unexplored. This project aims to model the dynamics of long-term market strategies and competition between different satellite internet providers by framing the problem as a multi-player, strategy-simulation game. We use a mixture of modeling and gamification to create the prototype of a game in which players construct and operate proliferated low-Earth orbit (P-LEO) constellations that provide satellite internet to simulated customers. In the gamified environment, players can take four basic actions: ordering satellites, building ground stations, launching satellites into orbit, and setting a monthly internet subscription price for their customers. The framework also includes a primitive technology tree where players can explore the impact of new technologies for better constellation performance and customer acquisition as emergent game strategies. Like previous strategy simulation games designed for research, a server authoritative software architecture is used to connect a hosted web server to multiple clients. The envisioned outcome of this project can be used as an educational tool, a business decision support system, a simulation environment for national security studies, an AI test environment, or simply a recreational game.

keywords: Proliferated Low Earth Orbit Constellations, Adversarial Strategies, Business Game, Constellation Modeling

1. Introduction

During the last century, the space domain was largely decoupled from economic and adversarial competition. Such decoupling enabled government agencies to implement relatively straightforward logistic and security paradigms for space applications. However, the space domain is now undergoing a radical transformation, rapidly becoming a complex territory of dispute where unknown logistic dynamics are emerging. Moreover, private sector expertise has played an outsized role in pushing the boundaries of space technology, for instance in the establishment of large satellite constellations in low-Earth orbit, also known as Proliferated Low Earth Orbit (P-LEO) constellations. P-LEO constellations are an example application where coupled techno-logistic-economic dynamics remain unexplored, but government units are asked to answer pressing strategic questions about

the sustainability and robustness of P-LEO assets operated by the private sector. While P-LEO constellations display hardware resilience avoiding single-point failure, it is yet to be proven that the corresponding market dynamics display the same type of resilience. Despite the large number of ventures, there are only a handful of players who are succeeding in deploying P-LEO constellations. Adversaries have already demonstrated the ability to utilize coercive economic measures to their advantage [1]. Understanding the interplay of technology progress, logistics and economics within P-LEO constellations to aid decision makers, requires identifying novel intersections and synergies across several fields at the fundamental research level.

Historically, the commercial satellite communications industry has been separated into two categories: unidirectional broadcasting and

* Department of Aerospace Engineering, Auburn University, United States, guzzetti@auburn.edu

** Department of Computer Science and Software Engineering, Auburn University, United States, dtauritz@acm.org

§ Department of Aerospace Engineering, Auburn University, United States, rsq0001@auburn.edu

§§ Department of Computer Science and Software Engineering, Auburn University, United States, car0106@auburn.edu

§ Department of Aerospace Engineering, Auburn University, United States, mzi0022@auburn.edu

§§ Department of Aerospace Engineering, Auburn University, United States, leb0078@auburn.edu

‡ Department of Computer Science and Software Engineering, Auburn University, United States, eak0043@auburn.edu



bidirectional broadband internet services. Companies such as Iridium, Globalstar, Teledesic, and Orbcomm all launched P-LEO constellations in the 1990s, but had difficulty gaining market share due to their poor performance and limited use cases [2, 3, 4]. However, due to changing consumer tastes, advances in key technologies, and the popularization of internet streaming, existing constellation operators have shifted away from the shrinking broadcasting sector and are growing their broadband internet services. This shift towards broadband has been even more expedited with the emergence of several new competitors in the broadband internet services marketplace. According to [5], by 2019-2020, 11 applications had been submitted to the Federal Communications Commission (FCC) from new companies requesting to launch thousands of high-throughput satellites into low Earth orbit (LEO) to form new “mega-constellations”. Existing constellation operators such as ViaSat and SES are also launching more satellites to boost their constellation performance and appeal to potential users.

With the announcement of several private companies all requesting permission from the FCC to expand or begin construction of P-LEO constellations to provide internet services, the competition for space-based services has never had more funding or scrutiny from the private sector than it does today. As each new constellation operator attempts to become a new internet service provider (ISP) and gain market share, they not only contend with other constellation operators, but with terrestrial ISPs as well. This hyper-competitive environment poses business, strategy, and national security risks, so investigation into simulating the feasibility of business strategies becomes critical to determining sustainable operations.

In this work, we detail a new approach of modeling multiple adversarial agents vying for economic gain through gamification. Section 2 covers the general game mechanics. The salient gamification elements needed to model complex interactions are discussed in Section 3. Section 4 discusses the software architecture needed to actualize the game, and Section 5 provides a discussion regarding our modeling rigor, playability, the current limitations of the game, and future expansions.

2. Gamification

This framework aims to use zero or first-order models to inform business strategies involving competitive satellite internet markets. However, such entangled techno-logistic-economic dynamics are complex to model and predict. This work utilizes a mixture of gamification and modeling to make

informed hypotheses regarding the use of partial information games and dominance strategies. The salient challenges associated with the complexity of implementing this framework are summarized in Figure 1 below. Such strategies may inform optimal strategies with respect to constellation congestion, resource scheduling and allocation, different technology development options, and demand management in the future.

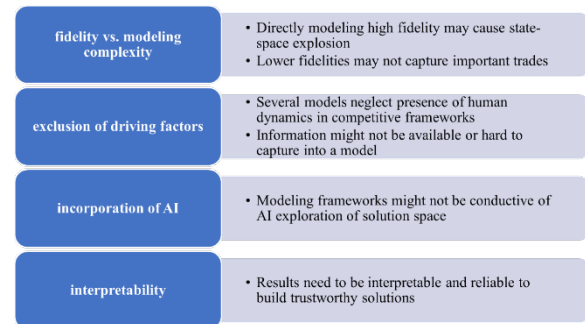


Figure 1: Summary of selected challenges for modelling space logistics and business dynamics

Gamification is a possible approach to develop a framework that may aid understanding complex space logistics and business models for P-LEO assets. Gamification has been shown to be a viable method of simulating complex research problems with no direct or computable solution. Reference [6] reports that there have been over 800 studies into gamification research, advancements, and limitations. Such studies cover domains such as education [7], health [8], and social behavior [9]. While sparse research has been conducted in the fields of business, consumer behavior, innovation, and communication, to the best of our knowledge, no work has been conducted towards the gamification of constellation management and space resource allocation. Gamification particularly benefits our work of creating multi-agent simulations which have very high dimensional or continuous action spaces that require multiple player interactions or critical decision making. Additional benefits of gamification include:

- Games have been the preferred testbed for developing AI programs
- Gamification has been proven to increase user engagement and intrinsic motivation
- Gamification allows testing risk-taking strategies and developing efficient decision-

making processes when facing uncertainties or asymmetric information

- Gamification will allow devising strategies to mitigate externalities and black swan events [10]

Within a gamified framework, elements of interest for space logistics and business dynamics for P-LEO constellations may be mapped to game components via zero-order and first-order modeling. For example, business dynamics may be mapped to game mechanics, strategies driven by confrontation and/or cooperation may be mapped to multi-player interactions, strategic technology investment may be mapped to a game technology tree, and satellite coverage figures of merit may be mapped to player revenues via internet quality and customer satisfaction.

3. The Satellite Tycoon Game

We have developed the prototype of a multi-player game framework, dubbed Satellite Tycoon or Sat-Tycoon in short, to model economic competition within the P-LEO constellation business. The development of a simulation framework, in the form of a business game, may enable us to gain insight into the embryonic stage of the P-LEO constellation industry.

Within Sat-Tycoon, human or AI players may compete to sell high-speed internet access to virtual end-users. Each player controls a P-LEO constellation. Given a fixed start budget, a player’s task is to build and manage their satellite constellation. The success of a player’s strategic decisions can be measured by several metrics, such as the acquisition or loss of end-users, or profit. The game mechanics are that of a resource management game. The game advances through cycles as illustrated by Figure 2. Human players manage resource allocation and develop their constellation through control panels on the game interface, see Figure 3.

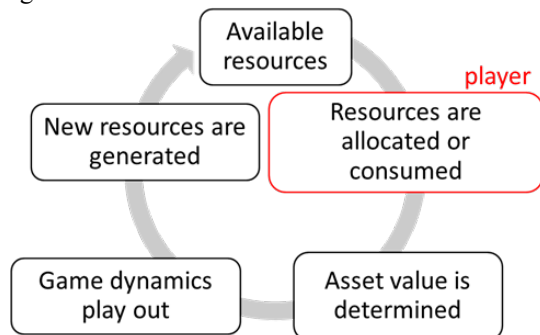


Figure 2: Sat-Tycoon game mechanics



Figure 3: Sample graphical user interface for Sat-Tycoon

Each player’s goal is based on progressing their economic metrics (profit), rendered by cost and reward models simulating the incurred business costs and obtained revenue from satellite internet customers. Recurring and non-recurring business costs such as capital expenditures (CapEx), purchase orders (PO), and Operating Expenditures (OpEx) were modeled. Revenues were modeled based on the players’ chosen internet price and customer base. We model different types of collective customer behavior by framing the customers’ choice as a multi-attribute decision making problem. Coverage and data rate are among the attributes for the customers’ allocation. Approximate coverage and data rate are computed via a constellation engine which models simplified orbit dynamics. The constellation engine is based on semi-analytical formulas that are derived from classical orbital mechanics principles.

For human players, the game is realized through a graphical user interface (GUI) employing a React* client. This client introduces two important visualization tools to aid the players: an interactive map showing geospatial data (e.g., population distributions, constellation coverage, customer allocations, and ground station data rates), and a 2D “Space Deck” where players can visualize orbit shells (altitudes at which satellites operate) and ground tracks associated with specific right ascension and inclination values. The game is designed using a server-authoritative software architecture (basically isolating the GUI seen by human players and the backend game hosted on a dedicated server). This architecture allows potential users to create their own custom GUIs for the game. Additionally, AI agents can leverage the server-authoritative architecture to play the game and train only using the server data. Finally, Figure 4 breaks Sat-Tycoon down in key components and models.

* <https://reactjs.org/>
IWSCFF-22

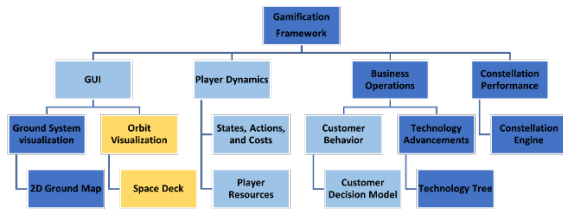


Figure 4: Sat-Tycoon component structure

4. Key Game Elements

The following section describes key elements of the Sat-Tycoon game framework. Key elements include: the constellation engine, the player model, the revenue model, the customer model, and the game progression.

4.1 Constellation Engine

To determine end-user accessibility to the internet service provided by each constellation, basic coverage figures of merit are computed for each constellation. To determine end-user accessibility and quality of the internet service, Sat-Tycoon utilizes a constellation engine that maps constellation geometry and attributes to relevant figures of merit, such as data throughput and coverage gaps. Simplifying assumptions are introduced in the construction of the constellation engine to trade fidelity of the orbit motion with computational speed (which is important for game playability). For a given orbit and ground station, the constellation engine first estimates along-track coverage. Then, the constellation estimates coverage over a full right ascension range at the ground station latitude. Note that coverage over right ascension values may be mapped to coverage over longitudes as a function of the simulation epoch.

4.2 Player Model

Players are modeled as individual agents with a set of actions with continuous form inputs and indirect interactions. Players must focus on becoming a competitive ISP and compete with one another *economically*. The basic actions a player may take are shown in Figure 5.



Figure 5: Base actions available to all players

Players may build a ground station in any valid terrestrial location by clicking directly on the

terrestrial map or by inputting a specific set of coordinates, contingent upon them having sufficient funding. Each ground station has attributes of access area and operational personnel. The size of the access area impacts the number of nearby customers that have access to the service. The number of operational personnel at each ground station directly impacts its recurring cost in the form of salaries.

When ordering satellites, players may choose any positive integer value and that number of satellites will be manufactured for the player, contingent upon them having sufficient funding. Each satellite has basic attributes of mass, power, and lifespan. The mass of a satellite determines how many of that type may be launched using a particular launch vehicle. The power of a satellite is used to compute the average data rate capabilities of a player’s space system. The lifespan is a future metric used to decommission orbit planes once the lifespan has been depleted.

To launch satellites into a valid orbit plane, players must: purchase a launch vehicle, select the number of satellites to launch, and identify the orbit parameters using an inclination and a right ascension of ascending node (RAAN). Once successfully in orbit, the constellation engine computes new constellation parameters for each ground station the player owns. These parameters include coverage and data rate. Once a player has built a valid constellation (of any size), they are able to sell a monthly subscription for their satellite internet services all over the globe. This set price directly impacts not only the number of customers, but also the quality and type of customers acquired by the player.

4.3 Cost and Revenue Model

Each player has, at most, three cost functions at any given point during gameplay. These cost functions are: non-recurring, recurring action-based, and recurring time-based costs. Components of each of these functions are adapted from [11]. The player also has a revenue function which is operated per quarter (discussed below). The interaction between the cost and reward functions is summarized in Figure 6.

A player’s non-recurring costs are defined within this game to be those costs that are only incurred once during the entire game by a user. Such costs may be thought of as capital expenditures (CapEx) of a company. These expenditures typically include: design, development, manufacture, and testing of the qualification model. For initial development, we will restrict ourselves to two simple non-recurring costs: satellite research costs and ground station (gateway) research costs.

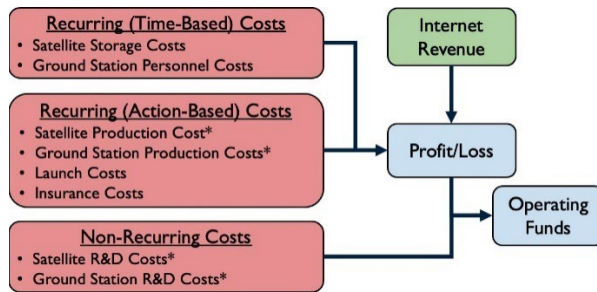


Figure 6: Financial model of players

Recurring costs can be broken down into action-based and time-based. Action-based recurring costs are costs that are computed when the player provides input to the game. Such inputs include: buying additional satellites, buying additional ground stations, and launching satellites. Such costs aren't exactly non-recurring, but their recurrence isn't time-based so they must be modeled differently. Both satellites and ground stations are modeled with dynamic pricing using a cost function that leverages the learning rate during production. Launch costs for Sat-Tycoon are a fixed quantity; further expansion and game dynamics can be achieved by offering tiers of launch vehicles. For instance, players could opt to pay a non-recurring Launch Vehicle charge to upgrade to a better launch vehicle system. Such a non-recurring cost would need to be computed based on the development costs of current launch vehicles (which may prove to be difficult). An additional layer of complexity that may be added in the future could be the variations in orbit altitude. A higher altitude would require more change in velocity and more fuel, thus resulting in a higher launch cost. Lastly, insurance costs must also be computed for each launch. Since these costs are not completely non-recurrent, we cannot classify them as strictly operating expenditures or capital expenditures. These actions are a mixture of both.

The last type of costs to the player are time-based recurring costs. These types of costs can be thought of as pure operating expenditure (OpEx), since they are assessed during every single epoch. Note that we will give costs over a certain sized epoch, but these costs can be divided or multiplied as needed to assess cost over varying epoch sizes. The main time-based recurring cost of a mega-constellation system is operating costs. Such costs can be functions of time, ground station size, telecom system complexity, and personnel required to operate the ground station.

Finally, we assume that the player has assigned a price for their internet service in each area that they operate a ground station. We assume that a Customer Model allocates a set of customers to this player based on the function of the Customer Model. Given these assumptions, computation of a player's revenue may be

computed and assigned each month using the following simple equation $R_{M,N} = I_{M,N} \times N_{Cust}$, where $R_{M,N}$ is the revenue generated per month at grid cell corresponding to M, N ; $I_{M,N}$ is the internet price per month assigned to the grid cell corresponding to M, N (or a specific ground station); and N_{Cust} is the number of customers located within the grid cell corresponding to M, N . Specifics of customer acquisition will be discussed more in the next section.

4.4 Customer Model

The quarterly revenue a player generates is approximated as a function of their chosen internet price and customer base. Each player's customer base can be represented as the summation of all the customers allocated to them by the environment based on their constellation performance and salient business strategies. Such a formulation in which simulated customers choose internet service providers (players in our case) is known as a sealed-bid, reverse-auction process and is shown in Figure 7. Our method simulates customers using a Population Distribution and Growth Model that discretizes the global population by projected squares. In each square, a percentage of the population requests internet access, and this subset of the population is further broken down into four different batches of customer types: loyal customers, price-conscious customers, power user customers, and nominal customers. Each customer type is modeled to have different preferences (modeled using a vector of weighted preferences) when selecting an internet service provider. Each customer type's selection of an internet service provider is formulated as a separate bidding evaluation problem that is then modeled using a Customer Behavior Model. Once a player is selected by the Customer Behavior Model, all customers of that particular type, in that particular projected square, are allocated to the chosen player. Once all the projected squares have had their customers allocated to an internet service provider, each player computes their revenue based on their total global customer base and their set internet price.

The customer behavior model is framed as a multi-attribute decision making (MADM) problem in which batches of customers, in a particular projected square, decide upon an internet service provider given a set of weighted preferences. This method of customer decision-making in the context of a reverse-bidding formulation is adapted from Section 3.2 of [13]. The general Technique for Order of Preference by Similarity to Ideal-Solution (TOPSIS) method, proposed by [14] and outlined by [15], is used to solve the MADM problem for each customer type, in each grid cell.

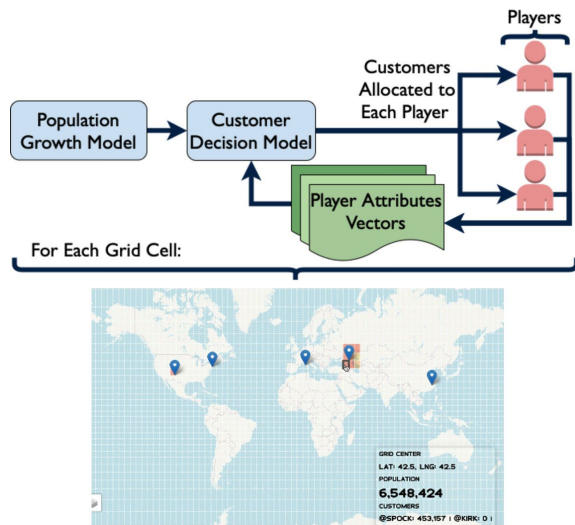


Figure 7: Customer Model Flowchart

4.5 Game Progression

Sat-Tycoon is formulated as a multi-player, strategy-simulation game in which players act as constellation operators vying for virtual customers and subsequent revenues. Players are allocated funding at the beginning of the game and can set up a limited constellation network before the game clock begins. This is known as game state initialization. Once all players have confirmed that they are ready to begin the game, they are dropped into the main game dashboard that has all the controls of the game. The game then progresses over a fixed period with players receiving quarterly revenue.

4.5.1 Game State Initialization

Slow-paced competition dynamics during the build-up phase of a P-LEO constellation may challenge human player engagement and must be very finely tuned with reward, costs, and game progression. To alleviate some of this fine tuning and accelerate the constellation build-up phase closer to a state of competition for scarce resources, Sat-Tycoon starts off with a zero-turn phase. In the zero-turn phase, players complete a full game cycle that defines their initial constellation. The zero-turn phase currently comprises a sequence of four actions: 1) build the first ground station; 2) order the first batch of satellites; 3) launch satellites into a specific orbit plane; 4) set internet price. The zero-turn phase may be customized based on a player's level of experience, thus effectively functioning as a short tutorial for new users.

Once the game state is initialized and the game has begun, game time progresses on a simulated clock. This clock moves at an accelerated rate that can be customized to provide games of different length and pace. The default rate at which the game progresses is

two days per every second of real time. A standard ten-year game of Sat-Tycoon at this rate will last thirty minutes. Each quarter the global population, player customers, and player revenue will be updated and sent to the clients.

4.5.2 Technology Progression

As time progresses in the game, players may research and upgrade their future technology purchases by executing a non-recurring cost to develop these technologies. Better satellites, ground stations, and launch vehicles may all be unlocked if their respective technology level is researched.

Throughout the progression of each technology class, key attributes associated with constellation performance or recurring operational costs are improved. As players unlock more and more sophisticated satellites, transmittable power increases, satellite mass decreases, and cost decreases. Ground stations act similarly; however, the recurring operational costs of a ground station decreases due to automation of tasks (and therefore fewer employees to pay). Launch vehicles exhibit a similar pattern but are specifically modeled with current commercial launch vehicle performances [12].

Technology development not only unlocks a new aspect of game complexity, but may also allow for investigating how competitive advantages are afforded to those that invest strategically in selected technologies.

5. Software Implementation

Sat-Tycoon is developed to be capable of multi-modal, multi-player interactions. Utilizing a custom-built WebSocket API, the server can host the game for any number of custom human or AI clients. Clients can be built in any language and join the same lobby to play the game using Sat-Tycoon's custom-built API. The Sat-Tycoon team has built and provides a web browser client built in Javascript and React so that players may play the game without developing their own client or AI agent. The browser client boasts an intuitive interface that is both fun and professional, providing an ideal game feel.

5.1 Server Authoritative Architecture.

Sat-Tycoon is developed as a software architecture that is capable of multi-modal, multi-player interactions. The server is constructed utilizing a Server Authoritative Architecture (SAA), which is a network communication architecture for games in which clients communicate with a server that maintains a master data set [16]. The clients have no ability to directly act upon this data set, and instead make requests to the server which then acts upon the

data set. Clients maintain local data sets that they then keep as accurate to the master as possible. SAA also removes the need to synchronize clients. All clients pull from the same central data set. The only synchronization a client needs to be concerned with at any time is between its local data set and the master data set. All game logic is performed server-side, and only the results are communicated to clients. In order to cheat, a malicious agent would need to access the server directly and act upon the master data set. This architecture results in a much more robust environment, removing client-based gameplay errors entirely.

Occasionally a client may experience desynchronization between local data and the master data set. This can be caused by several things such as network lag, packet loss, or disconnection. When this occurs, the client will need to update its local data by requesting accurate data from the server. This can be partially remedied by implementing client-side prediction, or having clients calculate the results of their actions locally before sending a request to the server. When the server responds, if the client's results are incorrect, they will be updated. This is called server reconciliation. This may allow network delays and inconsistencies to go unnoticed by users, but requires the client to be more complex. In summary, a SAA offers the following benefits: 1) clients cannot modify the game state; 2) there are no synchronization requirements; and 3) it prevents intentional or unintentional cheating. Figure 8 schematically depicts the SAA that is implemented in Sat-Tycoon.

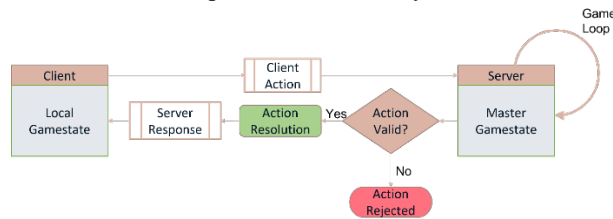


Figure 8: Schematics of a Server Authoritative Architecture

Additionally, the Sat-Tycoon backend server (i.e., the framework component that handles player interactions and game mechanics) is developed using an object-oriented programming approach (OOP) to support expandability, encapsulation, and AI interfacing.

5.2 Python Server.

Sat-Tycoon's server is written in Python 3.9 and utilizes the WebSockets library for network communications with the player clients. Therefore, clients can be written in any language supporting WebSockets. The WebSocket protocol is used to asynchronously handle player connections and

communicate actions between the client and server. Actions are formatted as JSON objects containing a plain English "action" string, and data as a "payload." These actions are interpreted by the server and used to perform game logic as requested by a client. New clients can be created without modification of the game, as the backend server hosts and handles the game loop. Sat-Tycoon is usable by both human and AI clients because through the developed API the server does not differentiate requests based on the type of requesting client. The server will store a player object and accept actions regardless of what type of client has initiated the connection and sent the actions. Games on the server are organized into player lobbies that each host independent game sessions. This allows multiple games of Sat-Tycoon to be played simultaneously. Player clients within these lobbies can communicate with each other using specific WebSocket actions while playing the game. Figure 9 shows an example of possible player connections, emphasizing that the current architecture may enable transforming Sat-Tycoon, with further development, into a Gym environment for AI agent developers.

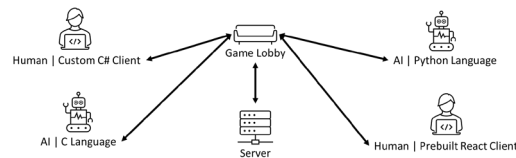


Figure 9: Example of possible player connections that are enabled by a WebSocket API

5.3 Graphical User Interface and React Client.

The Sat-Tycoon React Client offers a graphical user interface (GUI) for human players who may not want to build their own client or develop an AI agent. This default client for human players is built in React with Javascript, provides a GUI, and handles all WebSocket communication with the server. In the following, GUI elements that are critical to the gamification process of logistics and business dynamics for P-LEO constellations are described.

5.3.1 Ground Map

The gamification of P-LEO constellations requires user interaction with geospatial information. For example, players may need to evaluate coverage, data rate, and customer population distribution across the globe. Players may also need to take actions that possess geographical attributes, such as building a ground station or setting regional internet price. The solution for user interaction with geospatial information should accommodate players of different

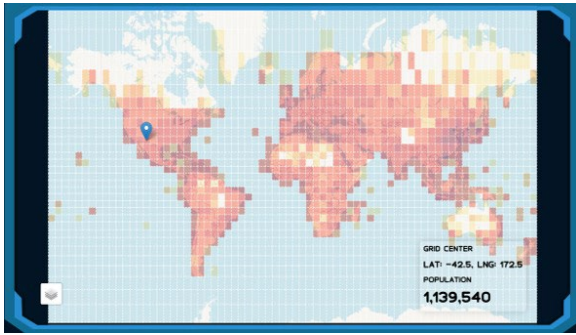


Figure 10: Sample color-coded grid map displaying global population distribution that is part of the Sat-Tycoon graphical user interface

skill levels, should return meaningful feedback for player ground system actions, and should be realizable with limited memory, graphics, and computational speed. As a solution, the Sat-Tycoon React client allocates the most relevant information and action that have geospatial connotation to an interactive, two-dimensional ground map, as shown in Figure 10. The React client utilizes the react-leaflet library to provide an interactive map of the world. The map displays key information to the player, such as global population, player customers, player data rate, and player satellite coverage. This data is sent to the client in real-time from the server using WebSockets and GeoJSON. Current geospatial data that can be visualized on the interactive ground map includes:

- Customers are displayed as local heatmaps around player ground stations. The darker the color of a tile, the more customers a player has in that location.
- Coverage is displayed as heatmap bands across the map. The darker the color of a tile, the more satellites are passing over the area on average.
- Data rate is displayed as a local heatmap on the tiles surrounding player ground-stations. The darker the color of the highlighted tiles is, the higher the data rate is in that area.
- The population is displayed on the map as a population density heatmap. Tiles that are darker red are more heavily populated, while tiles that are lighter are less populated.

5.3.2 Space Deck

Like the visualization and interaction with geospatial information, the gamification of P-LEO constellation logistics requires user interaction with orbit information. For example, players need to select combinations of orbit planes and altitude to build constellation shells and qualitatively evaluate satellite ground tracks. The solution for user interaction with orbit information should accommodate players of different skill levels, should return meaningful feedback for players to gauge their launches, and should be realizable with limited memory, graphics, and computational speed. In response to these requirements, the Space Deck is a custom-built interface for launching and visualizing satellite constellations. The Space Deck interface (see Figure 11) is presented as a set of grids with two axes: Inclination and RAAN. Each grid represents an altitude, and each grid cell represents a location at the altitude at which to launch satellites. After clicking a grid cell, ground tracks are visualized in real-time on an animated map to show the player the resulting satellite coverage.

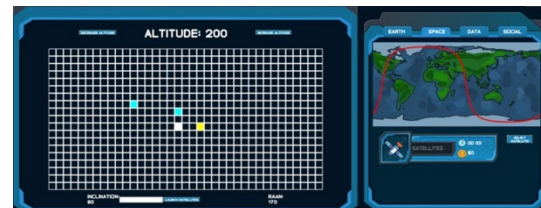


Figure 11: Sample Space Deck Implementation

5.3.3 Tech Tree

To gamify the business dynamics of P-LEO constellations, an approachable interface for the research and purchase of constellation technology was required. The technology tree was developed to present players with lists of the different available constellation technologies as well as important information about their capabilities. Each technology type is broken into categories, and then each item in those categories provides important information on the capabilities of a particular technology. The technology tree also provides players with intuitive visual comparisons between their current technology and the technology presented in the interface. Properties that would be upgraded by implementing the new research are presented in green, while properties that would be downgraded are presented in red (see Figure 12). A system to allow players to buy improved technology increases gamification by providing depth and a sense of progression.



Figure 12: Sample Technology Tree with Comparisons

5.3.4 Social Panel

To provide a richer multiplayer experience and improve gamification, a social panel was developed where players can communicate and interact with each other during gameplay (see Figure 13). The ability to discuss strategies, events, and other aspects of the game may lead to a more enriching game experience for all players. The social panel consists of a chat window, and two player lists. The first list displays all players online on the Sat-Tycoon server, and the second list shows all players in the current lobby. The server does not differentiate between human and AI clients, resulting in the social features being available to AI agents. An AI agent will be able to see online players and communicate with them via the chat system. Human players will also be able to see and communicate with AI agents.

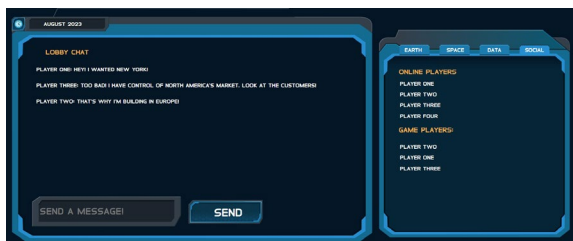


Figure 13: Example of multiple players discussing their Sat-Tycoon game session

5.3.5 Analytics Panel

The gamification of business dynamics necessitates the development of a panel that provides relevant constellation performance and historical business data. The analytics panel is currently being developed to provide this business information to players. The panel will include important regional data for player ground stations such as current customers, available customers, average customer count, average revenue, and average gross domestic profit. Providing a broad view of performance data for their constellation provides players with the necessary information to

develop strategies with depth and purpose during the game. The current analytics categories are: Earth, Space, and Finance. Analytic data for Earth includes customer, ground station, data rate and population data. Analytic data for space includes orbital, lifespan, and coverage data. Financial data includes revenue, gross domestic product (GDP), and expenditures.

6. Discussion

Using first order astrodynamics modeling techniques, principles of gamification, and state-of-the-art software practices, Sat-Tycoon demonstrates a gamified framework that allows multiple players to develop constellations and compete against one another for virtual customers. As shown in Figure 14, a working Sat-Tycoon prototype is hosted on a university server in which multiple players can join a lobby and play through a full game together. Figure 15 shows an example of multiple players joining a Sat-Tycoon game session. This framework has led to the development of multiple “plug-n-play” modules that can be updated with future expandable models for added functionality and more complex strategy simulation.

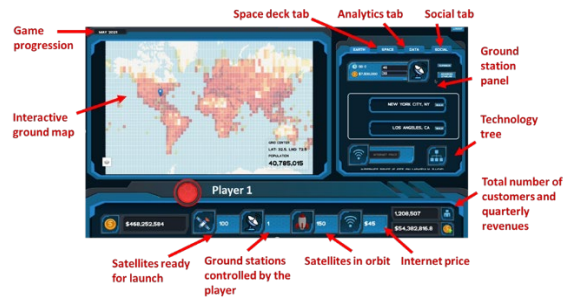


Figure 14: Prototype of Sat-Tycoon hosted on Auburn University server

The realization of the Sat-Tycoon prototype has also led to the identification of opportunities and challenges for the analysis of space logistics and business dynamics that further development may address. Some possible future expansions involve incorporation of individualistic goals and missions for each player (company), more rigorous modeling of satellite capacity and technology developments, and artificial agents that employ rudimentary and advanced game theory principles.

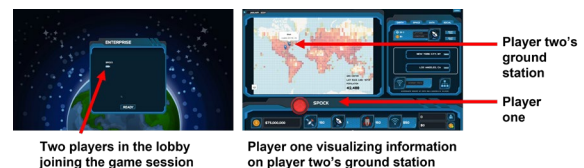


Figure 15: Example of multiple players joining a Sat-Tycoon game session



6.1 Game Playability and Refinements

Different playing and testing sessions within the development team have highlighted the challenges of presenting orbital mechanics information, coverage, and coverage figures of merit to non-expert users or users that are less familiar with orbital motion and orbital mechanics. Understanding of such elements is necessary to correlate design choices to their impact on constellation performance, and, thus, on revenues. Future research may investigate visual-analytics techniques in the context of P-LEO constellation and orbital mechanics.

6.1.1 Future Playability Study

As mentioned above, the current version of Sat-Tycoon should be Beta tested to identify gaming strategies, game pace, and overall playability. To do this, we propose to undertake a playability study to help inform development and modeling regarding player experience and future expansions in modeling. This study will coincide with the Beta Testing phase of the project in which invited participants will play the game a certain number of times or for a certain length of time. Once a certain number of hours or game sessions are logged, feedback will be collected, and bugs or gameplay issues will be resolved.

6.2 Real-World Strategies & Game Theory

An overarching goal of this framework is to be able to use zero or first-order models to inform business strategies involved in a competitive market landscape. Using game theory and the current modeling approach, we may make informed hypotheses regarding the use of partial information games and dominant strategies. Future modeling is aimed to inform strategies involving constellation congestion, resource scheduling and allocation, different technology development options, and demand management.

6.2.1 Congestion and Resource Scheduling

As more companies, agencies, and nations compete for space access and thousands of satellites are launched into orbit, bureaucratic and physical congestion in space increases. From such a congestion, strategies emerge that can slow down competition or render certain orbit shells inoperable by any competitors.

For example, novel strategies that may emerge but need to be verified with further modeling include: purposeful congestion of FCC applications to access certain bandwidths and orbit shells, and purposeful physical congestion of certain orbit shells to force future operators to navigate around existing constellations. The former of these strategies indicates a bottleneck in processing requests to launch mega-constellations and yields a first mover advantage. Any

slowdown in process costs opponents potential revenues and incurred costs and is therefore a potential strategy. However, additional modeling of application filing and queueing control will be needed to simulate administrative congestion scenarios. The latter example strategy potentially leads to greater costs to opponents: to achieve a comparable constellation performance when the desired orbit shell is inaccessible, opponents will have to launch at different altitudes, one that may correspond to suboptimal design solutions. This strategy also benefits from first mover advantage and incurs residual costs on opponents as well.

Like strategies involving congestion, resource scheduling involves timing certain activities to either benefit a player or cost to opponents. In the case of scheduling launches, realistically, there are a finite number of launches available for constellation operators to purchase. Therefore, purchasing all or most of the available launches before a competitor can cost the competitor revenues as well as give the player economic advantage. Additionally, scheduling manufacturing of satellites has a similar effect, but must be mitigated due to the costs associated with the storage of unlaunched satellites.

Administrative congestion and resource scheduling may be modeled within the Sat-Tycoon framework by queueing, delaying, or cancelling certain actions' execution, in particular satellite launches. Physical congestion of orbit shells may be modelled within the Sat-Tycoon framework by associating a probability of satellite collision with each grid cell of the Space Deck. Such a probability of collision may be a function of the number of active and passive objects that reside in the orbital space described by each grid cell of the Space Deck.

6.2.2 Applicable Domains for Further Exploration

Additional domains the Sat-Tycoon framework may explore with further development include technology development forecasting and demand management in shifting market landscapes. Satellite technology progression may present more development paths than the number of opportunities that a decision maker can commit resources and time to pursue. The Sat-Tycoon framework potentially offers a virtual experimental laboratory to simulate and test different technology progressions in a competitive environment.

Technology progression may be coupled to adversary technologies, driving which technologies a player chooses to develop to remain relevant as a company. Another possible aspect of exploration for technology development is the development of technologies "in-house". In-house development lends itself to several new strategies due to the increased CapEx required for technology development and



progression, hidden technologies from adversaries (due to proprietary development by each player), and a potentially (but not guaranteed) lower OpEx in long-term operations of the constellation.

Technology progression may be modeled within Sat-Tycoon by introducing technology tokens and modifiers that upgrade the attributes of the satellite (e.g., mass, power, reliability, etc.). The introduction of technology tokens and satellite attributes allows the circumvention of the problem of modelling technologies that do not exist in order to simulate technology progression. With technology tokens and attributes, we may model the effects of unknown technology on the game dynamics (that is, we are agnostic to the cause of the attributes' improvement).

Finally, considering the recent uptick in residential satellite internet demand, traditional use cases for satellite internet may not be adequate for players to sustain profitable businesses. Such shifting market landscapes (along with demographic and population changes) require players to consider demand for their internet services in different geographic areas and use cases. In these new use cases, continuous, expensive coverage over a certain area might in fact be detrimental to a player's overall profits. Further modeling of customers for specific use cases and conversion of stateless customers to customers with historical information will yield additional, complex strategies.

6.3 Interfacing with AI

The Sat-Tycoon API provides a starting point to interface the game with AI players. The server does not have any special requirements for AI agents, and AI clients communicate with the server using the same WebSocket API that player clients use. For example, an AI gym environment for reinforcement learning agents could be developed to act as a client that handles all the WebSocket calls for an AI developer. Using this client, a developer would need to provide the AI code for their agent. The Sat-Tycoon server is currently capable of supporting such a client; however, the game has only been tested with human players and modifications to the server or WebSocket API may be necessary as issues with AI players are discovered. For example, the pace of the game may need to be changed to train certain AI agents. A special game mode could be developed on the server that uses a different clock speed and game loop to facilitate better AI training. Another possibility is that upon training an AI agent, it is discovered that the information returned from the server could be better tailored for training AI.

7. Final Remarks

This work details a new approach of modeling multiple adversarial agents vying for economic gain through a framework involving gamification and first-

order modeling. Custom game mechanics, an orbital dynamics framework, game elements needed to model complex interaction, and a software architecture needed to actualize the game have all been developed in conjunction to facilitate this framework. The goals of the project and the internal requirements conducive to a functional game both drove modeling, design, and implementation to the current version of the game. Through this project, we show that gamification can be used to develop adversarial P-LEO constellations that offer satellite internet access to retail customers.

Currently the provided React client and server are internally hosted on a university server and are not accessible to players or developers outside of Auburn University. An internet facing server may be realized in the future.

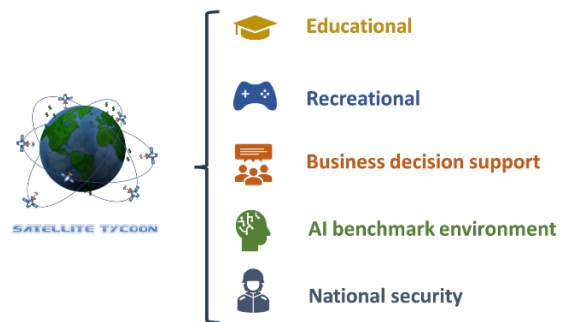


Figure 16: Five possible, longer-term pathways for Sat-Tycoon development

With further development, a gamified framework to simulate the interactions between constellation operators and virtual customers may aid in determining the feasibility of possible business strategies, but may also inform use cases for national security, education endeavors, and additional research areas, such as technology portfolio development. Figure 16 visualizes the possible, longer-term pathways to continue Sat-Tycoon development. The framework demonstrated with the Sat-Tycoon prototype may also serve as a baseline for frameworks to simulate P-LEO asset servicing. A gamified framework for P-LEO asset servicing may include logistics dynamics based on responsive launching, refueling, refurbishing, repairing and debris removal, both with a competitive or collaborative gameplay.

8. Acknowledgement

This material is based on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA9453-20-1-0008. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.



9. Disclaimer

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of Air Force Research Laboratory (AFRL) and or the U.S. Government.

8. References

- [1] Peter Harrell, Elizabeth Rosenberg, and Edoardo Saravalle. *China's use of coercive economic measures*. Center for a New American Security Washington, DC, 2018.
- [2] Sydney Finkelstein and Shade H Sanford. Learning from corporate mistakes. *Organizational Dynamics*, 29(2):138–148, November 2000.
- [3] Jaejoo Lim, Richard Klein, and Jason Thatcher. Good technology, bad management: A case study of the satellite phone industry. *Journal of Information Technology Management*, 16(2):48–55, 2005.
- [4] Edward W. Ashford. Non-Geo systems—where have all the satellites gone? *Acta Astronautica*, 55(3-9):649–657, August 2004.
- [5] Inigo del Portillo, Bruce G. Cameron, and Edward F. Crawley. A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. *Acta Astronautica*, 159:123–135, June 2019.
- [6] Jonna Koivisto and Juho Hamari. The rise of motivational information systems: A review of gamification research. *International Journal of Information Management*, 45:191–210, April 2019.
- [7] Katheryn R. Christy and Jesse Fox. Leaderboards in a virtual classroom: A test of stereotype threat and social comparison explanations for women's math performance. *Computers Education*, 78:66–77, 2014.
- [8] Brooke A. Jones, Gregory J. Madden, and Heidi J. Wengreen. The FIT Game: preliminary evaluation of a gamification approach to increasing fruit and vegetable consumption in school. *Preventive Medicine*, 68:76–79, November 2014.
- [9] Jennifer Thom, David Millen, and Joan DiMicco. Removing gamification from an enterprise SNS. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12*, Seattle, Washington, USA, 2012.
- [10] Taleb, Nassim Nicholas, *The Black Swan: the impact of the highly improbable* (2nd ed.), London: Penguin, 2010
- [11] Kelic Andjelka. *Assessing the technical and financial viability of broadband satellite systems using a cost per T1 minute metric*. MS thesis, Massachusetts Institute of Technology, May 1998.
- [12] Harry Jones. The recent large reduction in space launch cost. 48th International Conference on Environmental Systems, 2018.
- [13] Chi-Bin Cheng. Solving a sealed-bid reverse auction problem by multiple-criterion decision-making methods. *Computers & Mathematics with Applications*, 56(12):3261–3274, 2008.
- [14] Gwo-Hshiung Tzeng and Jih-Jeng Huang. *Multiple attribute decision making: methods and applications*. CRC Press, Boca Raton, FL, 2012.
- [15] Taho Yang and Chih-Ching Hung. Multiple-attribute decision making methods for plant layout design problem. *Robotics and Computer-Integrated Manufacturing*, 23(1):126–137, 2007.
- [16] Gabriel Gambetta. Fast-paced multiplayer (part i): Client-server game architecture. client-server game architecture. <https://gabrielgambetta.com/client-server-game-architecture.html>. last accessed: 11.20.21.