Do A Barrel Roll, But In Real Life

Joshua Pulido
jpulido@oxy.edu
Occidental College

1 Problem Context

With the rapid development of technology in both the virtual reality and drone fields, what were once niche experiences have now become household brands, especially thanks to companies like Meta and DJI. That begs the question: why has there not been software that brings the two spaces together in a entertaining way? While there are certainly methods for using VR headsets to view drone footage, very little effort has been put into pushing the space further and innovating in the augmented reality field using drones. My goal is to be among the first to look into the space; more specifically, I plan on creating a video game that has the player fly a real world drone, while interacting with a virtual environment using the other buttons of a game controller.

Before talking about the game itself, it is notable to mention that such a game would bend the definition of augmented reality (AR) as commonly known today. The most popular AR games, including the popular game Pokemon GO, often focus on overlaying digital objects on top of the real world, thereby ”augmenting reality”. Virtual reality (VR) loses the ”reality”, instead focusing on creating an immersive virtual world that feels as real as possible. While augmented reality is not limited to smartphones, other devices such as Microsoft's HoloLens have not yet reached the commercial viability of their VR counterparts. This is where the use of a drone, with it's included camera, comes into play. By combining streaming the video feed from a drone to a VR headset, an AR game becomes possible where it otherwise would not be; being able to physically fly the drone around comes as a bonus that adds to the entertainment value and allows for interaction with the real world.

It is important to note that we are still relatively early in what will likely be a long history for both the VR/AR and drone spaces of technology, and as such the two are not necessarily designed to function together. Nevertheless, I have multiple ideas for how to make them cooperate, which I will go more in depth on in the Technical Background section. The obstacles may be difficult to overcome, but through a series of compromises, I believe that a prototype game can be developed, with the goal of creating a more integrated system in the future. There are also ethical considerations that must be made, such as whether or not inexperienced pilots should be allowed to fly drones, especially while trying to play a game at the same time. Within the scope of this project, it may not be possible to address such issues; however, in the creation of a full game that would be released to the general population, they would more than likely be dealt with through a required training program before being allowed to pilot the actual drone.

2 Technical Background

In an effort to capture the feel of the first person view (FPV) headsets that are traditionally used to fly drones, it makes the most sense to use a VR headset. Notably, recent advances in mobile technology have led to a split in the VR headset field, with some headsets requiring a connection to a computer and others having the ability to be completely independent. The ideal way of making this game would be the development of an application that runs natively on an ”all-in-one” headset, most likely the Oculus Quest 2. However, depending on other factors that could come into play, it might not be possible to run this type of game on Android based systems. In this case, it would make the most sense to use a PC to run the game alongside a headset such as the Valve Index. Regardless of which platform ends up being preferred, the most obvious solution for developing a 3-dimensional (3D) video game would be Unity.

Although there are many popular game engines, the two that come to mind for 3D video game development are Unity and the Unreal Engine. I have chosen Unity because of its integration with the C# language, for which there are many APIs that allow for computers to interact with drones. Unity also provides many built in tools that help simplify the game design process. While I am still unsure about exactly what the gameplay itself will look like, one important feature I want to include is the procedural generation of levels. There are algorithms out there that can create some very detailed levels, but since my game will have to be relatively simple, I imagine that I will end up making my own. Currently, my idea is to create an algorithm that takes the dimensions of the environment as input, up to a max value equivalent to the drone's range. Then, using a random number generator and pre-established rules for game features, I hope to create a graph, where each node is a feature, such as an enemy or a power-up, and the weights are the distance between each item. The final part of the algorithm would just perform a clean up, again based on rules that it is fed. With that being said, there is still a lot more to determine before writing this algorithm, such as where the inputs are collected. Ideally, this would be done using sensors on the drone itself, but not every drone has access to distance-sensors. There is also the challenge of mapping the real world to the virtual environment by defining what 1 meter in the real world means in game. It is common for games to exaggerate in game physical values to make them more fun and engaging, but in the case of this game, doing so could potentially detract from the immersiveness. Therefore, there will have to be a balance between the entertainment value and the realness that the game offers. There is also the potentially difficult and non-trivial issue of getting a drone to interact with a computer.

Currently, the industry standard for piloting drones is through the use radio frequencies to transmit the required data to control a drone. This includes the instructions given to the drone by a controller, as well as the video feed given to an FPV headset by an onboard camera. Therefore, interacting with most drones in the market through a computer would require radio-capable hardware that is uncommon on PCs. With that said, drone company DJI has set a precedent of using the WiFi-standard 2.4 and 5 Ghz frequencies for their drones, allowing computers to easily connect through their WiFi chips, which potentially eliminates an entire step in the process. It is important to note, however, that latency plays a huge issue when working with any device over wireless communication, and in this case it is especially important to evaluate the latency of the video stream. One study on latency[3], performed by researchers at the National Institute of Information and Communications Technology in Yokosuka, Japan, found that the use of the 2.4 GHz radio band caused major latency issues due to the large amounts of interference. Although their research focused more on beyond-line-of-sight flight, the latency still poses an issue at closer distances, and worsens with the amount of wireless devices in the area. Theoretically, even with the interference and latency issues, the video output should be sufficiently fast enough to provide for a playable experience, but in practice this could end up not being the case. While it would be preferred to have a perfect product, compromises will clearly have to be made to make this project possible. For this reason, I have decided to use DJI and Ryze's Tello drone, which has a public SDK and a large developer community. While the video quality is not the highest resolution, at 720p, the latency should be reasonable within approximately 100 meters from the player. This might restrict what is possible with gameplay somewhat, but it should allow for a working end product, and support for other drones could be added after the fact, although that is likely outside the scope of this prototype.

3 Prior Work

In the department of video games that interact with real drones, very little has been accomplished. Only a single commercial game can be found using search engines that matches the description of this type of game. Named Drone Prix AR[5], this game by developer Edgybee was the first known attempt at creating an augmented reality game using a drone, in this case developed for the Android and iOS platforms. Despite lacking the immersiveness that a VR headset would bring, the game largely accomplishes a similar feel to my idea. However, it is important to note that the game no longer exists on both Google's Play Store and Apple's App Store, and there is very little evidence that the game ever existed. Perhaps this is not by coincidence - creating a game like this is something that few have done before, and given the current size of the drone market, such a product is likely not commercially viable. Nevertheless, the game and its idea, while novel, is one that many would gladly play. In fact, although there are very few video games that use AR technology with drones, there have been many other projects that run parallel to this idea. For example, one group of researchers[6] sought to create an augmented reality spectating application for drone FPV racing, and was written several years ago when the technology was not as capable as it is today. Their work mostly involved overlaying a visual UI and smart assistant that explained the race as it progressed, so it shares some similarities with my concept. However, they also used external motion detectors and other devices that will not be used in my project.

When it comes to working with Tello drones, the specific drone that I will be using in this project, there are many interesting projects that developers have made publicly available on Github. While most of these have nothing to do with video games, there are a couple that stand out as ones I can take some ideas from. One such project, created by user carter-james89[2], allows for the Tello to follow a ”virtual drone” in Unity, with real time video output in the Unity game scene. Although this is not necessarily a video game, it proves that interaction with the drone in Unity using the C# library[4] created by user Kragrathea is possible. Another really interesting library is Joy[1], a simple tool developed by GiovanniGrieco that allows for the Tello drone to be flown using a traditional game controller, and written in Python. Unlike most of the other projects that use DJI's own API, Joy sends packets directly to the drone, which allows for faster responsiveness, an absolute necessity for my game to be playable. I believe that these libraries will provide me with sufficient information to make the Tello work with the game, but there is an active community of developers, including an entire website, dedicated to the Tello that I can go to for any help if need be.

Aside from the technology side to this game, equally important is the actual gameplay. While I am still unsure exactly what genre of game I want to play, the core of this game will be an immersive flight experience, a necessity when a real drone is being flown. The biggest inspiration for my game is No Man's Sky, a space exploration game, particularly the space flight mechanics that exist. In the game, the player can fly around to collect resources, and can also optionally engage in combat with other starships. This is not the only game to explore this genre; however, it is one of the most immersive experiences, and also has a VR mode to add to the experience. Regardless of the actual gameplay and mechanics that end up making it into the final product, I really hope to capture the same feel of flight from No Man's Sky, where it is arcade-like yet still feels very real. Other games with similar gameplay that have inspired me include Star Wars: Squadrons and the Star Fox series from Nintendo. On the AR side of things, there have been relatively few games that have achieved commercial success, but games like Pokemon Go have proven that augmented reality can be viable if done right. Although Pokemon Go is of a completely different genre, its ability to captivate players through interaction with the real world, in this case through the use of GPS, is the driving force behind the use of a real drone in the game I will be making. The entire game itself can be made in full VR, but the addition of the camera view and real drone provide for a similar engaging feature that will add to the experience.

References

[1] Grieco, Giovanni. *Joy*. 2021. URL: `https://github.com/GiovanniGrieco/Joy`.

[2] James, Carter. *UnityControllerForTello*. 2021. URL: `https://github.com/carter-james89/UnityControllerForTello`.

[3] Kagawa, T. et al. *A study on latency-guaranteed multi-hop wireless communication system for control of robots and drones*. Tech. rep. 2017. URL: `https://ieeexplore.ieee.org/abstract/document/8301849`.

[4] Kragrathea. *Tello Lib*. 2018. URL: `https://github.com/Kragrathea/TelloLib`.

[5] Murison, M. *Edgybees Launches Drone Prix Augmented Reality Game for DJI Pilots*. 2017. URL: `https://dronelife.com/2017/05/02/edgybees-augmented-dji-drone-prix/`.

[6] Sueda, K. et al. *Research and development of augmented FPV drone racing system*. Tech. rep. 2018. URL: `https://dl.acm.org/doi/10.1145/3283289.3283322`.