CT6COPRE REPORT

UP814853

Contents

[Design 2](#_Toc41298968)

[High Concept 2](#_Toc41298969)

[Virtual Reality Considerations 2](#_Toc41298970)

[Controls 2](#_Toc41298971)

[Cockpit 3](#_Toc41298972)

[Flight mechanics 5](#_Toc41298973)

[Gameplay Mechanics 5](#_Toc41298974)

[Standard Game flow 6](#_Toc41298975)

[Plan 7](#_Toc41298976)

[Life Cycle 7](#_Toc41298977)

[Schedule 7](#_Toc41298978)

[Risks 7](#_Toc41298979)

[Coding Standards 9](#_Toc41298980)

[Version Control 9](#_Toc41298981)

[Testing Plan 9](#_Toc41298982)

[Reflection 10](#_Toc41298983)

[The artefact 10](#_Toc41298984)

[Bibliography 10](#_Toc41298985)

# Design

## High Concept

VR Apache Simulator is a fully interactive Apache helicopter simulator. The game will allow the player to fly and operate the apache as the pilot or gunner and gives them a sense of what it would be like to fly the airframe in real life.

## Virtual Reality Considerations

With the whole game being in virtual reality there are an increased consideration opposed to a standard flight simulator game.

The first consideration is the interaction of the cockpit, while a fully interactable cockpit would increase immersion it would also add to the number of objects needing to be collision checked and possibly decrease the performance. Adding to this having such a realistic cockpit would add to the time needed to learn how to fly and would make the game less accessible to a general audience.

The second consideration is the index controllers, with the index controllers you have an increased strain on your wrist because of how you hold the controllers, this makes rotating what you are grabbing increasingly difficult especially if they have forms of RSI. A way to fix this could be to develop the project to use rudders for people who have these issues but that is an expensive fix.

The final consideration is the game performance, virtual reality needs to be a smooth experience especially when flying otherwise users may experience motion sickness when lag spikes occur, the feeling needs to be that they are in a smooth experience that is not jittering around. To do this I will ensure that all scripts are necessary and are not take performance away at all.

## Controls

The grab button on the index controller will be the main interaction for any object that can be picked up and moved around in the cockpit like the Flight Stick and collective. The trigger button will be used for switches or anything else that requires a toggle on/off state, it is also used to fire the weapons. Other buttons include A to switch weapons and B to reset the seated location.

The tracking on the controllers (rotation) will be used to set the joysticks rotation which will set the flight sticks rotation to control the Apache.

## Cockpit



(Brown, 2018)



(Brown, 2018)



(glyndavies02, 2007)

The Apache Cockpit is one of the most advanced in the world, taking at least 6 months just to know how to fly the Apache (Maksel, 2012), in real life the Apache pilots were a monocle that gives them a heads up display allowing them to view instruments without needing to look away vital in combat zones (Maksel, 2012). I decided to change the monocle to a full heads up display as the monocle can cause headaches even in experienced pilots as the eyes need to work independently. So instead it is just a full heads up display making it more practical for the players.

Designing the cockpit, it was decided to keep the airframe as realistic as possible while still being able to be flown by new commers, while all the buttons would be present only a select number of them would be interactable. The flight stick, collective and throttle as well as start-up buttons and Multifunctional displays (MFD) are the only button which will be interactable.

With any current VR controller, the user will not be able to move their thumbs to different controls on the collective or flight stick. Most VR controllers have a thumb stick which has a pressable button, as the game will be intended with the use of an index controller the user will be able to press their thumb down and it will allow them to move their thumb onto another flight stick control with the axis, this allows the user even more control over the helicopter and controls. Below is an image which shows how the user can switch their thumbs between two hat switches on the flight stick



This feature is an intended fix so the user can have more in depth control over grabbed objects and the thumb has more interactable, however there is still the downside of not being able to feel the switches so the user will not build muscle memory on what each switch feels like.

Other features of the cockpit include the ability to use the controller A and B buttons for alternate actions on a flight stick like switching weapons and zooming in and out with a camera.

The design phase of the cockpit is simple due to the intention to replicate the apache cockpit and the only limiting factor is getting realistic reference images and documentation for what each button does and controls and how the apache cockpit actually functions.

## Flight mechanics

A helicopter is a unique airframe as it can take off and land in tight spaces and hover. Helicopters flight physics is a complicated subject to recreate in a video game, there is allot of maths and factors which go into helicopters flight mechanics, for the purpose of the game only a few have been created to improve the games appeal to new commers.

The most important is the thrust and collective, the helicopters blades when spinning create downwards thrust, the blades rotate to produce torque and rotate the helicopter in the direction the pilot wants to go. The collective controls the rate of pitch enabling the helicopter to go up and down and the flight stick controls the direction of the helicopter.

The helicopters also can rotate the blades without using the flight stick to enable them to hover or move in a direction while the airframe is centred.

The apache simulator will do this but ignore temperature, atmospheric pressure, barometric pressure, and the ground effect to enable a control state in which the user can easily fly. The simulator could have options to enable these in the simulation but for the project the focus will be to get the helicopter to fly around.

The one simulation aspect that will be in the project is wind, different areas of the map can push the helicopter in differing directions increasing the challenge to fly the helicopter but this is only a small amount and can easily be countered by pilots.

## Gameplay Mechanics

As the game is a flight simulator with a war helicopter it is important to make sure that the user can use the weapon system on the apache.

The first step is the ability to switch to the gunner cockpit and control the guns and missiles on the apache, the pilot only has control over unguided hydra rockets due to this a switch will be present in the cockpit to switch to the gunner view and put the helicopter into a hover.

The first step of the gunner is getting the ability to range and identify the target, done with a camera system where they will press the A button it will get the range of the current object that they are looking at, the B button will change the view from day, night vision, thermal black hot and thermal white hot. The range of the gun will affect how far the apache gun will shoot.

The next weapon system is the hellfire missiles, these missiles can lock onto any thermal source or a laser designated target, implementing this feature will be done by having targets with a toggle state, if the toggle state is true the apache can lock onto them with missiles selected and the missile will fire to the locked on target even if moving.

The camera controls the guns rotation, the camera can be moved either with a joystick or set to the players head rotation meaning they can look around. The gunner cockpit is a stretch goal for the project and is not the focus for the project, the focus of gameplay is the flight mechanics and the interactable cockpit.

## Standard Game flow

The game will start off in a briefing room where the pilots will get a briefing on the terrain, mission, weather, and friendly forces. They will then get to choose the loadout and start up the airframe, from there they will complete objectives and support friendly forces with dynamic missions from AI troops requesting their help.

The player can choose not to help these troops if they are too busy, the missions will be varying difficulties, and some can even shoot down the helicopter if the pilot is not careful. The primary game flow is brief, loadout, take off, tasking and RTB.

The players can request taskings as well and can go again infantry, tanks, APCS and other helicopters

The player can choose to have a person with them in the gunner seat, but this is a stretch goal for the project and will likely not be a main factor in the game flow.

# Plan

## Life Cycle

Due to the short prototype nature of the project the life cycle is a modified development cycle. The project will start with a 1-week planning phase. The project will then go into the development phase where most of the features for the prototype are created. The development phase will maximise time available to try and create most if not all features for the project.

The project from here will enter the testing phase to fix most of the bugs and get a working prototype of the game finished. From here the project will be complete and decided if it is viable for further personnel development.

## Schedule

As the project has a short time frame a fixed development for features will not fit the project constrains, instead the project will manage its time by having a maximum time spent on each feature which is based on an estimate of how long it should take for it to be done for a prototype, if that feature is taking too much time it will get pushed to the back of development so the whole game should be developed. As the project only has 2 – 3 months to be completed most of the time is being allocated to development of needed features. This time management is intended to stop too much time being spent on a feature which is not working, the schedule includes stretch goal features which will be created if features are quick in development.

Below is a table showing the schedule.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Minimum time | Intended time | Maximum time |
| Interactable Cockpit | 2 weeks | 3 weeks | 4 weeks |
| Flight Mechanics | 3 weeks | 4 weeks | 5 weeks |
| Weapon systems | 1 week | 2 weeks | 3 weeks |
| Targets | 4 days | 1 week | 2 weeks |
| Dynamic Mission Generator | 1 week | 2 weeks | 3 weeks |

## Risks

Risk management is intended to avoid any foreseeable risks and ensure there is a backup plan and the developers know what to do to reduce the risk if a scenario does occur.

|  |  |  |
| --- | --- | --- |
| Risk | Mitigation | Plan if it happens |
| Flood/Fire | Source control, fire drills, Computers on desk,  Offsite backups | Repair office, Replace broken computers, Work from home |
| Source Control down | Daily off-source project back up to ensure that if source control is down people still have the last changes | USB sticks,  External Drive,  Alternate source control |
| Source control lost | Off-site backups taken on a regular basis | Backup all changes made on  USB sticks,  External Drive,  Alternate source control |
| Stollen files | Source control,  Security Measures, Vetting, Key card system, Source files need to be on site to work. | Pull changes, Contact police |
| Hardware issue | Monthly check-ups, Bi-yearly changes of hardware | Replace hardware |
| Software issue | Software upgrades, Training on different software | Use backup software, Replace software, Upgrade software |
| Network wipe | Off-site backups, Network Maintenance | Reinstall network  Roll back to latest project backup/source |
| Illness | None | Spend more time on features that were missed out due to illness, look at removing some features due to lost development time |
| Sabotage | Source control,  Security Measures, Vetting, Key card system, Source files need to be on site to work. | Revert to last source control, remove necessary people, look at security procedures |

## Coding Standards

For the code upper camel case standard will be used with any public variable using understandable names to reduce the need of commentation within code. Function code will be based on what they are doing, and nothing more e.g. reset player will not reset the helicopter.

## Version Control

For the version control git hub will be used. No git flow will be used in this project due to the short time spent on the project and a sole developer remove the need to follow a git flow. Pushes will be made at the end of each day and when moving computers.

## Testing Plan

To test the project the following test plan was created and will be used during the reflection to test the project and what it can and cannot do.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Item | Expected Result | Actual Result | Conclusion | Notes |
| Helicopter Start up | Helicopter will start up when the correct procedure is followed |  |  |  |
| Helicopter Shut down | Helicopter shuts down if the engine is switched of |  |  |  |
| Helicopter Goes Up/down | When the collective is raised the helicopter will go up. When lowered it will go down |  |  |  |
| Helicopter can move horizontally | The Helicopter will move left/right forward and backwards if the flight stick is moved that direction. |  |  |  |
| Helicopter can rotate | Helicopter can rotate left or right on the spot. |  |  |  |

# Reflection

Before the Covid-19 pandemic occurred, the project was at crucial stage of development with the flight mechanics almost being finished and VR development starting, due to moving to a home environment. So far, the Apache is in game with the player sitting down inside the cockpit, the hands are also spawned in so the player could interact with the cockpit. With the flight stick in development I ran into issue with the development of the flight stick, Unreal 4’s Steam VR plugin works by parenting the object you grab to the hand grabbed, the object grabbed will be an actor and when grabbed the centre of that object is what is set to the hands location and affected by rotation. The flight stick needs to stay where it is while rotating around simply done by constantly setting its transform to a certain position so if you grabbed it would only rotate the object. When you grab an object it will try and set the objects centre to the hands centre, the issue is when the joystick is grabbed at the top the rotation will not match the hands due to the offset.

This was the main issue of the joystick development as the hand was never on the joystick and would not match the players hand movements, if covid-19 had not occurred I would set the joystick to work like a joint on a bearing ball so it rotates correctly but still stays in the centre of the object. The collective would use the same interaction system as the joystick but with different rotation clamps. The switches would be grabbed by the player and flicked on or off. Apart from this it would be carrying on sorting out the flight settings and VR interaction ensuring for a smooth experience.

There was a smaller issue with the flight physics in virtual reality, often the helicopter when moving would suddenly sky rocket up or plummet to the ground, I also had the issue with rotating the helicopter and it would not move in the direction the helicopter was facing like it would do in real life. To fix this with more time I would ensure that the helicopter would be more effected by gravity and its weight would better pull it down to earth the way it would in real life.

Upon reflection I should of switched the order I worked in, Unreal 4s VR plugin is different from Unity’s VR plugin so if I was to do this project again I would work on the VR interaction first allowing more time for errors and having a fully interactable cockpit first then having the flight systems. I would also work on Unity for a prototype next time, Learning how to use Unreal 4 while creating the project was a mistake as I was stuck with issues that I had no knowledge of how to fix and this greatly delayed the development of the project. Unity is also easier to use with the systems it has and is more fit for this type of project as I can use prefabs and components to better set up the Apache and set up the project.

Overall, with more development and better foresight of issues to come the project could be finished and create a successful project to fly a helicopter around, I would focus on the interaction side first in the development and ensure that the player could interact with the helicopter. I would also ensure to keep better time on development and spend less time on non-vr features.

# Bibliography

Brown, D. (2018, Febuary 27). *We climbed into an Apache helicopter's cockpit and saw why it's one of the most difficult aircraft to fly/Business Insider*. Retrieved from Business Insider: https://www.businessinsider.com/apache-helicopter-photos-tour-2018-2?r=US&IR=T

glyndavies02. (2007, August 4). *glyndavies02Follow*. Retrieved from flickr: https://www.flickr.com/photos/glyndavies/1009200258

Maksel, R. (2012, Febuary 3). *Is This the Hardest of All Aircraft to Fly*. Retrieved from Air Space Mag: https://www.airspacemag.com/daily-planet/hardest-to-fly-87132849/

Marheine, C., Voelter, M., & Grot, A. (2018, November 18). *297 – Flying the AH-64 Apache*. Retrieved from Omega Tau: https://omegataupodcast.net/297-flying-the-ah-64-apache/