Simulation of an Automated Tow Tractor Operation for First Person Training

Gabriel Font ¹, Jordi Pons-Prats ², Javier Mora ¹, and Felipe Muñoz La Rivera ¹

¹Affiliation not available
²Universitat Politècnica de Catalunya

October 30, 2023

Abstract

The Taxibot® is a hybrid electric vehicle that can tow medium to large aircraft in the movement area of an airport. Its use can lead to lower noise levels near terminal buildings and reduce fuel consumption and emissions; Aircraft emissions and cost of the aircraft ground operations are reduced significantly. The Taxibot®, which is the name that receives the vehicle developed by IAI (Israel Aerospace Industries), has been deployed in very few airports around the world, but the number is envisioned to increase in the following years. If the system goes fully autonomous, pilots will have to learn and accustom themselves to the differences when performing the procedure with the vehicle and airports need to make studies on the effect it could have on capacity and safety. The aim of this paper is to describe the development of a simulator of the operations of a remotecontrolled tow tractor for first person training. The objective of the simulator is evaluating and improving human factors in terms of required skills and ergonomics. The study takes into account a review of the current state of the art on available tools and their shortcomings to establish requirements of the proposed solution. The steps taken to create the virtual environment, physics simulation and other systems of the implementation are disclosed. Validation of the simulated experience was conducted with users of different expertise levels. Although the validation methodology was limited in scope, most users reportedly became more aware of the processes involved in the operation of a tow car. Additionally, the experience can be updated to test new vehicles and conditions.
Simulation of an Automated Tow Tractor Operation for First Person Training

Gabriel Font\textsuperscript{a}, Jordi Pons-Prats\textsuperscript{a,b}, Javier Mora\textsuperscript{b}, Felipe Muñoz La Rivera\textsuperscript{b,c}
\textsuperscript{a}Universitat Politècnica de Catalunya BarcelonaTech (UPC), Physics department - Aeronautical Division, Campus PMT UPC, c/ Esteve Terrades 5, Edifici C3, 08860 Castelldefels (Barcelona), Spain
\textsuperscript{b}Centre International de Mètodes Numèrics a l’Enginyeria (CIMNE), Campus Nord, Edifici C1, c/ Jordi Girona s/n, 08034 Barcelona, Spain
\textsuperscript{c}School of Civil Engineering, Pontificia Universidad Católica de Valparaíso, Av. Brasil 2147, Valparaíso 2340000, Chile

Abstract—The Taxibot\textsuperscript{®} is a hybrid electric vehicle that can tow medium to large aircraft in the movement area of an airport. Its use can lead to lower noise levels near terminal buildings and reduce fuel consumption and emissions; Aircraft emissions and cost of the aircraft ground operations are reduced significantly. The Taxibot\textsuperscript{®}, which is the name that receives the vehicle developed by IAI (Israel Aerospace Industries), has been deployed in very few airports around the world, but the number is envisioned to increase in the following years. If the system goes fully autonomous, pilots will have to learn and accustom themselves to the differences when performing the procedure with the vehicle and airports need to make studies on the effect it could have on capacity and safety. The aim of this paper is to describe the development of a simulator of the operations of a remote-controlled tow tractor for first person training. The objective of the simulator is evaluating and improving human factors in terms of required skills and ergonomics. The study takes into account a review of the current state of the art on available tools and their shortcomings to establish requirements of the proposed solution. The steps taken to create the virtual environment, physics simulation and other systems of the implementation are disclosed. Validation of the simulated experience was conducted with users of different expertise levels. Although the validation methodology was limited in scope, most users reportedly became more aware of the processes involved in the operation of a tow car. Additionally, the experience can be updated to test new vehicles and conditions.

Index Terms—Taxibot\textsuperscript{®}, autonomous tow car, tow tractor, simulated experience, virtual environment.

I. INTRODUCTION

TAXIING is the process by which an aircraft moves along the ground, usually before takeoff or after landing. Aviation has a high environmental impact [1] [2] and around 30\% of emissions in a large airport can be traced to taxi operations [3] [4] [5]. Engineers are constantly looking for new technologies and systems to reduce fuel consumption. Electric tow tractors are on the table as a possible mean to tow the aircraft along the taxiways. The Taxibot\textsuperscript{®} is one of the proposed solutions, developed by Israel Aerospace Industry (IAI)\textsuperscript{1}, and commercialised by Smart Airport Solutions (SAS)\textsuperscript{2}. It is a vehicle that can tow medium to large aircraft at conditions comparable to those of a regular taxi procedure. The final aim could be the get a fully automation, which could require a full automation of the airside. This is one of the reasons, the Taxibot\textsuperscript{®} would be controlled by the pilot using the regular tiller for nose wheel steering, rudder pedals and brakes within the cockpit, although the initial tests used a remote operator. Instead of using the main engines of the aircraft that burn jet fuel, the Taxibot\textsuperscript{®} uses a hybrid electric engine to reduce fuel consumption considerably. Additionally, its use during taxing can achieve a significant reduction of ground noise at airports [6]. The first Taxibot\textsuperscript{®} model has been approved and certified for some of the most used family of aircraft. The benefits and drawbacks of the Taxibot\textsuperscript{®} utilisation have been analysed and each new deployment on an existing airport is studied laboriously [7]. While the technology has been tested, the Taxibot\textsuperscript{®} is in an early stage of the adoption process. Since the nose landing gear is not designed to deal some specific loads that the Taxibot\textsuperscript{®} could produce, specific training is required. Several studies should still be conducted before full Taxibot\textsuperscript{®} deployment, and a large number of pilots and operators must be trained in the use of this kind of vehicles.

In this communication, a first person simulation of the Taxibot\textsuperscript{®} experience is developed in order to research the training-related aspects, attempting to resolve related to the gamification of the experience, the interaction with the user, and the overall training experience, while discovering unexpected issues or benefits that may arise. Several flight simulation programs exist nowadays and they recreate all the manoeuvres in a regular flight extremely well. Most courses in pilot training employ flight simulators as they can provide very similar environments without any risk involved at a fraction of the cost. Some airlines are even implementing Virtual Reality technologies to train pilots, cabin crew, and operators. However, the Taxibot\textsuperscript{®} is not featured in any openly available flight simulation software today.

The proposed experience could be used to train pilots and operators in the use of this new technology. Following the specific implementation, the experience was tested to study its effectiveness as a learning tool.

\textsuperscript{1}https://www.iai.co.il/p/taxibot
\textsuperscript{2}https://www.taxibot-international.com/
II. BACKGROUND

A. Use of Simulation and Virtual Reality in Aviation Training

The use of flight simulation software in the aviation industry is extremely common [8]. The FS1 Flight Simulator was launched in 1979 for the Apple II. It placed the user in the cockpit of a biplane fighter so the experience could be considered to have a first person view. Throughout the following decades, more advanced software was developed to aid in training, development or simply user entertainment.

Many companies offer full flight simulator devices that can re-create aircraft procedures and motion to achieve experiences incredibly similar to their real-world counterparts. The training effectiveness of flight simulators has been proven decades ago [9] and their use is well established in pilot training.

These devices are often very expensive and the software underneath is proprietary, so implementation of new technology or control over different experiences is limited.

In April 2021, the European Union Aviation Safety Agency (EASA) approved the first Virtual Reality based Flight Simulation Training Device (FSTD) [10]. The device was developed by VRM Switzerland\(^3\) and it is currently used to train rotorcraft pilots. The adoption of Virtual Reality (VR) enables more cost-effective experiences that can be used to complement full flight simulators and other training tools.

For instance, the dutch airline company KLM has begun training pilots, cabin crew and ground personnel using VR [11]. They have developed first person experiences that place the user inside the aircraft, on the apron or inside a pushback tractor [12]. The company shows interest in continuing to create VR experiences as training components and obtaining EASA certification for the developed courses.

As opposed to approved FSTDs, amateur flight simulation is very popular and several options are readily available for ordinary users and not just pilots in training.

B. Gamification of Training Processes

Gamification is becoming an important trend in simulation to ensure the fulfilment of the training experience. Larson, [13], describes a serious game as a strategy to ensure not only the engagement of the learners, but also to increase the learning experience including failures and repetitive attempts. Communication provides a comprehensive overview of gamification and corporate training. Armstrong and Landers [14], provide an interesting review of the studies on the effectiveness of gaming as training and learning strategies, while providing a roadmap for the training design. Similarly, Suh et al [15] describe the difficulties to implement gamified training in the workplace while proposing strategies to ensure engagement and upgraded learning experience.

Becker [16] proposes that the design of successful educational games should take into account instructional models present in successful commercial games and not necessarily serious games. Many existing principles of educational theory can be found in popular commercial games, for example, the nine events of instruction by Gagné et al. [17] which include gaining attention, providing guidance, assessing performance and enhance retention among others.

C. Perspective and Presence in a Virtual Environment

The perspective of the pilot from within the cockpit is first person view (egocentric view). The location of other objects is relative to the self. A third person view in contrast could be the viewpoint of a flying bird above or from behind the vehicle (allocentric view). The vehicle being controlled by the user is then considered an object and its position references other objects. According to Fabroyir and Teng [18], users perform faster and are less prone to errors while navigating a virtual environment using egocentric navigation. The third person perspective however presents other advantages such as facilitating far-scope navigation [19].

Presence is the acceptance of being in a virtual environment, even if the user is physically elsewhere. There are several factors that contribute to the feeling of presence, such as freedom of actions, understanding sizes, distances and orientation in the virtual environment and realism [20]. Whether or not presence is important for task performance is a point of contention between different authors [21] [22] For entertainment however, the sense of presence is a design choice and it will depend on the goals of the experience. For training applications, Regenbrecht and Schubert [23] argue that a high realism of the virtual environment assists in the transfer of learning to the real world.

D. Taxi Operations and use of the Taxibot\(^\text{®}\)

The need to reduce noise and pollution in airports and their vicinity helped the development of several taxi solutions, from the Taxibot\(^\text{®}\), used as reference on this study, to electrical motors on the nose gear, all being assessed in [24]. Among them, the Taxibot\(^\text{®}\) can be considered the most similar one to the standard tow tractors, following the same vehicle concept and having similar characteristics. The Taxibot\(^\text{®}\) entered regular operation at Frankfurt Airport in 2015. The vehicle has been deployed in Indira Gandhi International Airport (Delhi) and Kempegowda International Airport Bengaluru (Bangalore) for a few years. Indira Gandhi completed 1000 movements using the Taxibot\(^\text{®}\) in May, 2021 [25] Amsterdam Airport Schiphol has been studying and testing a deployment since July 2020. One of its programs aims to achieve sustainable taxiing by 2030\(^4\) and it expects several Taxibot\(^\text{®}\) to join the fleet at the start of 2022. Originally, the use of autonomous tow tractors was an objective of the project but it was not achievable given the target schedule.

Autonomy is a popular proposed approach to reduce taxi times, delays, and human errors. The Taxibot\(^\text{®}\) is an important milestone to tackle the subject of autonomy. A remotely controlled or unmanned Taxibot\(^\text{®}\) is achievable but a long development and verification phase is expected [7]. Although testing of an unmanned vehicle was estimated to start in 2020, no statement has been released. Developments on the subject

\(^3\)https://vrmswitzerland.ch

\(^4\)https://www.schiphol.nl/en/schiphol-group/page/sustainable-taxiing-schiphol/
of autonomy or control from a remote location are also slowly being carried out. However, no operation or test has been disclosed yet.

In 1998, NASA developed an integrated cockpit display solution to aid pilots in ground operations. The display of the Taxiway Navigation and Situation Awareness (T-NASA) system adapts both a first person view augmentation (on a Head-Up Display) and a third person view (Electronic Moving Map) to increase route awareness [26]. Augmented reality displays were deemed most appropriate as it was designed for low visibility use cases but no visibility cases would require virtual reality displays. Additionally, a decision not to display much information in the moving map as it might increase head down time instead of looking out the cockpit. These are some of the topics that needed to be accounted for during the design process.

III. RESEARCH OBJECTIVES

The main objective of this research is to develop a simulator of Taxibot® (tractor vehicle) Operations for First Person Training with the aim of evaluating and improving the human factors in terms of ergonomics and required skills. This objective can be broken down in the following sub-objectives:

1) Review of the existing procedures for conventional and tractor vehicle taxi operations as well as the design of interactive experiences with training purposes.

2) Modular design of the application to provide a suitable framework for straightforward replication of the simulator for different airports and vehicles. This design choice also facilitates the addition of other functionalities at a later phase.

3) Implementation of a specific use case, including:
   3.1) First person vision of the Taxibot® driver and the aircraft pilots.
   3.2) Visualisation and execution of the nose landing gear locking process, taxi procedure from boarding gate up to the holding position, unlocking and return route.
   3.3) Inclusion of different factors like different environments, vehicle dynamics, visibility and weather conditions, different plane sizes.

4) Evaluation of the best compromise between graphical fidelity, approximation of the physical systems, fluidity, cost, and portability of the simulation.

5) Definition of a validation method to analyse the extent to which the disclosed objectives are met.

IV. EXPERIENCE DESIGN

A. Requirements and Optional Systems

The series of functional requirements that have been foreseen for the simulator are the following:

- The experience must be able to track the steps and route taken by the user in order to evaluate possible errors they might have committed. Depending on the severity of the infraction, the simulation could come to an unsuccessful end. Performance analysis should be simple and automatic.
- An integrated solution to allow the user to check the relevant aerodrome charts. Since just the taxi phase is contemplated, at least the Ground Movement Chart must be included.
- A networking solution that allows another user to evaluate the pilot in real time should be implemented.

Additionally, a few optional systems that can be implemented to enhance the experience are listed.

- Control over the time of day and weather conditions during the experience.
- Porting the experience to a virtual reality device.
- Support for different peripherals to enable more accurate control. For example, a joystick or a steering wheel provide analogue controls that are not possible simply using a keyboard.

B. User Engagement and Difficulty

The amount of previous knowledge required to take part in the experience and find it useful will determine the overall level of difficulty. In order to carry out a taxi procedure, the pilot and co-pilot need to follow a very long sequence of steps. The assistance that the experience can provide to a beginner will determine the entry level of knowledge of the user. The objectives, setting, and mission should be explained in a short tutorial prior to the procedure. The experience is a piece of interactive audiovisual media, so a written document explaining the use of signs and markings in an airport, specifications or separation criteria could be overwhelming. In order to help maintain user engagement, a different solution should be studied to aid beginners to find their route and undergo the experience.

The three channel model of flow proposed by Kiili [27] asserts that if a challenge is lower than the skill level of the user, the player might feel bored. On the other hand, the player might grow confused or anxious if the challenge is greater than expected. Kiili proposes a zone of proximal development, which is the ideal flow state where skill level increases with the presented challenge. This zone of proximal development might be extended if guidance or help from other players is provided. This statement served to define one of the main features of the designed application.

C. Decision Criteria

Two decision criteria have been considered: Location Selection and Taxi Procedure.

1) Location Selection: To increase its value as a learning tool, the experience should not be a perfectly straightforward example of a taxi procedure. To add some difficulty, the airport where the experience takes place should enable obstacles or events that can come into play and disturb the operation. Based
solely on this assessment, an airport with a higher complexity is preferable. However, unless a 3D model of such an airport can be acquired easily, a higher complexity airport would require a longer period of development time (creating models and textures).

Valencia Airport (also known as Manises Airport) was selected because it is a smaller, single runway airport and the estimated time to create the 3D model and textures is relatively short. Valencia Airport was the eighth busiest Spanish airport in 2019 and the third considering only single runway aerodromes. The final traffic number came to 8,539,403 passengers in 2019 [28].

Only runway 12/30 is currently in use so the experience features a flight departing from this runway. The runway is 3215 meters long and 45 meters wide. The taxiway N (parallel to the runway) is 23 meters wide and the paved area is 44 meters wide. Both measurements for the runway and taxiway correspond to ICAO (International Civil Aviation Organization) code letter E. The taxiway to runway distance is 200 meters. The airport can then operate with heavy aircraft like the Boeing 747, B777, B787 or Airbus A340.

2) Taxi Procedure: The airport has already been selected because of its size and complexity. Between the two possible directions of runway 12/30, 12 was selected because the number of possible conflicts is larger when compared to the other configuration. A section of the Aerodrome Ground Movement Chart shown in Fig. 1 illustrates one of the possible hurdles of the experience. Following taxiway H4 instead of turning right on N2, a pilot might accidentally cause a dangerous runway incursion. This is called a hot spot, a location with a history or potential risk of collision or incursion.

![Fig. 1: Detail of a possible conflict zone in the Ground Movement Chart.](image1)

There are no specific limitations on the subject of taxi speeds. Instead, the aircraft manufacturer will recommend speeds that allow for safe stopping distances, turns, and avoiding brake degradation. For example, the Boeing 737 Flight Crew Training Manual states that a normal taxi speed is approximately 20 knots, with a maximum of 30 knots in straight sections and around 10 knots in high angle turns. The Taxibot® specifications are quite clear in this regard. The maximum speed of the vehicle under load is set to 23 knots. The distance from the terminal to the holding position H8 is approximately 2500 meters. Considering a speed of 20 knots in straight areas of the route and 10 knots on turns, the time an aircraft would take in an uninterrupted taxi procedure to move this distance is almost exactly 5 minutes.

Different events that a pilot might experience and add difficulty to the procedure could be implemented to add learning potential. These include low visibility, the presence of obstacles on the taxiway or runway, coming across other aircraft or vehicles and getting instructions to change the initial planned route for the procedure. The instructions to perform the procedure will use correct aviation English terminology and vocabulary when possible. While this aspect makes the experience more realistic, it brings the problem of reducing the target audience since some previous knowledge on aviation communication is expected of the user. If the airport zones, taxiways, runway and obstacles are all modeled correctly and the program can place the Taxibot® and aircraft that the user is controlling within those zones, the entire route from starting position to the holding position can be changed rather quickly.

For this implementation, the base selected taxi procedure is the following: “[Callsign], taxi to and hold short of RWY 12 via Gate C N H8. Contact tower on 118.55 when ready.” The user would then need to understand the instructed route from the terminal gate to the holding position. The pushback procedure is performed by the Taxibot® operator with help of the ground crew since the pilot cannot see behind the aircraft. To be able to simulate this part of the procedure correctly, the same user or another would need to play the role of Taxibot® operator initially and they should have a more complete view instead of the view from the cockpit. The pilot would take over from then on and follow the appropriate centrelines to piece together the route as is shown in Fig. 2.

![Fig. 2: Route of the designed operation.](image2)

A problem that presents itself as a result of the use of the Taxibot® is the fact that taxiways in most airports are not prepared to have a vehicle that needs to return to the apron. A pushback tractor disconnects from the aircraft while still on the apron and it can move safely back to a hangar or another aircraft that requires pushback. A Taxibot® however, should be able to disconnect from the plane on the holding position and return safely to the apron. The taxiway will most likely be in use already for a different operation, so an alternative needs to be in place. An aeronautical study to find an optimal solution needs to be made in any airport that is considering a Taxibot® implementation. But, as it escapes the scope of this
According to the ICAO SARPs (Standards and Recommended Practices) document [29], the correct minimum distance from the existing taxiway centreline for object separation is 43.5 meters to maintain the code letter E.

**D. Proposed Experience**

The proposed experience detailing the type of activities the user should take in order to receive information and apply it on the virtual environment to advance is shown in Fig. 3. This flowchart relies on the automatic evaluation system mentioned in the requirements section. More intricate scenarios could be created when the mantle of traffic controller is taken by another user connected to the virtual environment instead.

![Flowchart of the proposed experience](image)

**V. IMPLEMENTATION**

**A. Software**

Development of an experience in any engine can be a laborious project and it depends on how much work can be saved. There are several software applications available that can provide the services that were required for the implementation but these were selected because of familiarity.

- Unity: One of the most popular game engines currently available. It is often considered to be flexible and easy to get started. It has a free version as well as paid licensing options.
- SketchUp: A very simple 3D modelling software application. It offers a free version as well as a paid version and it is available as a web-based application on any browser. The program makes use of the 3D Warehouse, a place where users can sell their models or offer them for free. Thanks to its popularity, countless assets like existing buildings, vehicles and sections of world terrain can be found here.
- Adobe Photoshop: Adobe Photoshop is probably the most recognised image editor available. Since 2013, it changed its licensing scheme to software as a service. A user can subscribe to the Adobe Creative Cloud for a monthly fee. Numerous images need to be created for the experience and they are placed as textures or UI elements.

**B. 3D Models**

1) **Modeling and Adapting**: A 3D model of the Taxibot® could not be procured. Fortunately, the Narrow Body model is very similar to other towbar-less tractors, so a model of the Komatsu WZ4000 was used instead. Its size and proportions were modified to better resemble the Taxibot® [30].

![Komatsu WZ4000 model to be used as the Taxibot®](image)

As for the aircraft, there are countless options available on the Warehouse. For example, there are over one thousand free results from the query “Airbus A320”. The Boeing 787-9 Dreamliner was selected simply because a good model was available [31] and it can operate on Valencia Airport. It is not certified for Taxibot® operations at the moment.

Since the experience must undoubtedly place a camera that simulates the point of view of the pilot, the interior of the cockpit is also needed. A model of the cockpit that includes the flight display and other instruments was fortunately also available from the 3D Warehouse [32]. The model had to be modified slightly because its scale was not entirely precise. The inner walls, windows, floor and seats were modeled to fit both the plane and instruments 3D models.
Many small elements in the models were removed because they would not be visible and they were only increasing the file size and rendering load. The original plane model was 16 MB and the cockpit model was 7 MB. The final model that merges them together where I added interior details is 11 MB in size. In total, this 53% size reduction was achieved removing unnecessary textures, details on the fuselage like pitot tubes, and remaking the seats and smaller objects to reduce polygon count. This example of optimisation is useful to make portability to mobile devices possible.

2) Importing Models: Textures are images that are added to surfaces on 3D models to define color and details. Several maps like the diffuse map, the normal map, reflection map and others determine how the surface looks under a certain light and at a certain angle. Many texture examples can be found in the Unity Asset Store or other free sources online.

After the 3D models are imported into Unity (in one of the several supported formats), materials that use these texture maps need to be created and assigned accordingly.

3) Environment Recreation: Initially, the option to procure the model of an airport with correct markings in place was considered. Unfortunately, none that meet the criteria was found. The airport model was created from scratch and it took around 60 hours to complete.

The resolution of Google Earth images is sufficiently high to be used directly as textures on certain areas of the recreated environment. However, thin lines like centrelines and smaller text seem blurry, especially when viewed at an angle. Images captured in Google Earth were used as reference to recreate markings and signs as well as provide the measurements of all elements for the 3D model.

The model was imported into Unity and textures for the different surfaces were assigned. A terrain object was created to add some elevation and it was painted with grass, dirt and sand textures. These assets are included in the Terrain Tools Sample Package, created by Unity and available for free.

C. Physics Simulation

Unity has built-in tools to make the development of the physics simulation quite straight-forward. The tools revolve around GameObjects, components and the functions and classes they contain. Both the Boeing 787 and the Taxibot® objects require colliders to detect when they come into contact with other colliders and Rigidbodies to define mass, center of gravity, and allow them to receive forces. In object-oriented projects such as this, the Nvidia PhysX physics engine is used. For the Taxibot®, the Rigidbody component was configured with a mass of 27 000 kg. The mesh collider component uses the same imported 3D model but the convex option is enabled. This option simplifies the mesh to its bounding convex shape as it saves on performance. The four wheel objects themselves use a different 3D model and they only serve visual purposes.

Objects with the Wheel Collider component are used to define physical characteristics of the wheels. This collider component is included in the Unity libraries and it suffers from a few shortcomings. Other, more complete wheel components and scripts can be found on the Asset Store but at this
moment, none of them are free. Parameters like the wheel mass, suspension distance and spring constant, forward and sideways friction all need to be playtested thoroughly. The script that controls the Taxibot® needs to read the user inputs and apply forces to the wheels to simulate the behavior of the vehicle. Different functions that handle the engine, steering and brakes are in place. They change the torque and steering angle (built-in parameters of the Wheel Collider class) according to the previously explained input variables. The Taxibot® specifications mention that the vehicle can operate using traction in two wheels or all four wheels. A flag to change this parameter was included. The initial engine parameters were obtained translating the power specifications to torque considering a wheel of around 1.4 meters in diameter. Afterwards, tuning was required to achieve the dynamics shown in reference videos. The maximum speed of 12 meters per second derive from the maximum speed of 23 knots under load. The maximum steering angle was a value that could not be procured form the specifications. The value that was obtained is one that ensures that the maximum steering radius is approximately that of the unassisted Boeing 789. This value is closely related to the clamping mechanism of the aircraft nose wheel and how the joint behaves.

Up until this moment, the Taxibot® has been implemented as a very simple car. The interaction with the plane is one of the most important aspects as the accuracy of the simulation is one of the main attributes to validate the experience. Each of the different groups that make up the model has its own mesh collider. This allows for movement of each object independently, separating them from the plane object or collapsing the entire model. The plane object has an assigned mass of 180 000 kg, which is a takeoff weight within the acceptable range. The plane object also makes use of Wheel Collider components on the landing gear.

A Hinge Joint component is in place to connect it to the Taxibot® RigidBody. The position and axis need to be configured, as well as some limits. The maximum angles that the joint can take are defined by the aircraft specifications. The fact that steering is no longer directly driven by the nose wheel changes the steering maximum angle. The minimum nose turning radius of the aircraft was employed to define the Taxibot® wheel steering and Hinge Joint limits.

One of the shortcomings of using the built-in Wheel Controller objects emanates from the fact that wheels can "go to sleep" if they are resting for a few seconds to save on resources. The translation of forces that allow the Taxibot® to push or pull the airplane works sufficiently well in movement. However, at the start of the simulation, the landing gear wheels are resting and the forces applied by the Taxibot® can never move them. This bug has been known for a few years and it requires a workaround. A similar script that reads the power input was used and assigned to the Boeing 787 GameObject. If the Taxibot® is accelerating, then a very minor torque is also applied to the airplane wheels that contact the ground.

While the development of these scripts was straightforward, a meaningful amount of time was spent tuning every parameter. In total, approximately 30 hours of work were spent writing the described scripts, tuning and testing.

D. Events

Besides following the initial instructions to complete the taxi route, elements that disrupt the procedure were designed and implemented. Both a pilot and a ground vehicle operator have a responsibility to avoid collisions. However, controllers exercise their best judgment to avoid conflicts and their instructions are mandatory.

Two similar events were added to the experience. A scripted message from the ground traffic controller issues instructions to maintain separation from a fire fighting vehicle and a smaller leading aircraft on the taxiway N3. The vehicles are objects that are activated on proximity and move along a route at set velocity profiles.

E. Automatic Ground Movement Control

In order to guide the pilot to perform the taxi procedure correctly, the experience includes a system that tracks the position of the user controlled vehicles. In an airport environment, the pilot interacts with the traffic controller that is in charge of ground movements via radio. The simulated traffic controller sends similar messages using proper aviation vocabulary that can help the user.
Different areas of the airport were modelled using SketchUp and the original composite satellite image obtained from Google Earth. These simple models are imported into Unity and labelled using a simple script that defines a zone name and calls a function when a collision with either the Taxibot® or the Boeing 787 is detected. For this purpose, a messaging system was added and placed in the Head-Up Display (HUD). The user can receive messages from the simulated traffic controller using a panel on the left. “AFR 421 stop immediately. You have not been cleared to enter the runway.” is an example of one of these messages. These messages were later recorded and the audio files are played at the same moment the respective message is sent.

It is important to note that many mistakes that can be committed in the experience are more dangerous than the traffic controller suggests. A taxiway excursion can damage the landing gear very seriously and the flight would likely need to be evacuated. Failing the experience would require the user to start a new attempt entirely. Instead, a scoring system was designed to aid the pilot in recognising if a mistake has been made.

Points that reward users who accomplish a task correctly is one of the basic gamification strategies. A scoring system that makes use of the detection and instructions scripts was implemented. Apart from route mistakes, disregarding separation criteria and leaving the taxiway unexpectedly, other aspects like centreline accuracy and time to finish the procedure also influence score to a smaller degree. The points themselves are arbitrary, but the consequences of their inclusion are discussed in the validation stage.

![SUCCESS!](image)

**Fig. 10:** Results panel with the score obtained throughout the experience.

Once the procedure is finished and the user stops at the position where the Taxibot® will disconnect, the score and all the factors behind it are displayed. If all areas of an airport are modelled, imported and labelled, different ground operations could be implemented. To create a new trajectory, the changes to make are the starting and final positions, creating links between the erroneous zones and the script in charge of scoring, and setting the appropriate messages. In the end, this evaluation approach is very limited. Each new mission needs to be developed attentively, and they constitute a significant time investment, especially if there are many messages in text or audio form. The fact that the user cannot communicate with the traffic controller removes a very important aspect of airside operations.

**F. Networking**

During development and initial tests, People with low qualification in airport procedures were lost and they could not find the described route. In these cases, I would explain the use of centrelines and indicate their position on the ground movement chart. With minimal indications, the experience was much more manageable for a larger audience. The automatic controller system cannot provide dynamic indications, so a different solution was explored. Networking was implemented to connect two or more users to the same virtual environment. The official Unity multiplayer solution is under development at the moment. Many functionalities have not been implemented yet, so an alternative was required.

Mirror is an API (Application Programming Interface) built for Unity, by third-party developers. It is open source and it can be imported as a package for free. Its use in Unity is based on pre-built components that can be added to GameObjects to add functionalities like synchronising variables, sending commands to execute functions on the server or a client, enabling and flagging sections of code as “server only” or “client only”.

With this functionality, short additional scripts that indicate which user has authority over an instance of an object can be assigned. The same Taxibot® and aircraft objects can be reused in this new mode. A simple object for the traffic controller that can toggle different views was created subsequently. After connecting to the server, the user selects a profile from the UI panel that can send the command to create an instance the corresponding object.

![Testing two profiles, pilot and controller connected to the same virtual environment.](image)

**Fig. 11:** Testing two profiles, pilot and controller connected to the same virtual environment.

This system was designed keeping in mind that it should be easy to implement new profiles in the same environment. To test this, I added a new vehicle monitoring the time it takes me. I decided it would be a small aircraft, so a free model of a Cessna 172 was downloaded from the 3D Warehouse.
The new model was swiftly imported into Unity and its materials were assigned. A script using mouse and keyboard controls to fly the aircraft was procured online [34] (free under MIT License). Some parameters like forces and mass needed to be configured but it was an overall simple process. From the moment the 3D model and the script were found online to the moment the Cessna 172 was flying in the virtual environment, the process only took 35 minutes.

G. Additional Functionalities

Several more functionalities were implemented to improve usability and user experience. These include a main menu where the user can select the mode (automatic traffic controller or another user through the network functionality) and a tutorial section.

A toggle to display the mapped airport zones to the user was added in a pause menu. Being able to see the zones that should be avoided and the requested route simplifies the experience to users with little knowledge of airport maneuvering and situational awareness.

Additionally, control using a steering wheel and pedals were implemented. A free Software Developer Kit package to integrate Logitech devices was available from the Unity Asset Store. After studying an implementation example, all that was required was to read the controller data and assign the variables to the correct axes and buttons.

H. Experience Flow

In case the user selects the single player experience with automatic traffic controller, the full flow can be divided in the previously mentioned tasks.

VI. Validation

The purpose of the validation section is to analyse the extent to which the objectives have been met. To do so, the experience was tested and evaluated by users of different levels of knowledge on the subject. A survey was designed to obtain mostly qualitative data on the experiences of each user. The setup consisted of a computer using a steering wheel and pedals for the test user performing the operation playing the
role of pilot. Meanwhile, a different computer was connected using the network functionality to play the role of traffic controller monitoring the user. This arrangement allowed me to solve doubts in case the automatic controller messages and chart were insufficient. The percentages are calculated over a total of 16 users who tested the experience and answered the questionnaire minutes later.

- 87.5% of the users were able to complete the experience following the given instructions with varying scores reflecting their performance.
- Most of the test users considered they did not have much knowledge on the subject of airport ground movement. However, the majority of users stated that they were familiar with video games or other software that makes use of mouse and keyboard control schemes.
- A qualitative assessment of the perceived value of the experience as a learning tool was gathered through the question “After the experience, do you consider yourself more aware of the aspects that a pilot should keep in mind when undertaking a taxi procedure using the Taxibot®?”.

![Reported Awareness Chart]

Fig. 18: Awareness following the experience. 1 means the experience was not very useful, 5 means they are highly aware now.

- As positive feedback, 93.8% of test users stated that they were engaged throughout the experience. The most popular enjoyable aspects were graphical fidelity, physics simulation, and the sense of controlling the Taxibot® and aircraft using the steering wheel and pedals.

As for constructive feedback, the questionnaire included the question of “How do you think the experience could be improved or expanded?” The most common recommended feature was implementing a different and maybe larger environment. The “add more vehicles” suggestion can be grouped together with the “adding take-off and flight” prompt. It is important to indicate the evaluation session took place before the implementation of the Cessna 172.

One of the test users was a pilot who obtained his commercial pilot license in 2021. I was very interested in any further insight he could provide and this is an attempt to summarise it.

- He indicated that the dynamics feel well recreated. While he had never participated or seen a Taxibot® operation, the acceleration of the aircraft feels very similar to that of a similar weight class aircraft.
- Turning the aircraft is different to what he was used to as the tiller movement is not directly translated to the nose wheel. Instead, the wheels on the Taxibot® turn first and the axle on the docking mechanism follows, which adds some lag when changing directions.
- No additional indications or information were required. Following the automatic controller instructions and making use of the provided chart were sufficient to complete the procedure.
- Different conditions for similar procedures could be implemented, such as changing the time of day or reducing visibility conditions.

Unfortunately, the session could not reach more pilots or pilots in training as these opinions seem very valuable. A thorough evaluation methodology could compare results in two groups of users, one being the experimental group and the other being the control group. The experimental group would test the experience and the control group would receive a theory session with a similar duration covering the same subject. Testing awareness before and after the test would also provide a useful metric to show in the comparison. Such a methodology was not feasible as there were few participants on a single simulator setup. Most of the questionnaire focused on qualitative aspects of the experience, such as entertainment and value as a training tool. Quantitative aspects of the virtual experience that are affected by design choices, for example performance, capacity or safety were not measured as they are not the focus of the experience. The validation aspect could certainly be expanded upon at a later stage.

VII. Conclusion

The following conclusions have been reached by the results and analysis described above:

- A simulator of a Taxibot® and aircraft in a recreation of an existing airport has been created and it can be replicated in less than one month of time (150 work hours) by a single developer. This estimate considers that most assets like 3D models and textures are pre-made and development is focused on the core simulated experience.
- The validation sessions and processes were limited in scope, so it is hard to assess to what extent the experience can be utilised as a training tool. With that said, most test users relayed that the experience made them more aware of the processes involved in a Taxibot® procedure and that the experience was engaging, especially with more than one user.
- Besides training for pilots, crew members or even traffic controllers, the tool can be used to design and test runway or airport configurations for safety or capacity if the virtual environment is crafted. For example, measuring
the effect of adding additional beacons and markings or response times of emergency services.

- The simulation developed can be considered as the "foundations" of simulators with Taxibot, i.e. procedures are provided to replicate the development with (a) Different airports and runway configurations, as well as Taxibot models (which are direct and immediate); (b) Incorporate more casuistry, some of them of an extreme nature due to different adverse weather situations or incidents such as runway accidents; (c) Experiment with complementary signalling to reinforce safety, and other types of pilot assistants, such as simultaneous translators, among other variables that interest those interested.

- It is currently the only available software that features the Taxibot®. Other examples of new technology would be straightforward to implement as a new profile given the modular systems in place. Existing game engines offer the ability to build content such as the presented experience without necessarily incurring a massive development time. While simulation and games are customary in aviation, serious games still have a high potential for training purposes in aviation and other fields.

- Given the trends of working in virtual environments, such as the Metaverse, this type of development contributes towards the massification of these technologies, familiarising users and showing the potential uses and alternatives for speeding up and massifying the construction of these experiences and environments.

On the subject of future lines of research, there are several options to expand the project:

- Porting the experience to different devices, such as smartphones or virtual reality (VR) headsets. Heavy optimisations would need to be made to allow the application to run on smartphone hardware but it is most likely possible. Implementing VR requires a new input reading system to be developed but it could be ported in a short amount of time.

- Having more options and control over training content. For example, operations at night, during a thunderstorm, or with low visibility would not be hard to implement in terms of the systems that need to be developed. There are free or paid packages and learning material about simulating night, rain and fog conditions in Unity.

- Improve the data gathering system that runs during the operation to evaluate how the users are learning and showing the potential uses and alternatives for speeding up and massifying the construction of these experiences and environments.

- The involvement of more concurrent users in the virtual environment through the network implementation could create more valuable educational situations.

ACKNOWLEDGMENT

I would like to thank members of online communities that created 3D models, tutorials and scripts that assisted in the development of this project and all other creators who contribute assets to the SketchUp 3D Warehouse and Unity Asset Store. The second author, JPP, is a Serra Hunter Fellow. This research has been partially supported through the Severo Ocho Centre of Excellence (2019-2023) under the grant CEX2018-000797-S funded by MCIN/AEI/10.13039/50110001033. Finally, the authors would like to thank the reviewers for the comments that helps improving the quality of the manuscript.

REFERENCES


