AFRL-RV-PS-
TR-2022-0013

AFRL-RV-PS-
TR-2022-0013

MODELING ECONOMIC COMPETITION IN THE BUSINESS OF MEGA-CONSTELLATIONS

Davide Guzzetti and Daniel Tauritz

Auburn University
107 Samford Hall
Auburn, AL 36849

6 Jan 2022

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776

NOTICE AND SIGNATURE PAGE

This report is the result of contracted fundamental research which is exempt from public affairs security and policy review in accordance with AFI 61-201, paragraph 2.3.5.1. This report is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RV-PS-TR-2022-0013 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//

RICHARD "SCOTT" ERWIN
Program Manager

//SIGNED//

SEAN PHILLIPS
Tech Advisor, Space Control Technologies Branch

//SIGNED//

JOHN BEAUCHEMIN
Chief Engineer, Spacecraft Technology Division
Space Vehicles Directorate

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
06-01-2022

2. REPORT TYPE
Final Report

3. DATES COVERED *(From - To)*
06-Jul-2020 06-Jan-2022

4. TITLE AND SUBTITLE
Modeling Economic Competition in the Business of Mega-Constellations

5a. CONTRACT NUMBER
FA9453-20-1-0008

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
C6601F

6. AUTHOR(S)
Davide Guzzetti and Daniel Tauritz

5d. PROJECT NUMBER
8809

5e. TASK NUMBER

5f. WORK UNIT NUMBER
V1Q7

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Auburn University
107 Samford Hall
Auburn, AL 36849

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Space Vehicles Directorate
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117-5776

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RVSW

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RV-PS-TR-2022-0013

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project aims at developing a game framework to model economic competition within the business of mega-constellations. A business game framework will enable to simulate the outcome of strategic decisions and acquire insight on which factors drive the success of a mega-constellation enterprise. 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 final outcome of this project can be used as an educational tool, a business decision support system, a simulation used for national security, an AI test environment, or simply a recreational game.

15. SUBJECT TERMS
Gamification, Proliferated Low Earth Orbit Constellations, Resource Allocation, Adversarial Strategies, Customer Modeling, Constellation Modeling. July 2018 ONR R&D General Terms and Conditions

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
Unclassified	Unclassified	Unclassified	Unlimited	46

19a. NAME OF RESPONSIBLE PERSON
Richard "Scott" Erwin

19b. TELEPHONE NUMBER *(include area code)*

(This Page Intentionally Left Blank)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. SUMMARY

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.

2. 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 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 logistics and economics of P-LEO constellations, and developing tools 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 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.

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 human interactions or critical decision making.

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 tobecome a new internet service provider (ISP) and gain market share, they not only contend with other constellation operators, but with traditional, regional 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 report, we detail a new approach of modeling multiple adversarial agents vying for economic gain through gamification. Section 3.1 covers the general game mechanics. Our orbital dynamics framework is discussed in Section 3.2. The salient gamification elements needed to model complex interactions are discussed in Section 3.3. Section 3.4 discusses the software architecture needed to actualize the game, and Section 4 provides a discussion regarding our modeling rigor, playability, the current limitations of the game, and future expansions.

3. METHODS, ASSUMPTIONS, AND PROCEDURES

This framework aims to use zero or first-order models to inform business strategies involving competitive satellite internet markets. However, such entangled 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.

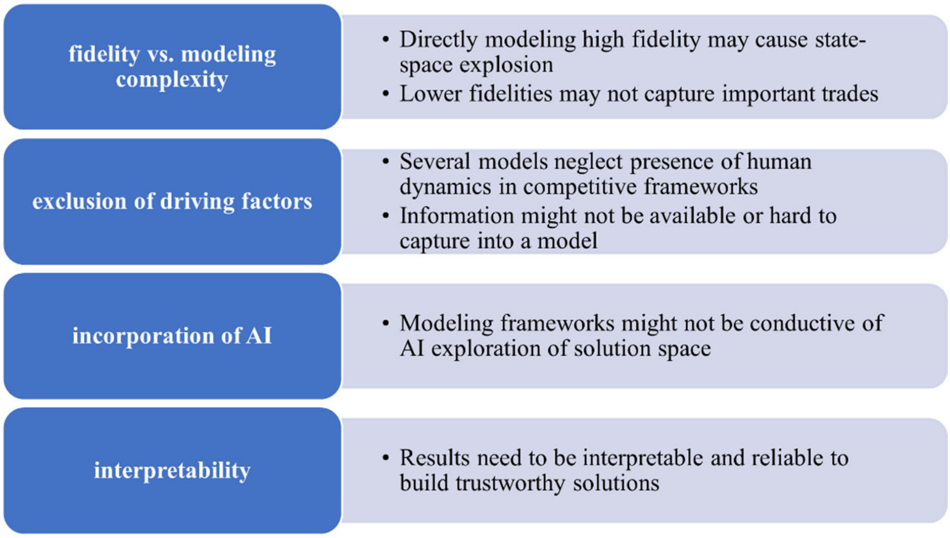


Figure 1. Summary of selected challenges for modeling space logistics and business dynamics

Gamification is a possible approach to develop a framework that may aid understanding complex space logistics and business models for P-LEO assets. Advantages offered by 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

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 maybe mapped to player revenues via internet quality and customer satisfaction.

3.1 The Satellite Tycoon Game

We have developed the prototype of a multi-player game framework, dubbed Sat-Tycoon, 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.

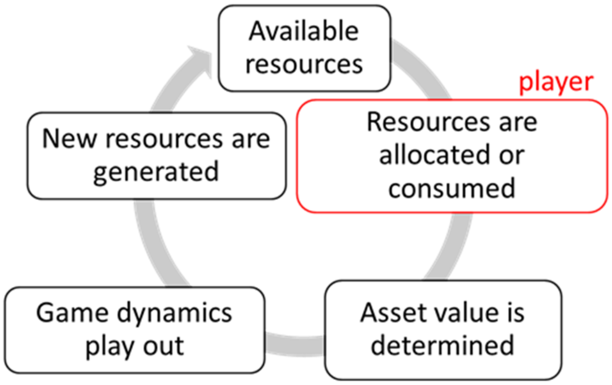


Figure 2. Sat-Tycoon game mechanics

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.

Figure 3. Sample graphical user interface for Sat-Tycoon

The constellation engine has been validated versus numerical simulations and commercial software. For human players, the game is realized through a graphical user interface (GUI) employing a React[1] 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 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.

---

[1] https://reactjs.org/

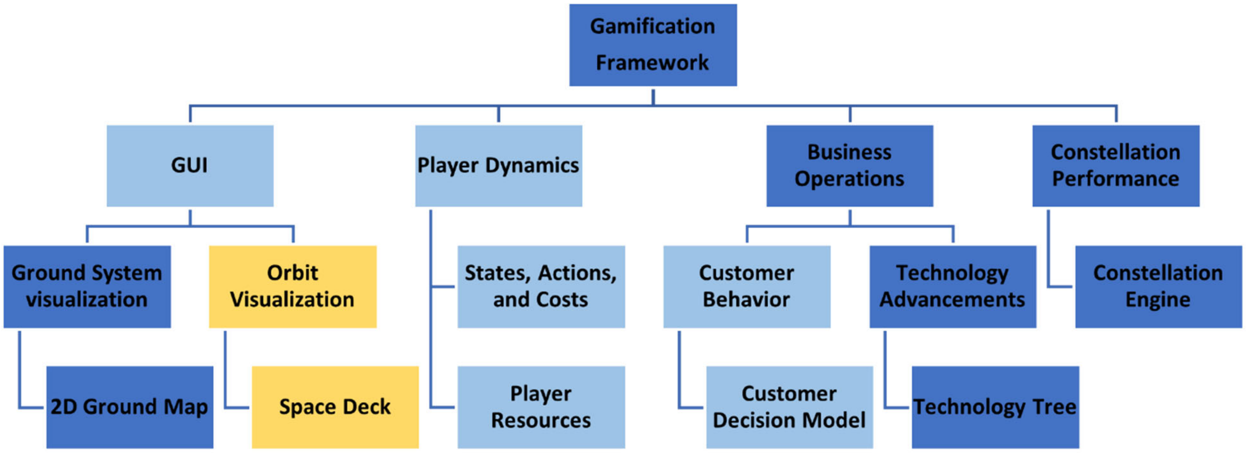


Figure 4. Sat-Tycoon component structure

3.2 Orbital Dynamics Framework

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, as detailed by Figure 5.

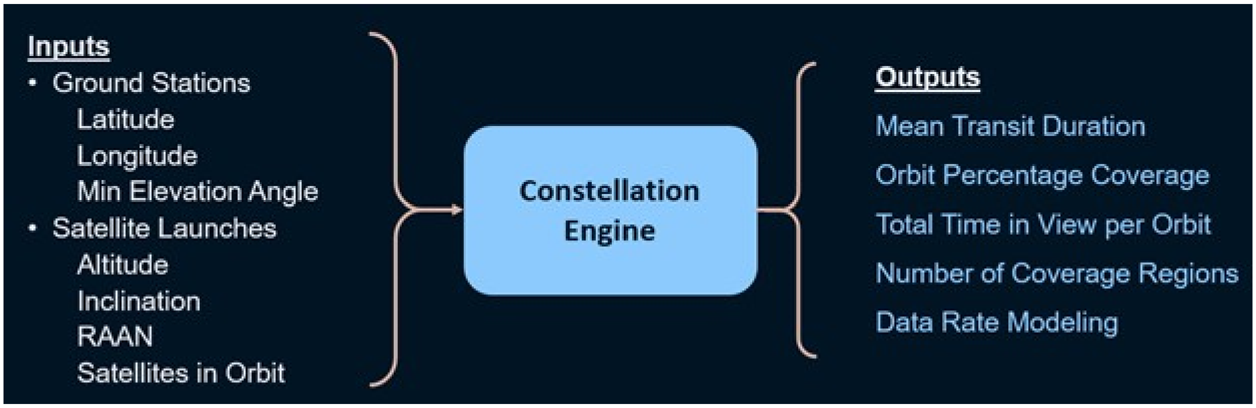


Figure 5. Overview of the constellation engine inputs and outputs

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. The along-track coverage is estimated using the following pseudo-algorithm:

1. Acquire orbit inclination and RAAN
2. Convert RAAN to longitude of the ascending node, include modeling of Earth rotation (360 degrees in 1436 minutes)
3. Compute the instantaneous latitude and longitude of the orbit pole
4. Compute the Earth's angular radius as a function of the orbit altitude, ρ
5. Read minimum elevation angle for the gateway, E_{min}
6. Compute the maximum nadir angle, μ_{min}, as a function of the minimum elevation and Earth'sangular radius
7. Compute the Earth's maximum central angle, λ_{max}, as a function of the maximum nadir angle and minimum elevation
8. Compute the Earth's minimum central angle, λ_{min}, as a function of instantaneous orbit pole angular coordinates and ground station angular coordinates.
9. Compute an average Earth's central angle, $\lambda_{\mathrm{ave}} = (\lambda_{\mathrm{max}} - \lambda_{\mathrm{min}})/2$
10. Compute the orbit period, P
11. Compute fraction of the orbit over which the ground station is in view, assuming non-rotatingEarth (reasonable for LEO), and duration of a single transit

    $$\Delta T = \frac{P}{\pi}\left(\frac{\cos\lambda_{max}}{\cos\lambda_{avg}}\right) \tag{1}$$

12. Estimate the average number of satellites visible along track at a random epoch as $n = \frac{\Delta T}{P} n_{sat}$ where n_{sat} is the total number of satellites along a given orbit.

The along-track algorithm is conceptually visualized in Figure 6.

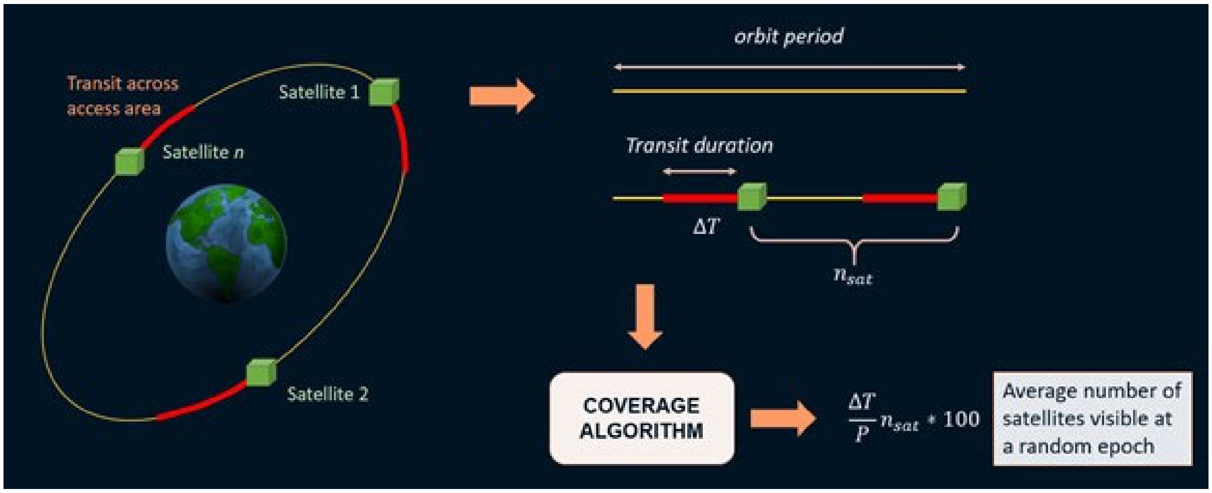


Figure 6. Overview of the along track coverage estimation

Next, the percentage of satellite orbits over one day that will cover a single point of Earth surface at the given latitude, C_ϕ is estimated. Depending on the selected orbit, three coverage scenarios are expected as illustrated in Figure 7: 1) no coverage regions, 2) one coverage region and 3) two coverage regions. For the case with one coverage region, the percentage of satellite orbits is $C_\phi = \frac{\Delta\lambda_1}{\pi} \cdot 100$ where

$$\cos\Delta\lambda_1 = \frac{-\sin\gamma_{max} + \cos i \sin\phi}{\sin i \cos\phi} \tag{2}$$

For the case with two coverage regions, the percentage of satellite orbits is $C_\phi = \frac{\Delta\lambda_1 - \Delta\lambda_1}{\pi} \cdot 100$ where

$$\cos\Delta\lambda_2 = \frac{+\sin\gamma_{max} + \cos i \sin\phi}{\sin i \cos\phi} \tag{3}$$

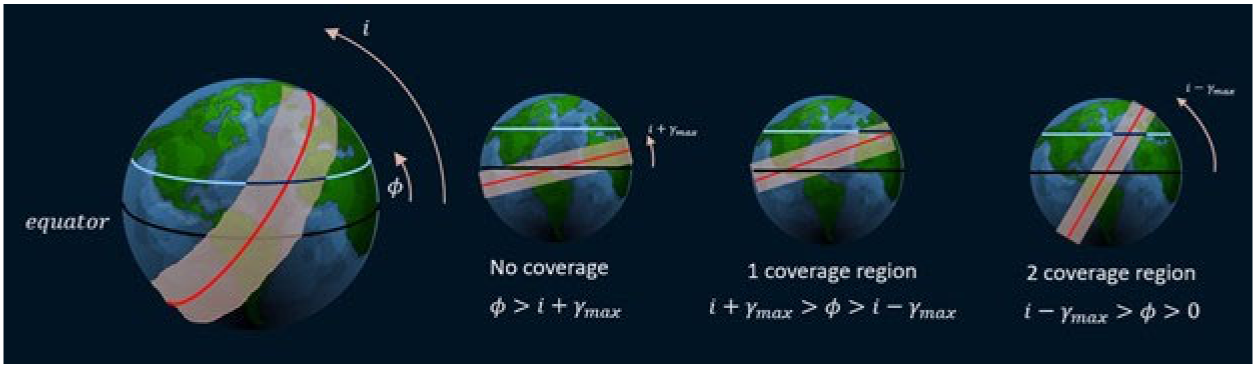


Figure 7. Overview of possible coverage regions for different orbit inclinations and ground station latitudes

Finally, the average number of satellites visible at a random epoch, *n*, is registered in right ascension bands covered by the selected street of coverage at the given latitude (i.e., the across-track coverage). Figure 8 depicts how the average number of satellites visible at a random epoch, *n*, is registered for the case with two coverage regions.

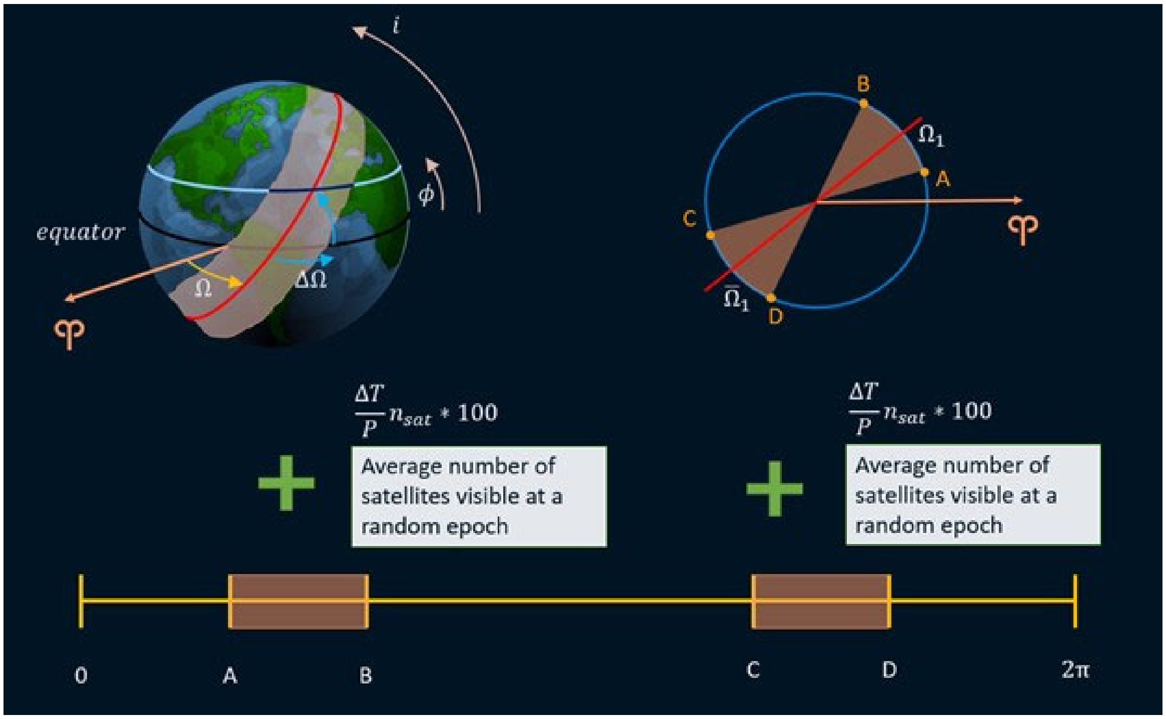


Figure 8. Overview of the across track coverage estimation

For regions on Earth where constellation coverage is present, a mean data rate is estimated by averaging across all satellites servicing the given location. The data rate for an individual satellite is estimated utilizing canonical link budget equations, and accounting for path losses and receiver noise. The equivalent isotropic radiation power, receiver gain, receiver noise, and signal bandwidthare quantities determined by the player’s choice for telecommunication technology. The path losses are a function of the satellite altitude. Atmospheric misalignment and cable losses are not currently accounted for in the computation of the link budget and data rate.

3.3 Salient Gamification Elements

The following section describes key elements needed to ensure accurate and playable gamificationof a reverse-auction game. These include: integrated players, cost and reward models, a novel game design, technology progression models, and a decision model for virtual customers. Such components are modular and may be improved if computational time is available.

3.3.1 Player Modeling.

Players are modeled as individual agents with a set of actions with continuous form inputs and indirect interactions. As in the real world, players are, legally, unable to directly impact their competitors. Instead, they must focus on becoming a competitive ISP and compete with one another *economically*. The basic actions a player may take are shown in Figure 9.

Figure 9. 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 minimum elevation angle and operational personnel.The minimum elevation angle is used to compute coverage, and therefore quality, over each particular ground station. 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.

3.3.2 Cost & Reward Modeling.

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 [10]. 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 10.

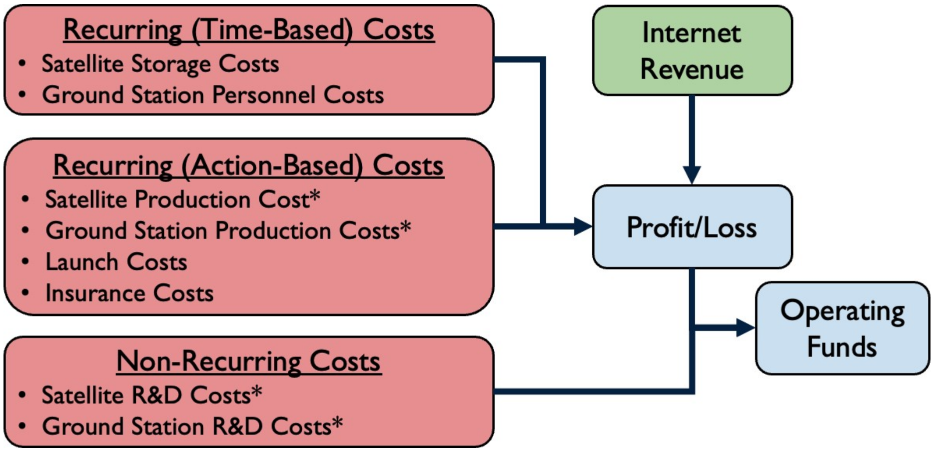


Figure 10. Financial model of players

Estimations of System Costs

System Costs may be summarized as the total instantaneous costs to a player at a specific pointin their gameplay. Moreover, the following approximation equation from [11] is given for this computation:

$$F(T) = [10 + T(6T - 15)]T^3 \tag{4}$$

where F is the fraction of cost (or in the player's case, available funding) consumed in time T and T is the fraction of total time elapsed. This is a very broad estimation and is therefore not readily used. However, it may serve as an indicator for post-game analysis in cases where players spent funds too quickly or slowly.

Non-Recurring Costs

A player's non-recurring costs are defined within this game to be those costs that are only incurred once during the entire game to 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. From literature [12], typical P-LEO satellite non-recurring costs are given as follows:

$$NRC_{SAT} = 7 \times TFU_{LEO} \tag{5}$$

where NRC is the non-recurring cost, TFU is the theoretical first unit cost of the product (a LEO satellite in our case), and 7 is a multiplier on the theoretical first unit cost based on industry experience from commercial satellites [12]. Also, from the same literature, we may take the non-recurring costs associated with researching and constructing an initial ground station:

$$NRC_{GS} = 5 \times TFU_{GS} \tag{6}$$

where these are the corresponding values to non-recurring costs and theoretical first unit costs. Such values will be incurred when the player purchases their first satellite or constructs their first ground station. Afterwards, purchases of these products will be considered recurring (action-based) costs and use a learning rate curve to simulate decreases in costs due to mass production.

Recurring (Action-Based) Costs

As previously mentioned, 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 launchingsatellites. Such costs aren't exactly non-recurring, but their recurrence isn't time-based so they must be modeled differently.

Both satellite and ground station costs are classified as recurring costs due to their ongoing cost to the player; however, we may model both products with dynamic pricing using a simple cost function that leverages learning rate during production. The general cost for each product may be given below:

$$C_{prod,SAT} = TFU_{LEO} \times L_{SAT} \tag{7}$$

$$C_{prod,GS} = TFU_{GS} \times L_{GS} \tag{8}$$

where TFU has already been defined as the theoretical first unit price of a given product and L is given as the learning curve factor. This factor is defined as:

$$L \equiv N^{B} \tag{9}$$

where N is the number of units ordered and B is defined with the following equation:

$$B = 1 - \frac{\ln((100\%)/S)}{\ln 2} \tag{10}$$

where S is defined as the learning curve slope (listed as a percentage). As a player requests the production of a resource, the overall cost of that resource will decrease from its TFU cost due to more efficient processes, manufacturing optimizations, and streamlining production lines.

Launch costs for Sat-Tycoon are a fixed quantity; however, a tiered system will be explained for further expansion. From literature [13], the base cost to launch a satellite into LEO is approximately \$15,500 per kilogram. This is the average value for launch vehicles from the late-1990s up until the mid-2000s (specifically the Space Shuttle). Further expansion and game dynamics can be achievedby offering tiers of launch vehicles. For instance, players could opt to pay a non-recurring LaunchVehicle 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). Recently deployed launch vehicles such as the SpaceX Falcon 9 and Falcon Heavy have a launch cost to LEO of approximately \$2,700 per kilogram and \$1,400 per kilogram, respectively. 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. The insurance can be computed using the following equation:

$$C_{ins} = 0.2(C_{SAT} + C_{launch}) \tag{11}$$

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.

Recurring (Time-Based) Costs

Finally, our 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. From literature [13], a typical gateway requires 4 shifts of 12 people operating the gateway. Their average yearly salaries with benefits are given to be \$150,000/year. We may use this information to compute the yearly (or monthly) cost of operating a single ground station and multiply this cost by the number of ground stations a player has built. See the computation below for the yearly cost of a single ground station:

$$C_{GS,oper} = 4 \times 12 \times \$150{,}000 = \$7{,}200{,}000 \tag{12}$$

This may be incorporated into a broader equation to compute the total recurring cost to a player:

$$RC_{GS,oper} = C_{GS,oper} \times N_{GS} \tag{13}$$

where $RC_{GS,\mathrm{oper}}$ is the yearly recurring cost of a player's ground stations and N_{GS} is the number of ground stations a player owns and operates. Note that this is a general equation and the time epoch will depend upon the epoch rate of the $C_{GS,\mathrm{oper}}$ value. In an optimal world, satellites would be instantaneously launched upon completion of production; however, we must account for satellites that are not launched. An additional recurring cost associated with unlaunched satellites must also be computed at every epoch:

$$RC_{SAT,Store} = N_{SAT} \times 0.15(C_{prod,SAT}) \tag{14}$$

where $RC_{\mathrm{SAT,Store}}$ is the recurring cost of storing satellites at a given epoch, N_{SAT} is the number of unlaunched satellites needed to be stored at the given epoch, and $C_{\mathrm{prod,SAT}}$ is the cost to produce a single satellite at the given epoch.

System Revenue

Specifics of customer acquisition will be discussed more in the Customer Modeling section. In thissection, we assume that the player has assigned a price for their internet service in each area that they operate a ground station. Furthermore, we assume that the Customer Model allocates a set of customers to this player based on the function of the Customer Model. Note: this set of customers may be an empty set (this would correspond to a player with infeasible or uncompetitive internet service). Given these assumptions, computation of a player's revenue may be computed and assigned each month using the following simple equation:

$$R_{M,N} = I_{M,N} \times N_{Cust} \tag{15}$$

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.

3.3.3 Game Design.

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.

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 to a state 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, illustrated in Figure 11, 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.

WELCOME TO SAT-TYCOON 101! LET'S SETUP THE GAME BY CREATING YOUR FIRST CONSTELLATION!

TELL ME, HOW EXPERIENCED ARE YOU WITH SATELLITE CONSTELLATIONS?

FUNDS: $500,000,000

PLEASE BUILD YOUR FIRST GROUNDSTATION!

NEW YORK CITY, NY
LOS ANGELES, CA
AUBURN, AL

FUNDS: $500,000,000

MASS: 250 KG
LIFESPAN: 60 MONTHS
POWER: 1.5 KW
COST: $400,000

FUNDS: $495,000,000

ALTITUDE: 1000

FUNDS: $415,000,000

PLEASE SET AN INTERNET PRICE

FUNDS: $115,000,000

Figure 11. Representative zero-turn game sequence

Game Progression

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.

3.3.4 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 as well. As discussed in Section 3.3.2, satellites, ground stations, and launch vehicles may all be unlocked if their respective technology level is researched. A primitive sample of the technology tree design is shown in Figure 12.

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 expensive 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 [13].

Technology development not only unlocks a new aspect of game complexity, but also allows for more realistic modeling capabilities and scenarios. As technologies such as phased array antennae and satellite interlink technology become more standard, more competitive advantages will be afforded to those that invest strategically in select technologies. The design of Sat-Tycoon's Technology Progression allows for exactly these types of competitive studies to be conducted in the future.

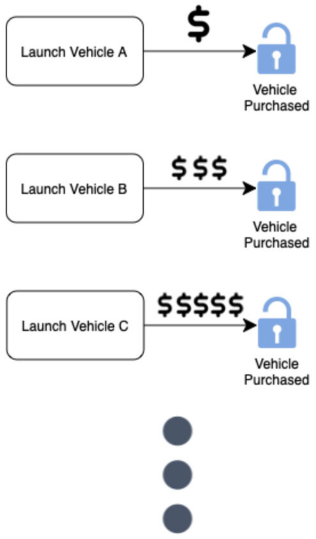
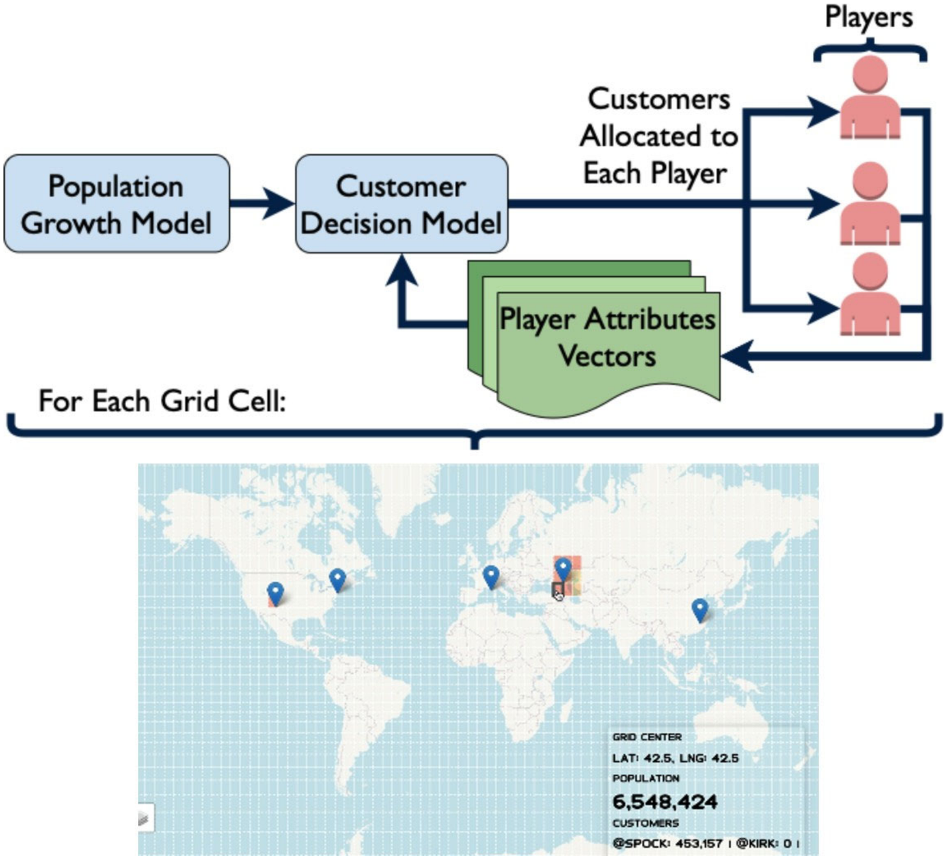


Figure 12. Sample design of our technology development roadmap

3.3.5 Customer Modeling.

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 13. 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 on 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.



GRID CENTER
LAT: 42.5, LNG: 42.5
POPULATION
6,548,424
CUSTOMERS
@SPOCK: 453,157 | @KIRK: 0 |

Figure 13. Customer Model Flowchart

Customer Behavior Model

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 [14].

The general Technique for Order of Preference by Similarity to Ideal-Solution (TOPSIS) method, proposed by [15] and outlined by [16], is used to solve the MADM problem for each customer type, in each grid cell. This method works as follows:

Step 1. *Matrix presentation*

The MADM problem can be written in matrix form with the rows indicating competing alternatives (players) and the columns indicating the different attributes considered in the problem. For the current version of Sat-Tycoon, we consider the following player attributes: offering price, data rate, data throughput, number of interruptions, longest interruption, and average latency between a random satellite in the constellation and a ground station. Equation 16 presents a MADM with n different attributes and m different alternatives,in which each entry of the matrix, x_{ij}, indicates the performance (raw value) of alternative (player) i for attribute j. This matrix varies temporally and is updated once per virtual month.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{16}$$

Step 2. *Normalization of performance ratings*

Once a MADM matrix is formulated, we must normalize the performance ratings of each attribute, such that they can be compared on an equal basis. We take the approach of [15] in which cost and benefit attributes are presented. Cost attributes are those that should be minimized for better performance and benefit attributes are those that should be maximized for better performance, both from the customer's perspective. In the MADM problem considered for our project, internet price, number of interruptions, and longest interruption are all cost attributes, while data rate, data throughput, and company reputation are all benefit attributes. Equations 17 and 18 show the formulations of normalized cost and benefit attributes used in the TOPSIS method, respectively.

$$r_{ij} = \frac{\max_i\{x_{ij}\} - x_{ij}}{\max_i\{x_{ij}\} - \min_i\{x_{ij}\}} \tag{17}$$

$$r_{ij} = \frac{x_{ij} - \min_i\{x_{ij}\}}{\max_i\{x_{ij}\} - \min_i\{x_{ij}\}} \tag{18}$$

We note that, through this normalization process, each attribute is expressed as a value in the interval [0,1]. These normalized values are irrespective of attribute type since the larger the r_{ij}, themore it satisfies the j-th attribute.

Step 3. *Weighting of attributes*

Weighting of attributes is an important component of the TOPSIS method since different customer types will have different priorities regarding attribute importance during their decision-making process. We may define a weighting vector to reflect each different customer type's preferences regarding the attributes in the MADM problem. This weighting vector is applied to the normalized matrix from the previous step and computed as:

$$v_{ij} = w_j r_{ij} \tag{19}$$

where w_j is defined as the weight (or importance) applied to the j-th attribute by the customer and v_{ij} is the weight-adjusted and normalized entry within the MADM matrix.

Step 4. *Identifying ideal and negative-ideal solutions*

The ideal solution, A^*, and negative-ideal solution, A^-, are defined by Equation 20 and Equation 21, respectively.

$$A^* = \left\{ \left(\max_i v_{ij} \mid j \in J \right) \mid i = 1, \dots, m \right\} = \{v_1^*, \dots, v_m^*\} \tag{20}$$

$$A^- = \left\{ \left(\min_i v_{ij} \mid j \in J \right) \mid i = 1, \dots, m \right\} = \{v_1^-, \dots, v_m^-\} \tag{21}$$

where J is the set of all attributes. The ideal solution is therefore a theoretical vector that takes thebest attribute values available from all alternatives and the negative-ideal solution is the theoretical vector of the worst attribute values available from all alternatives.

Step 5. *Distance calculation*

The distances from each alternative to the ideal solution and the negative-ideal solution are computed using the Euclidean norm and defined formally in Equation 22 and Equation 23, respectively.

$$S_i^* = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^* \right)^2}, \ i = 1, \dots, m \tag{22}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^- \right)^2}, \ i = 1, \dots, m \tag{23}$$

Note that S^* and S^- are now vectors describing the ideal and negative-ideal distances for each of the m alternatives, respectively.

Step 6. *Similarity calculation*

The similarity of each alternative is then computed using the following derivation:

$$R_i^* = \frac{S_i^-}{S_i^- + S_i^*},\ i = 1, \ldots, m \tag{24}$$

Note that the similarity of each alternative will always be in the interval [0,1] due to the nature of the derived similarity.

Step 7. *Ranking and selection*

Finally, we use an *argmax* function on the vector of similarity scores, R^*, to obtain the alternative with the highest similarity metric. The player chosen by the TOPSIS method is then assigned all the customers associated with the customer type in the evaluated projected square.

3.4 Software Implementation

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

3.4.1 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), whichis a network communication architecture for games in which clients communicate with a server that maintains a master data set [17]. 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 server authoritative architecture 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 14 schematically depicts the server authoritative architecture that is implemented in Sat-Tycoon.

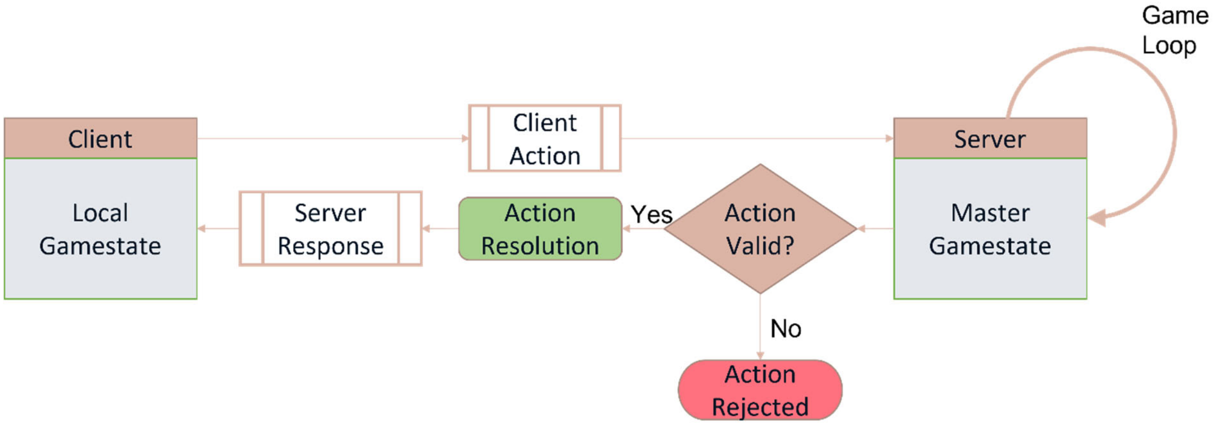


Figure 14. Schematics of a Server Authoritative Architecture

Additionally, Sat-Tycoon backend server (i.e., that 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.

3.4.2 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 15 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 agentdevelopers.

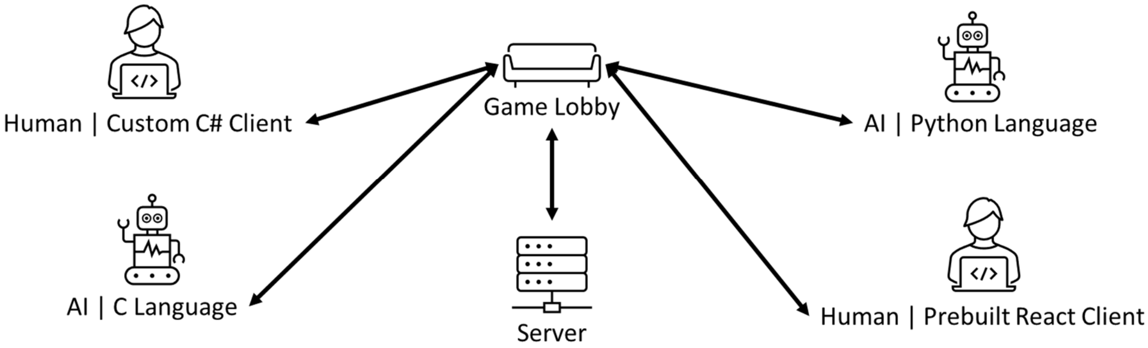


Figure 15. Example of possible player connections that are enabled by a WebSocket API

3.4.3 Graphical User Interface and React Client.

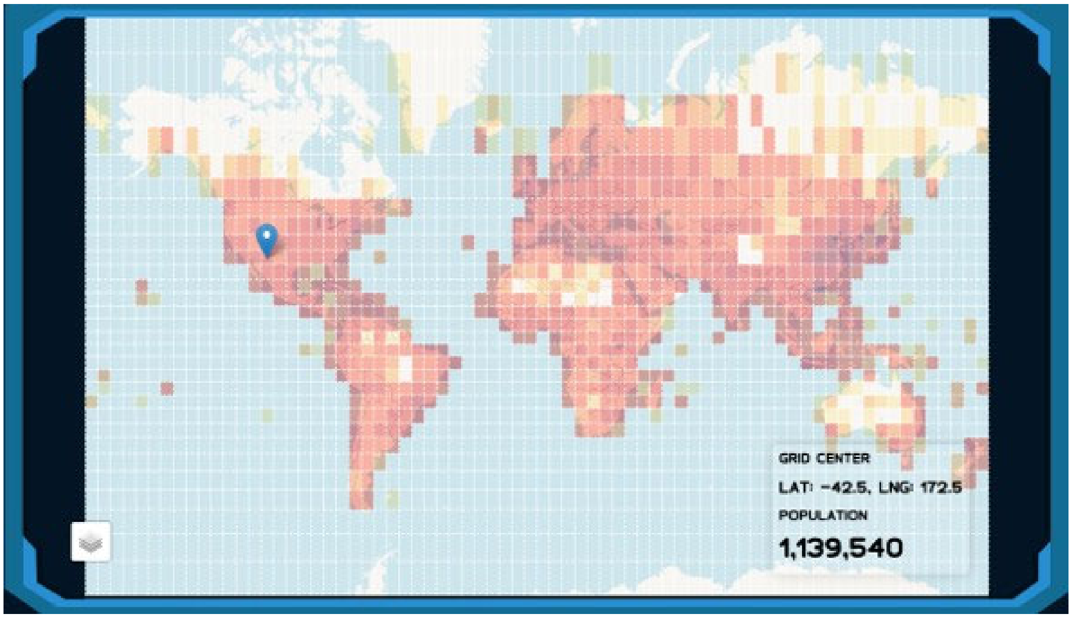
The Sat-Tycoon React Client offers a graphical user interface (GUI) for human players who may not want to build their own client or develop an AI agent. This default client for human players isbuilt in React with Javascript, provides a GUI, and handles all WebSocket communication with theserver. In the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

Ground Map

The gamification of P-LEO constellation 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 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 16. 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: 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: Coverage is displayed as heatmap bands across the map. The darker the color of atile, the more satellites are passing over the area on average.
- Data Rate: 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.
- Population: 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.

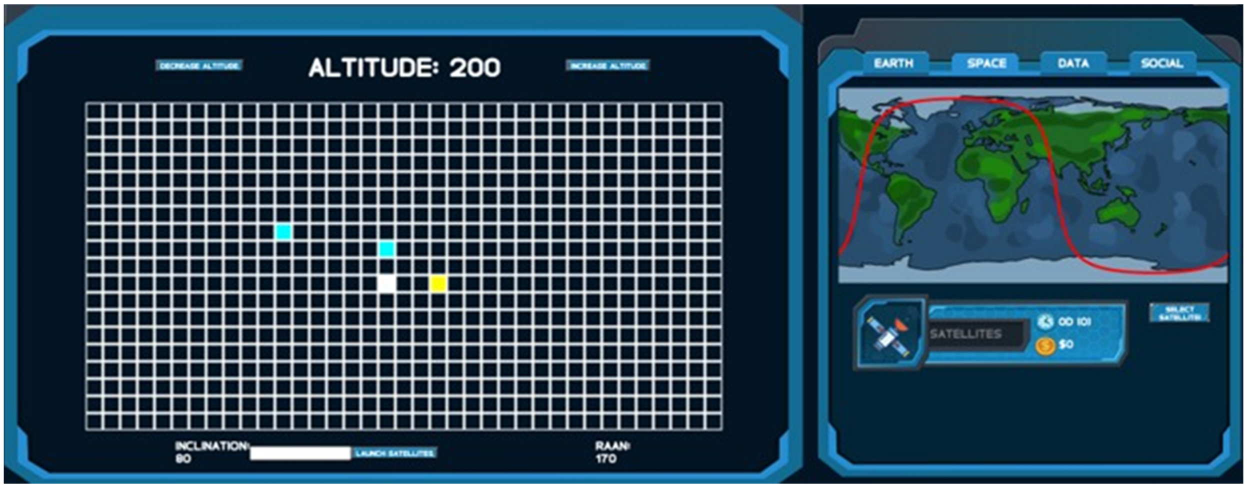


GRID CENTER
LAT: -42.5, LNG: 172.5
POPULATION
1,139,540

Figure 16. Sample color-coded grid map displaying global population distribution that is part of the Sat-Tycoon graphical user interface

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 17) 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.



ALTITUDE: 200

INCLINATION: 80

RAAN: 170

Figure 17. Sample Space Deck Implementation

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 piece of 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 18). A system to allow players to buy improved technology increases gamification by providing depth and a sense of progression.

MASS
100 KG

POWER
2 KW

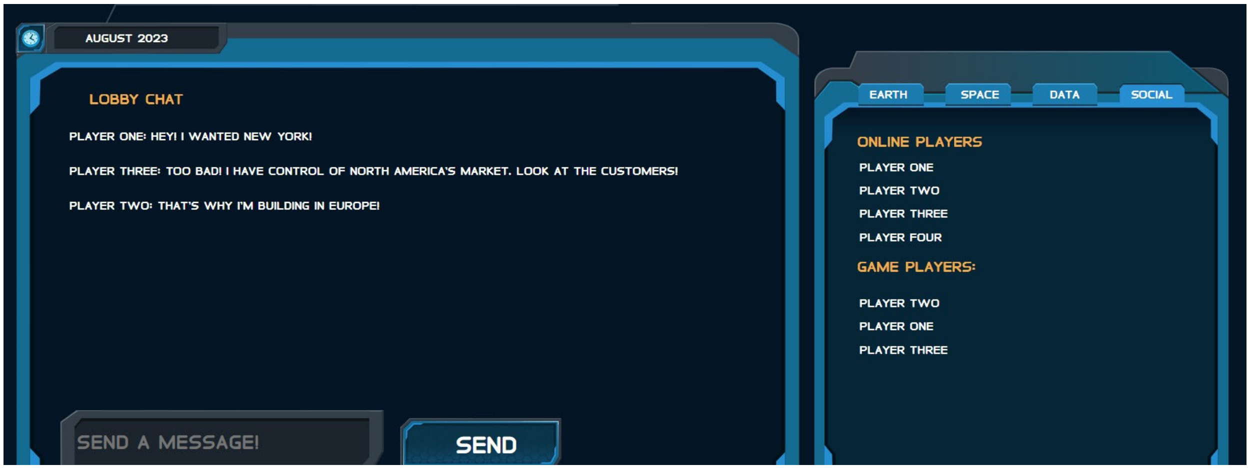
LIFESPAN
72 MONTHS

COST: $1,800,000

Figure 18. Sample Technology Tree with Comparisons

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. 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 (See Figure 19). 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.



PLAYER ONE: HEY! I WANTED NEW YORK!

PLAYER THREE: TOO BAD! I HAVE CONTROL OF NORTH AMERICA'S MARKET. LOOK AT THE CUSTOMERS!

PLAYER TWO: THAT'S WHY I'M BUILDING IN EUROPE!

Figure 19. Example of multiple players discussing their Sat-Tycoon game session

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 shown in Figure 20. 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 explained in more detail below.

GROUNDSTATION NUMBER 1

LOCATION: 40.7128, -74

MINIMUM ELEVATION ANGLE: 30

MAX ECA: 25

MIN ECA: 0

MIN NADIR:

SERVICEABLE CUSTOMERS: 1000000

CURRENT CUSTOMERS: 500000

DATARATE: 1000

AVERAGE GDP: 350000000

Figure 20. Prototype Analytics Tab

- Earth: Analytic data for Earth includes customer, ground station, data rate and population data.
- Space: Analytic data for space includes orbital, lifespan, and coverage data.
- Finance: Financial data includes revenue, GDP, and expenditure.
- To Be Determined: There is room to expand and add more analytical data in the future.

4. RESULTS AND 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 21, 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 22 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.

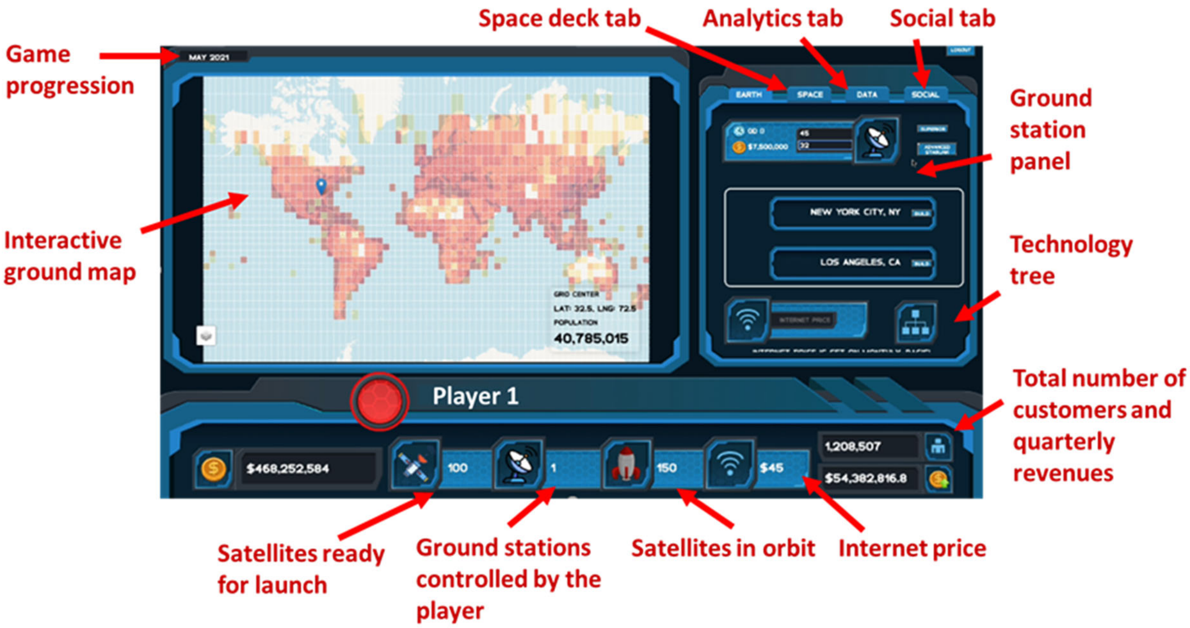
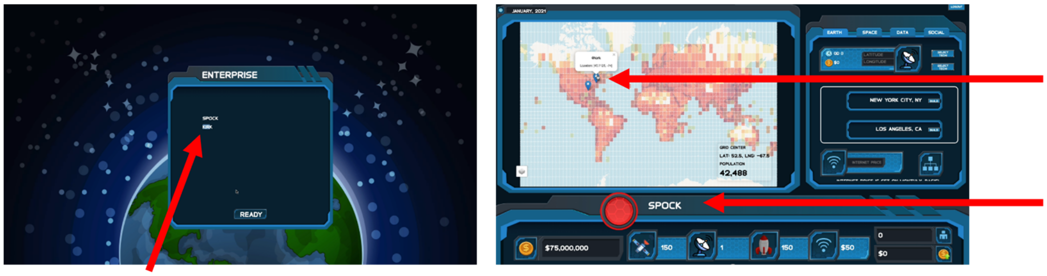


Figure 21. Prototype of Sat-Tycoon hosted on Auburn University server

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.


Player two's ground station

Player one

Two players in the lobby joining the game session

Player one visualizing information on player two's ground station

Figure 22. Example of multiple players joining a Sat-Tycoon game session

4.1 Verification & Validation

Verification was conducted in a two-fold approach: project verification and overall game verification. Project verification has been defined to be the successful development of a gamified framework that informs basic strategies in the satellite internet marketplace. Game verification has been defined as the successful implementation of requirements necessary for a minimum viable product in the scope of a playable game. Table 1 illustrates the initial project scope while Table 2 lists the requirements developed internally to deliver upon the project goals.

Table 1: Project Task Tracking & Verification

Goal/Task	Description	Completion/Alteration
WORK 1	***Game objects, attributes, and rules***	***Completed via Game Design***
TASK 1.1	Implement an object database for the game	Completed via JSON files
TASK 1.2	Implement a game engine in python using PyGame libraries	Altered to use a custom game loop without PyGame
TASK 1.3	Identify all possible player actions and their effects on the game state	Completed via Python game mechanics
TASK 1.4	Estimate the cost of each action in terms of game resources	Completed via Python game mechanics and cost functions
WORK 2	***Constellation engine***	***Completed***
TASK 2.1	Model the population distribution across the globe	Completed using a base global population and a population growth function for each nation
TASK 2.2	Determine the world distribution of end-user's maximum buyable capacity	Estimated using the Customer Behavior Model
TASK 2.3	Implement a constellation engine that, provided orbit shell information, computes coverage figures of merit for each city in the database	Completed via the developed Constellation Engine
TASK 2.4	Implement the coverage analysis using parallelization	Completed via the developed Constellation Engine

Table 1: Project Task Tracking & Verification (continued)

WORK 3	***Expected utility function and bidding agents***	***Completed via Customer Model***
TASK 3.1	Develop and implement an algorithm that determines end-users' behavior	Completed using MADM problem formulation and TOPSIS solution method
TASK 3.2	Write a utility function for each decision factor	Completed using MADM problem formulation and TOPSIS solution method
TASK 3.3	Test the end-users' behavior	Completed using MADM problem formulation and TOPSIS solution method

To validate the Sat-Tycoon game, a classic game development approach was taken to model, design, and validate different stages of the game. Figure 23 illustrates the development cycle of most typical videogames with specific sections added, removed, or iterated upon many times. We define the Concept/Prototype Version as a sample/mockup in which we have made modeling decisions and identified critical game elements. For Sat-Tycoon, this included conducting trade studies and design reviews to understand the game mechanics. In this version, a multiplayer, strategy game with a finite game length was chosen. Additionally, the metrics to gauge a player's progress were identified to be their overall profits throughout the span of the game and, indirectly, their customers.

Table 2. Internal Requirements for Sat-Tycoon Game

Requirement ID	**Requirement**
T0-GEN-R0	The game between 2-15 human players with the actions, costs and revenues **ACTIONS:** Create company Build satellites Build ground stations Launch satellites Allocate resources to recurring operations cost Decommission satellite Decommission ground station Set internet selling price **COSTS:** Satellite total production cost Non-recurring launch cost Non-recurring ground station cost Satellite recurring operations cost Ground station recurring operations cost

Table 2. Internal Requirements for Sat-Tycoon Game (continued)

	REVENUES: Initial capital Revenues from serviced customers
T1-MOD-R0	The game state of the game described in T0-GEN-R0 shall be unambiguously described by a JSON archive
T1-MOD-R1	All actions in T0-GEN-R0 LIST A shall map to a change of the game state archive defined in T1-MOD-R0
T1-MOD-R2	All mapping function defined by T1-MOD-R1 shall be documented
T1-MOD-R3	All cost items in T0-GEN-R0 LIST B shall be described as a function F(GAME STATE,ACTION)=COST
T1-MOD-R4	Modeling Team shall provide a justified estimate for the initial capital
T1-GEN-R0	Customers shall be model as reverse-auction bidding agents with utility function mapped to coverage properties in LIST A and market properties in LIST B **LIST A:** Number of connections(satellites in line of sight) Satellite Throughput Signal latency (e.g., altitude) Average Coverage gap Average Max response time **LIST B:** Selling price Inertia Regional GDP per capita Random behavior
T1-MOD-R5	Number of service buyers shall be mapped to revenues (see T0-GEN-R0 LIST C)
T1-GEN-R1	The world map shall be divided in a 5 deg by 5 deg grid of square cells
T1-GEN-R2	Properties listed in LIST A and LIST B of T1-GEN-R0 shall be available for each cell of the world map
T1-DEV-R0	The principal investigator and another human player other than the development team shall be able to play the game described in T0-GEN-R0 with objective: highest revenues in fixed time.
T1-DEV-R1	Players shall play the game described in T0-GEN-R0 via a fully functional user graphical interface. Fully functional means that the players are able to intuitively perform all actions in T0-GEN-R0 LIST A, intuitively access all relevant game state information specified by T1-MOD-R0, intuitively access map cell information specified by T1-GEN-R2, intuitively track all cost in T0-GEN-R0 LIST B, and intuitively track all revenues in T0-GEN-R0 LIST C
T1-DEV-R2	The game engine shall implement all mapping functions between action and

Table 2. Internal Requirements for Sat-Tycoon Game (continued)

	game state as specified by T1-MOD-R1
T1-DEV-R3	The game engine shall implement all mapping functions between (game state, action) and cost as specified by T1-MOD-R3
T1-DEV-R4	The game engine shall implement all mapping functions between number of internet subscribers and revenues as specified by T1-MOD-R6
T1-DEV-R5	The user interface shall include a world map as described in T1-GEN-R1 and T1-GEN-R2
T1-DEV-R6	The game engine shall include artificial agents to render the customers described in T1-GEN-R0
T1-DEV-R7	The game defined by T0-GEN-R0 shall be implemented as simultaneous (w/ cooldown)
T1-DEV-R8	The game-time to real-time ratio shall be user-configurable with nominal value 1 hour (game) to 10 years (real)
T1-DEV-R9	Market information shall be fully available to each player. Technology/constellation attributes specs and revenues shall only be available to the constellation owner.

The Alpha Version develops upon the Concept Version and demonstrates an actual framework of the game while adding modeling details and iterating upon implementation and testing to ensure a stable game. Most of the development time of Sat-Tycoon was spent in this phase designing and implementing customer behavior models, intuitive user interfaces, and expandable player and game models. Decisions in this version were more specific, sometimes requiring multiple iterations of a design or model before convergence. Specifically, different versions of the customer objective functions were explored before converging on a MADM formulation (which was chosen to allow expandability and modularity of attributes being considered). Different user interfaces involving the visualization of space resources were also developed before converging on the current space deck implementation (which was chosen due to its intuitive nature).

We define the Beta Version as one in which we have successfully conducted Beta Testing with volunteers and are able to refine and tune game elements such as progression speed and difficulty. Finally, the Release Version is meant to be the end of the overall development of Sat-Tycoon in which no additional features, tuning, or testing is required. In the current state of the game, we have defined formal details and have been through many iterations of testing, modeling, and development to reach a Beta Version. Additional progress will be made through a Beta Testing Phase with a potential playability study in the future.

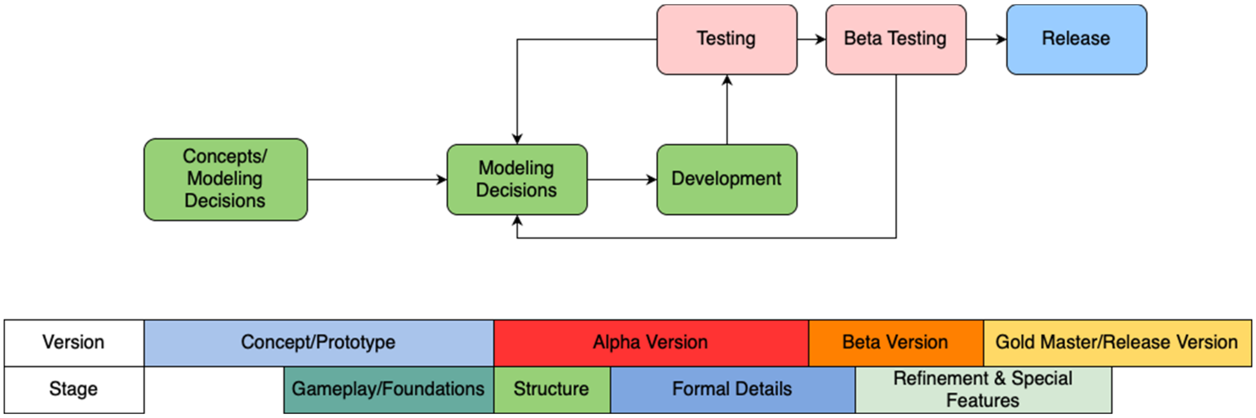


Figure 23. Development Process utilized in Sat-Tycoon

The methodology for game validation follows recreations of simpler instances and additional testing to confirm basic business and game theory strategies such as first-movers-advantage, dominance markets, and price undercutting. Further validation of the game involves comparisons of analysis provided by the game and specific analysis conducted by conventional means to solve individual resource management, constellation performance analysis, and customer estimation problems. Such conventional means may require more time than the gamified method.

4.2 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.

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.

4.3 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.

4.3.1 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 (altitudes at which satellites operate) 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 cost 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 costs opponents residual costs 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.

4.3.2 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 modeling technologies that do not exist in order to simulate technology progression. With technology tokes 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.

4.4 Interfacing with AI

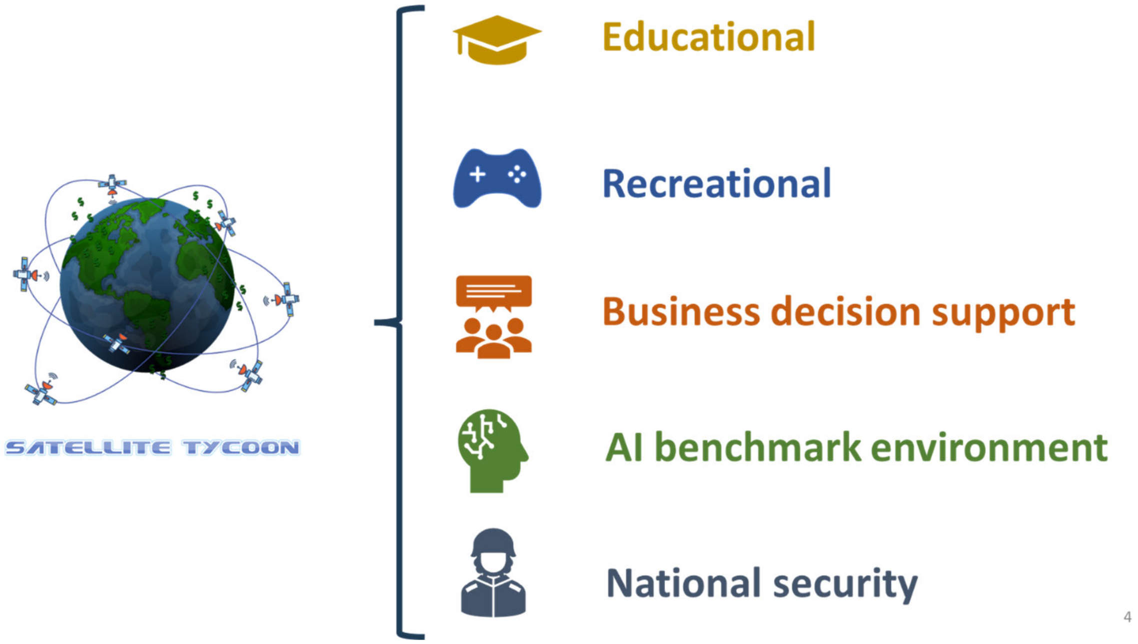
The Sat-Tycoon API provides a starting point to interface the game with AI players. The server does not have any special requirements for AI agents, and AI clients communicate with the server using the same Websocket API that player clients use. For example, an AI gym environment for reinforcement learning 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.

5. CONCLUSIONS

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 conjuncture 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 due to standard security protocols that the University adopts. An internet facing server may be possible with further development to ensure that the security of the University network is preserved.

With further development, a gamified framework to simulate the interactions between constellation operators and virtual customers may aid 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 24 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 assets servicing. A gamified frameworkfor P-LEO assets servicing may include logistics dynamics based on responsive launching, refueling, refurbishing, repairing and debris removal, both with a competitive or collaborative gameplay.



- Educational
- Recreational
- Business decision support
- AI benchmark environment
- National security

Figure 24. Five possible, longer-term pathways for Sat-Tycoon development

6. 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 ofgamification 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 enterpriseSNS, In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12*, Seattle, Washington, USA, 2012.

[10] 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.

[11] Larson, Wiley J. and Wertz, James R., eds. Space Mission Design and Analysis, Boston: Microcosm, Inc., pg 733, 1992.

[12] Lovell, Robert, "*The Design Trade Process*," Lecture Notes from MIT 16.89 Space Systems Engineering class, Massachusetts Institute of Technology, Cambridge MA, February 1995.

[13] Harry Jones, The recent large reduction in space launch cost, 48th International Conference on Environmental Systems, 2018.

[14] 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.

[15] Gwo-Hshiung Tzeng and Jih-Jeng Huang. *Multiple attribute decision making: methods and applications*, CRC Press, Boca Raton, FL, 2012.

[16] 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.

[17] 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*: 20 November 21.

7. LIST OF ACRONYMS & GLOSSARY

CapEx – Capital Expenditure

FCC – Federal Communications Commission

GUI – Graphical User Interface

ISP – Internet Service Provider

MADM – Multi-Attribute Decision Making

OOP – Object Oriented Programming

OpEx – Operating Expenditure

P-LEO – Proliferated Low Earth Orbit

RAAN – Right Ascension of Ascending Node

SAA – Server Authoritative Architecture

TOPSIS – Technique for Order of Preference by Similarity to Ideal Solution

DISTRIBUTION LIST

DTIC/OCP
8725 John J. Kingman Rd, Suite
0944 Ft Belvoir, VA 22060-6218 1 cy

AFRL/RVIL
Kirtland AFB, NM 87117-5776 1 cy

Official Record Copy
AFRL/RVS/Richard "Scott" Erwin 1 cy