The making of
Full Ahead!

Anna Catton FYP

20013485

Contents

Development Methodology

An agile development methodology was initially planned, in two phases.

Phase 1

Create prototypes for various aspects of the game, including the mobile input and tank mechanics, creating a base for continued development – researching each aspect as it is being developed. This was expected to take until the mid-point review.

Review	Develop	Test
Assess current progress, draw informed conclusions from all information available.	Implement changes to improve the project as informed by previous review.	Both internally and with external testers, play the game and note any issues/ tweaks/ potential improve-ments.

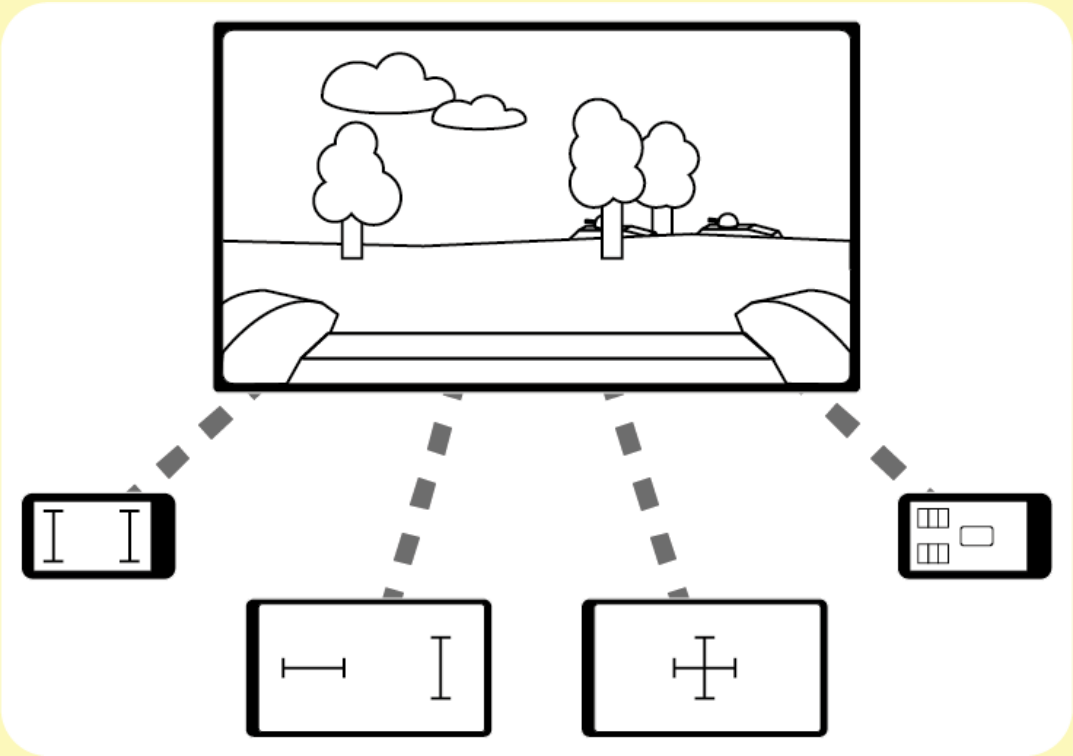
Polish

As the deadline approaches, stop the development cycle and focus instead on polish and evidence.

Full Ahead!

Phase 1 Proposal

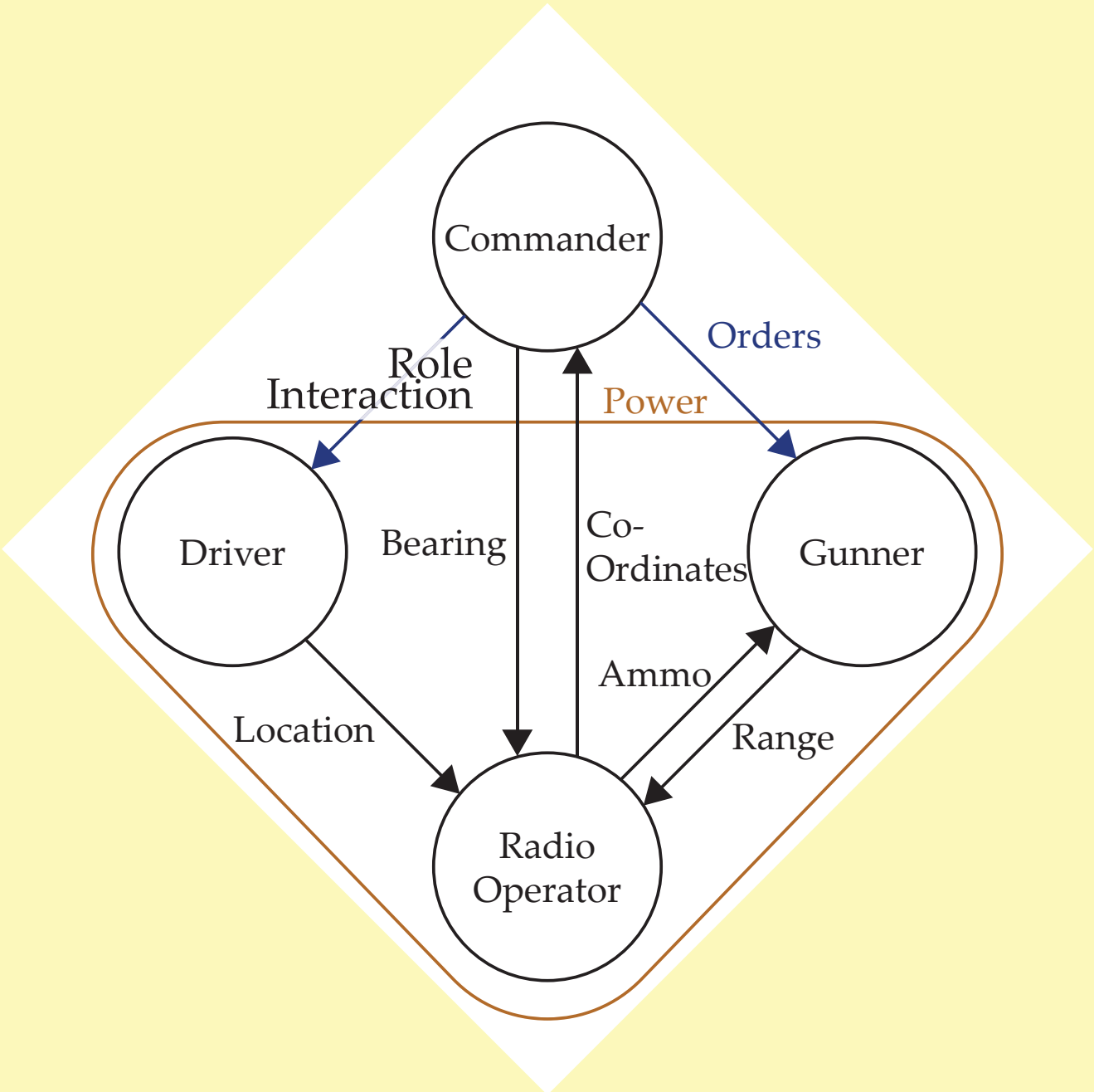
The initial ‘Full Ahead’ proposal was for a co-op multiplayer video game prototype in which up to four players cooperate to collectively operate a tank.



The gameplay would take place on a shared monitor which would show the outcome of the player actions and on an additional mobile device each which the players would use for input.

Phase 1 Proposal

The four roles below must work together as an effective team, and must have this teamwork tested - not just four players playing four separate minigames.



To this end, research was focused first upon multiplayer games of this type, to inform the design.

Phase 1 Research

Spaceteam. 2012. Android [Game]. Sleeping Beast Studios: Canada.

Star Trek: Bridge Crew. 2017. Windows [Game]. Ubisoft: Montreal.

Wolfpack. 2019. Windows [Game]. SUBSIM: Online.

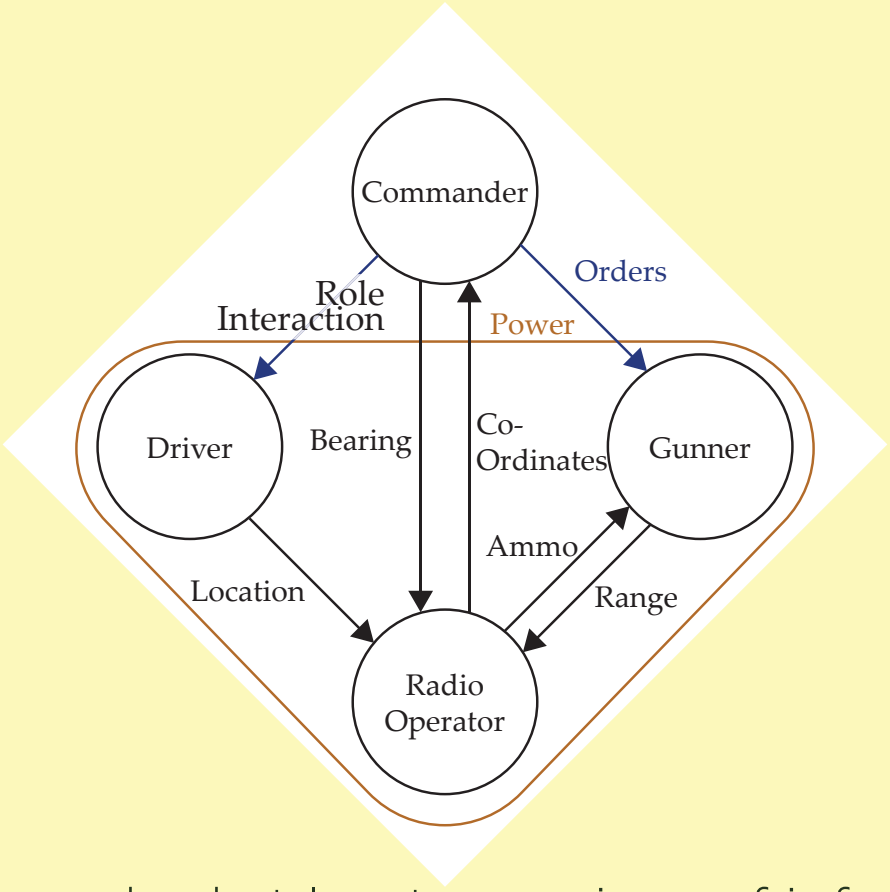
The initial idea for a co-operative game of this sort where each player had a single role was similar to several other games that were found during my research. The mechanics and gameplay of ‘Wolfpack’, ‘Star Trek: Bridge Crew’, and ‘Spaceteam’ were examined primarily for inspiration at this stage.

The primary takeaways were the separation of not just role but information – players had different information that they would have to share, and any communication barriers here tended to enhance gameplay due to the feeling of co-operation.

Using this, the roles of Commander, Driver, Gunner, and Radio Operator were prototyped to each need and share some resources or information.

Phase 1 Development

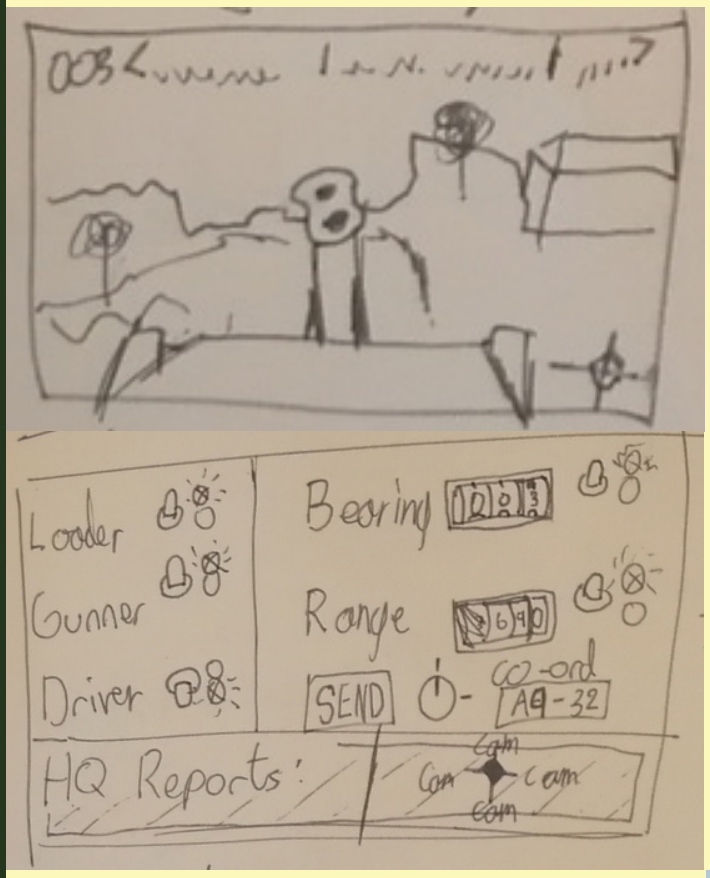
The primary design had each role have both an ‘output’ screen where they can see the results of their inputs, which could be seen on the main monitor, and an ‘input’ screen on their mobile devices. - Their inputs are always available but the results of those inputs require co-ordination to understand. The goal was to force the team to work effectively as a group or else have them become less effective in combat.



Each player had at least one piece of information available to only them but needed by another, and the additional interactions of the Driver, Gunner, and RO needing to share power and all players needing to co-ordinate who was able to see their output screen added to the concept’s testing of the group co-operation.

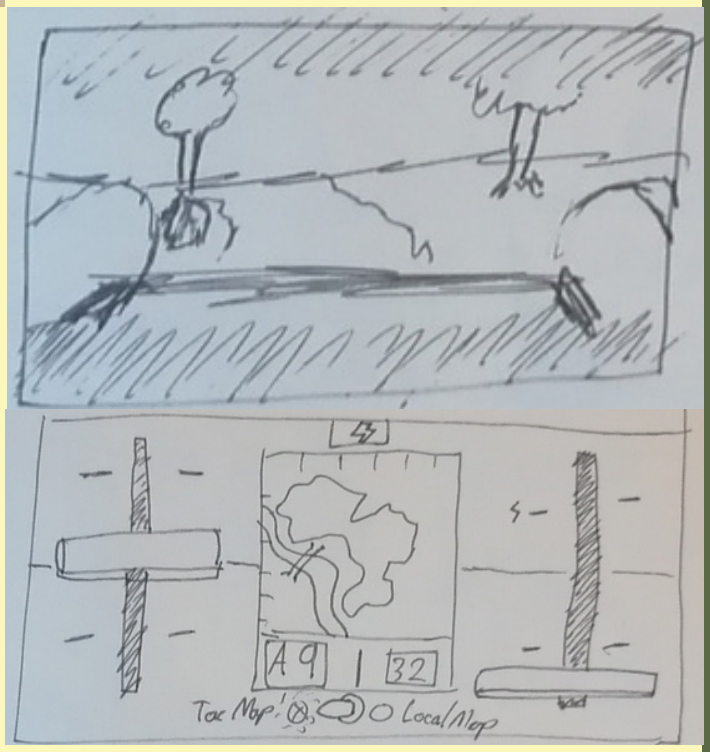
Full Ahead!

Phase 1 Development

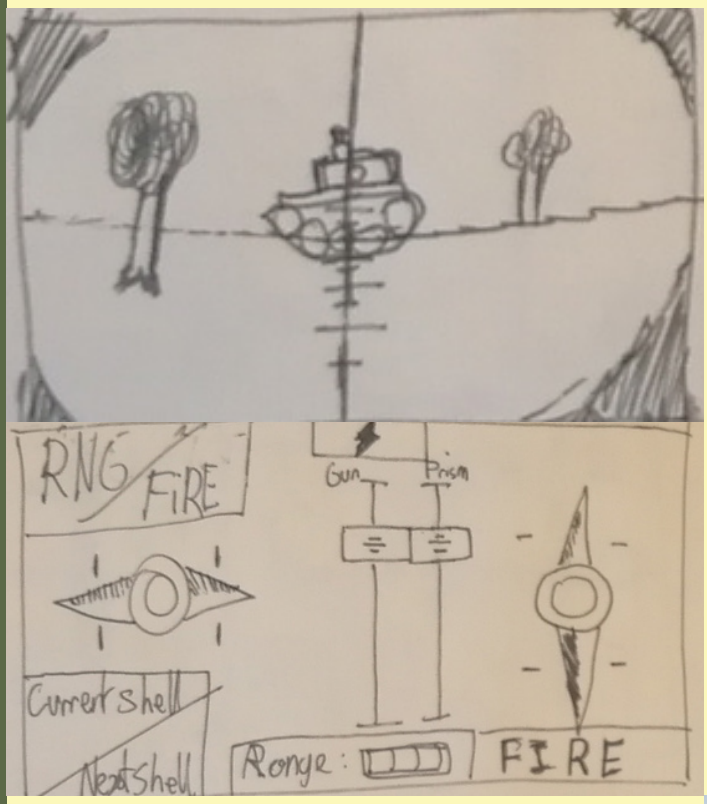


The commander is the heart of the operation.
Their job is to co-ordinate the rest of the players to achieve the tank's objectives.
They are the only player with the power to control the main view, on the large screen.
They are the only player with information as to the tank's compass bearing.

The driver is in charge of getting the tank where it needs to go, and knowing where exactly that is.
They have access to tank driving controls and the map view.
They are the only player with the tank's current grid co-ordinates.

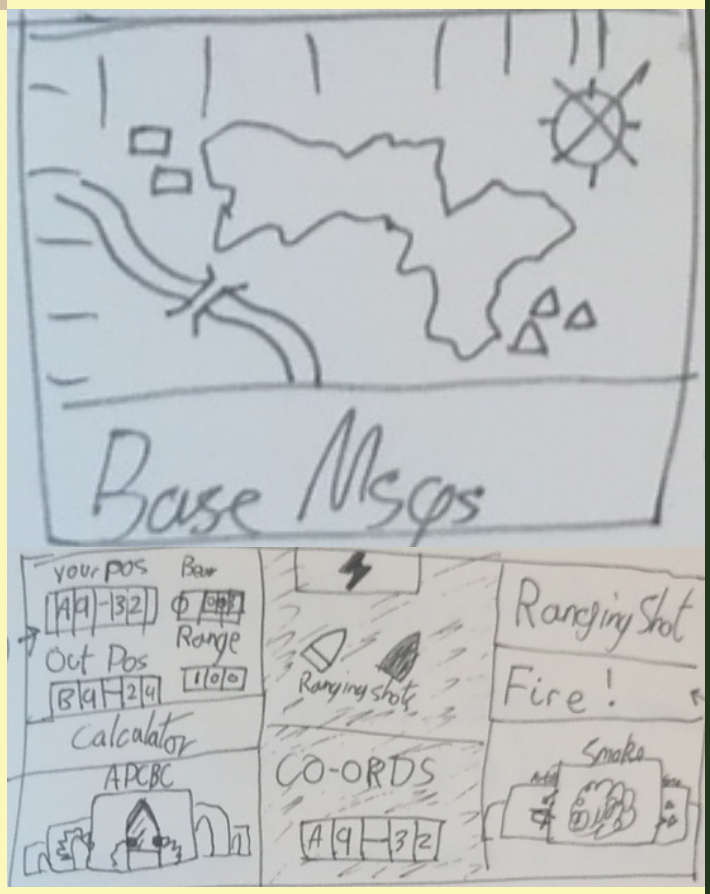


Phase 1 Development



The gunner controls the operation of the tank's turret and weapon. They can aim and fire the gun, but have limited vision. They are the only player who can find the range to a given target, using their rangefinder.

The loader/radio operator is in charge of calling in artillery, and ensuring the best ammo type is loaded.
They have a distance computer with which they can calculate co-ordinates, the ability to change ammo types, and the ability to call in support such as artillery using the radio.



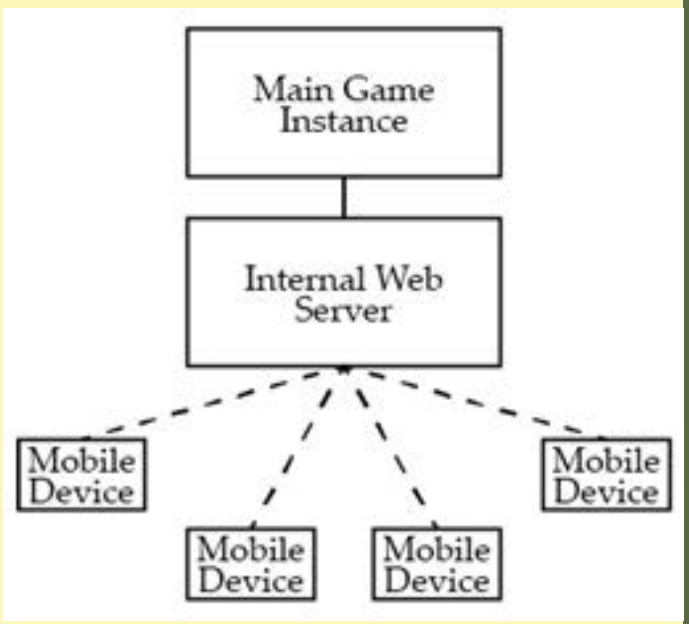
Full Ahead!

Phase 1 Technology Tests

The technology behind the mobile input was a stage that a long time was spent on. Inspired by the 'Jackbox Games' series, work on concepts and prototypes for how such input would work was begun.

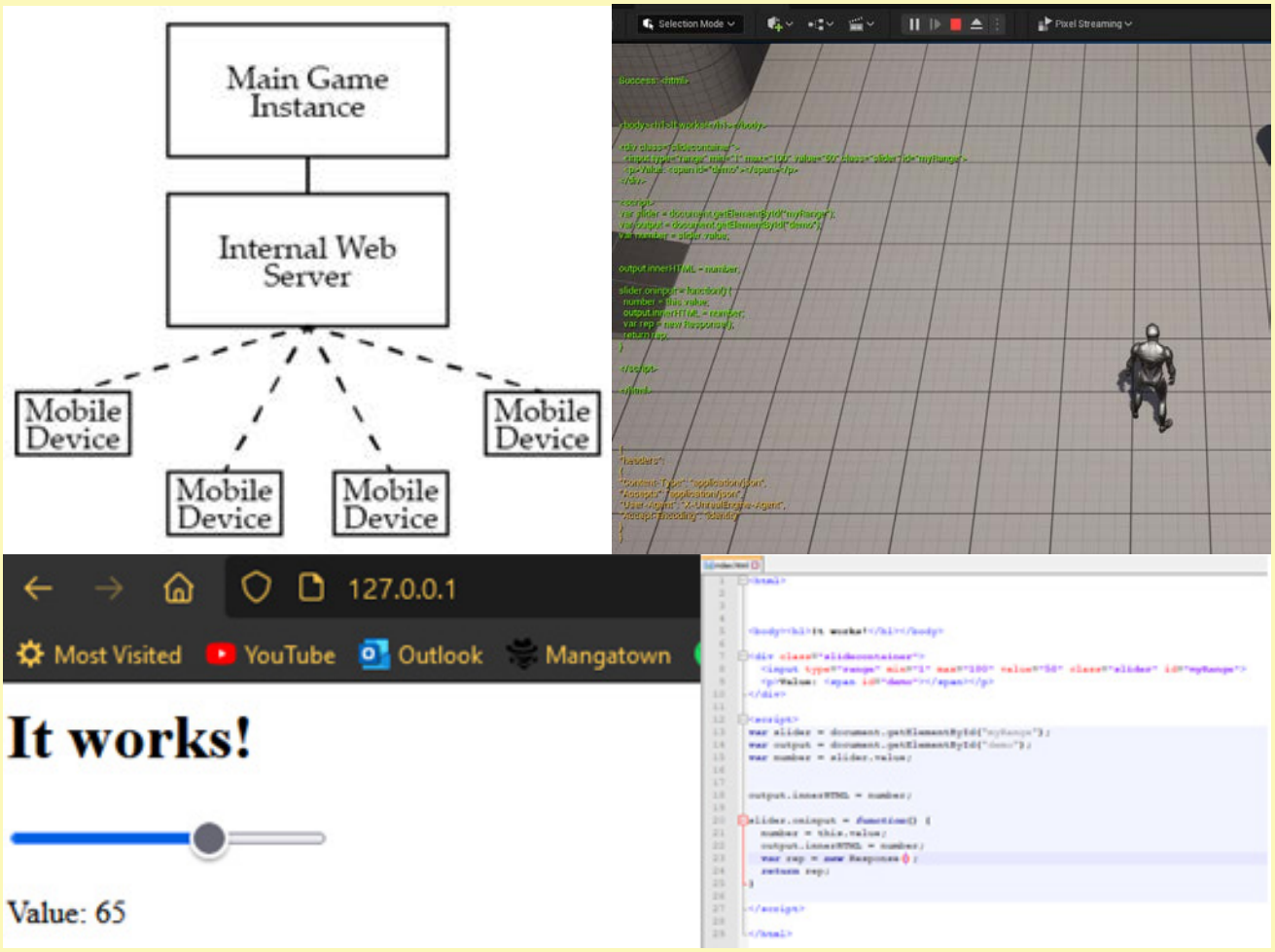
The goal was to create a system that would be system-agnostic and require as little setup as possible. This proved to be far more difficult than was expected.

The Jackbox Party Pack 3. 2016. Windows [Game]. Jackbox Games: Chicago.



Due to both familiarity and ease of use, all in-game development would happen now in Unreal Engine 5.2.

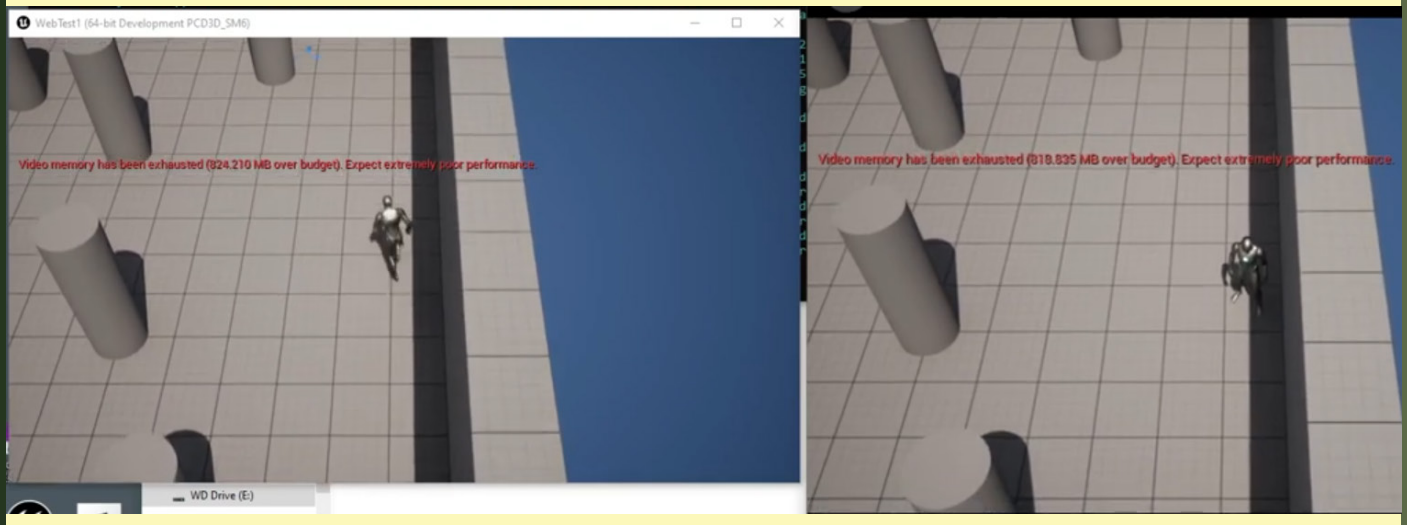
Phase 1 Technology Tests



it was then attempted to create an internal server which would then show a dynamic webpage from which the players would be able to input. This would mean there would be no setup beyond connecting to an IP address through a device's browser. Although making an internal server and fetching data from that into the Unreal Instance was possible, creating a responsive system without adding another layer for the server systems proved impossible, and this stage of development was taking a long time and beyond the expertise of the developer.

Phase 1 Technology Tests

A version using Unreal's 'Pixel Streaming' tools was also attempted, which – although successful internally – was incredibly delayed even when streaming to the same PC's web browser, and thus unsuitable for input/output purposes.
This issue became one that would wait until mid-point review.

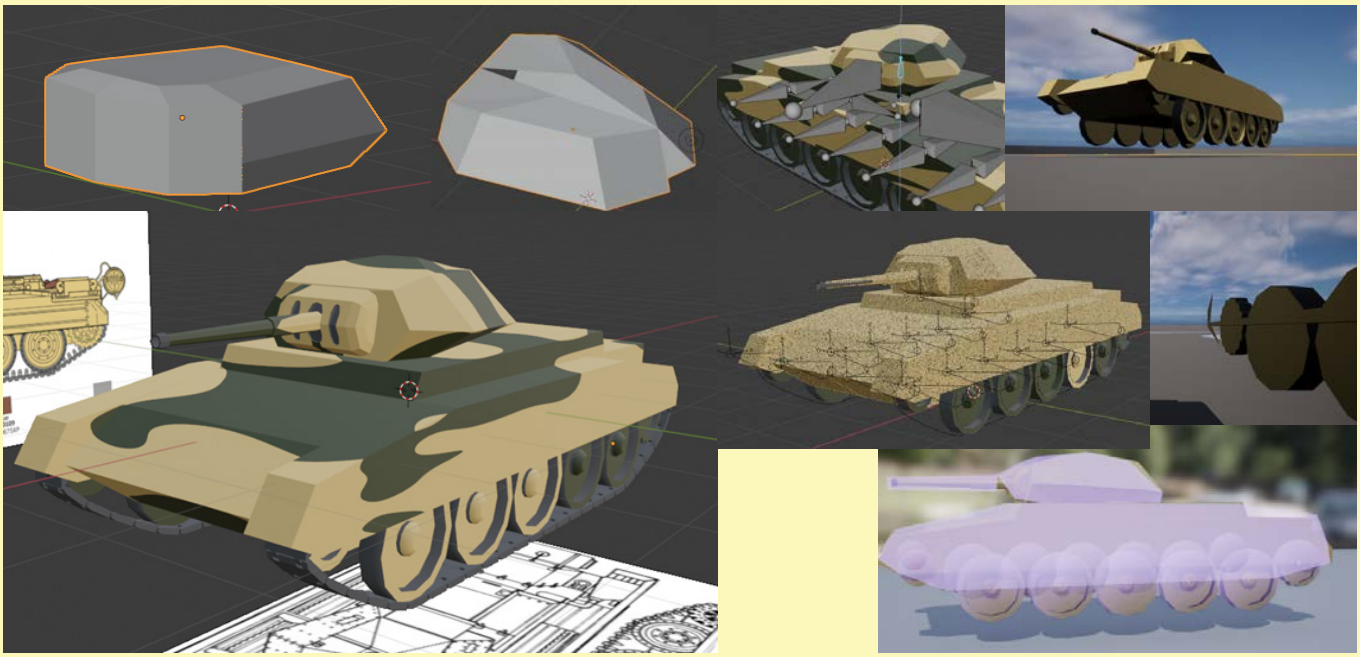


Phase 1 Asset Creation

Despite not being an art project it was decided that the tank asset itself should be created such that it could be edited to the needs of development without hassle or issues of copyright. An appropriate tank was selected, and from reference images modelled in Blender.

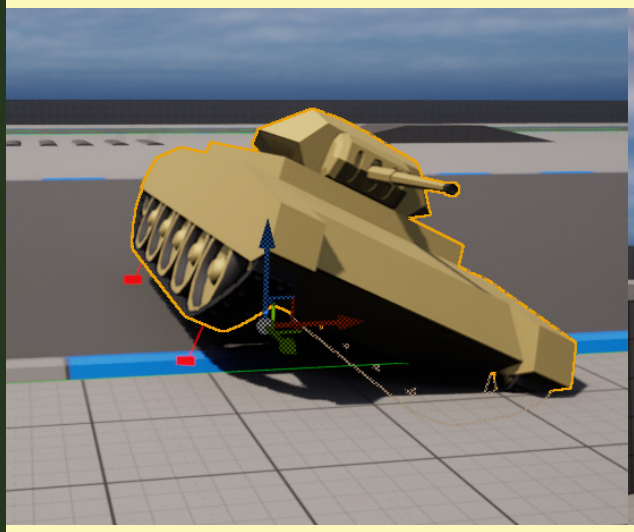
Catton, A.J. (2021) *Assorted Tank Images* [Photographs]. Unpublished.

The results were satisfactory, and the creative control would prove itself useful during development several times. Due to a lack of experience with rigging and animation, the ‘UE4 Vehicle Rigging Addon for Blender’ tool was used to rig the skeleton.



Full Ahead!

Phase 1 Technology Cont.

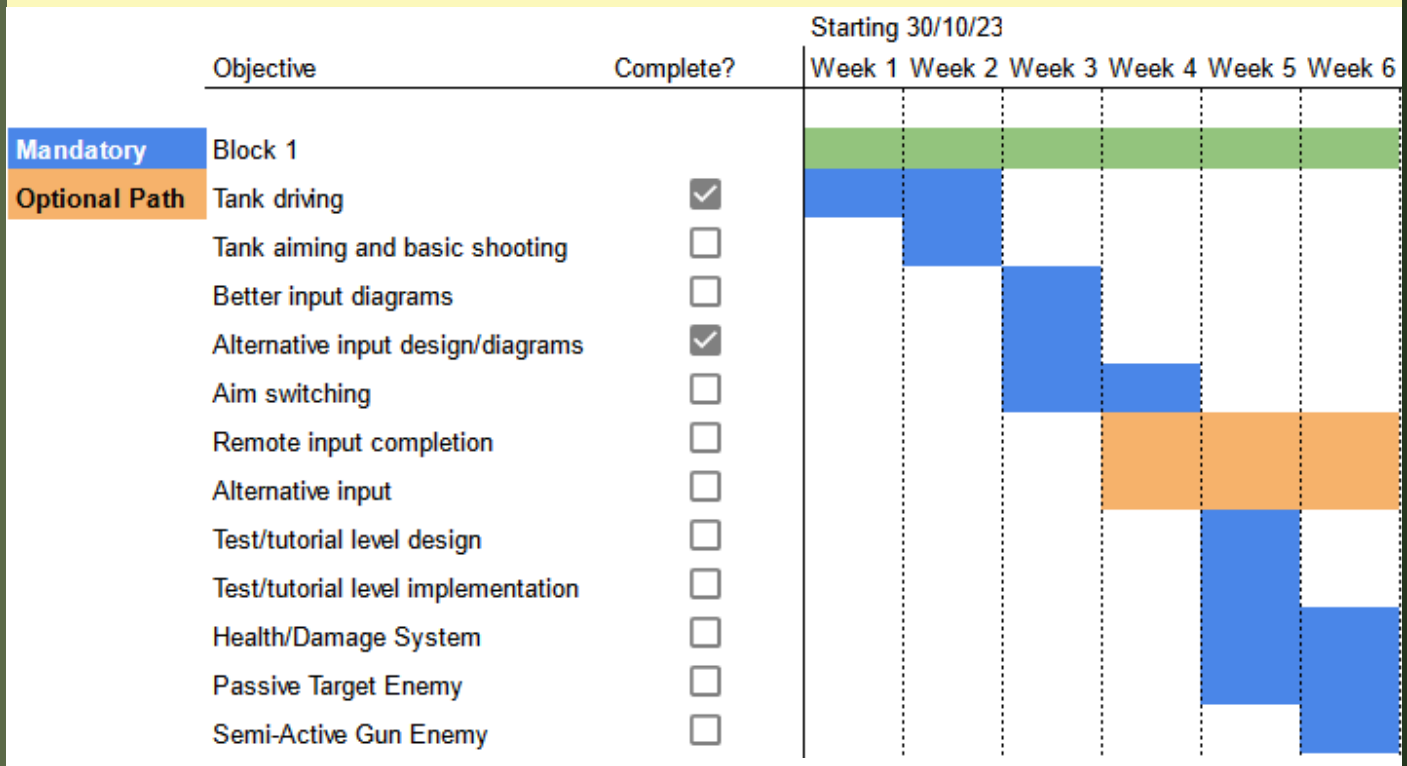


Creating the tank movement would also pose more of an issue than was expected. The default vehicle classes for Unreal did not support vehicles with many wheels, so it was attempted to create this movement manually – as it didn't need to feel realistic, but rather predictable. This would turn out to have too many issues to get the desired results.

The inbuilt ChaosWheeledVehicle component was much more complex, giving more options but also creating many aspects that would have to be disabled or else set to basic settings such that the tank would not be complex to control but rather act more 'arcadey'.

This took a lot of work but eventually it was gotten working, and the addition of more realistic suspension and traction turned out to be a major positive to the final gameplay.

Mid-Point Review



3D	Category:	Asset:	Source:	Status:	Notes:
	Pawns	Tank Model	Create	In Progress	
		Enemy Tank Model	Create	In Progress	Could use friendly tank but palette swap
		Static Enemy Model	Create	Not Started	Field gun?
	Landscape (Forest)	Trees (various)	External Source	Finished	Likely won't have both forest and desert
		Bushes (various)	External Source	Finished	'Stylised Forest' pack
		Rocks (various)	External Source	Finished	
		Grass/Ground mats	External Source	Finished	
	Landscape (Desert)	Rocks (various)	External Source	Finished	Likely won't have both forest and desert
		Foliage (various)	External Source	Finished	'Stylised Desert' and 'Stylised Egypt' packs
		Ground mats	External Source	Finished	
	Buildings	Buildings (various)	External Source	In Progress	Buildings available in 'Stylised Desert' pack could fit

Mid-point review showed that the project was severely behind on the expected progress at this point.

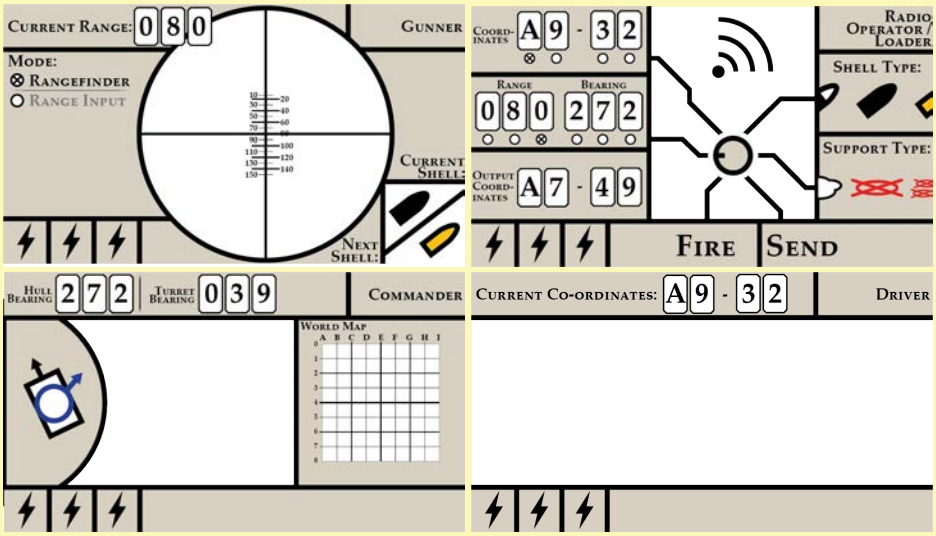
Networked multiplayer was cut to a stretch goal, and work was begun on redesigning and developing the project such that what was created could carry forwards, without the aspects of the project that were slowing it down.

At this point development phase two began.

Full Ahead!

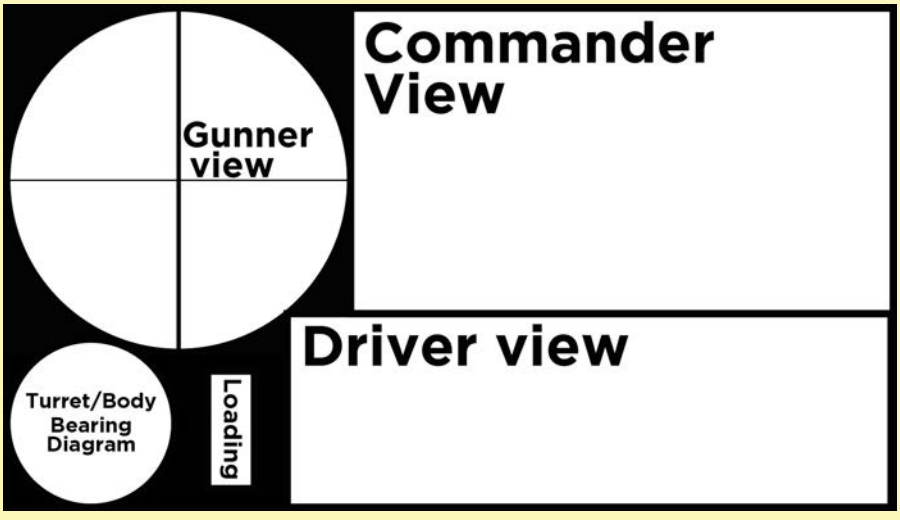
Phase 2 Redesign

The first redesigns, cleaned up.



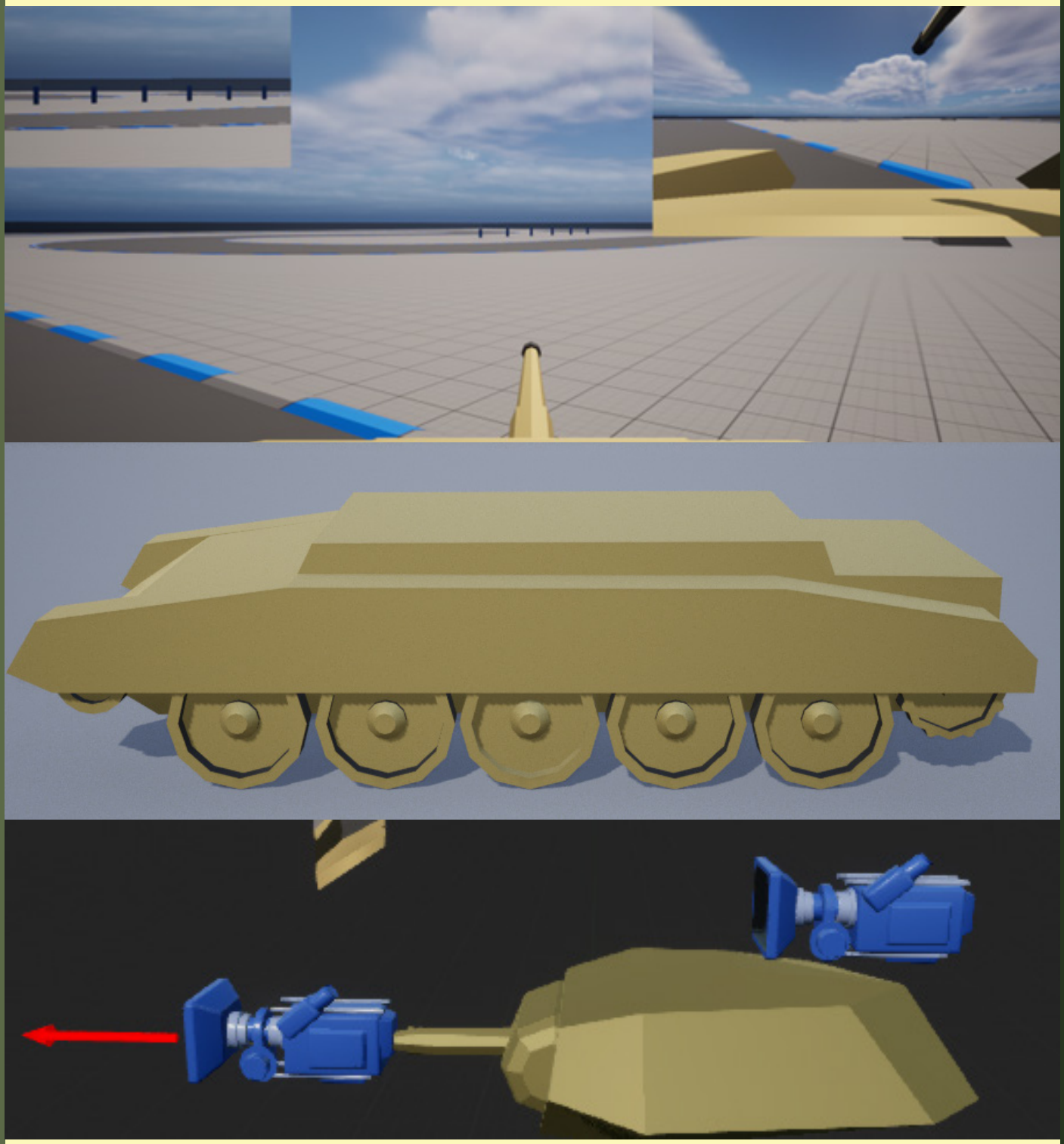
Under these new constraints the player interactions had to be redesigned. Initially, looking at keeping their roles the same and creating UI for them to replace the mobile screen was considered.

This could have been implemented but worrying about over-scoping again and creating something overly complex without being able to properly tutorialized players informed the decision to instead consider a more simplified option – two players, a gunner and driver, each sharing a single screen showing their state, with some extra information available.



The final redesign.

Phase 2 Gameplay Dev.

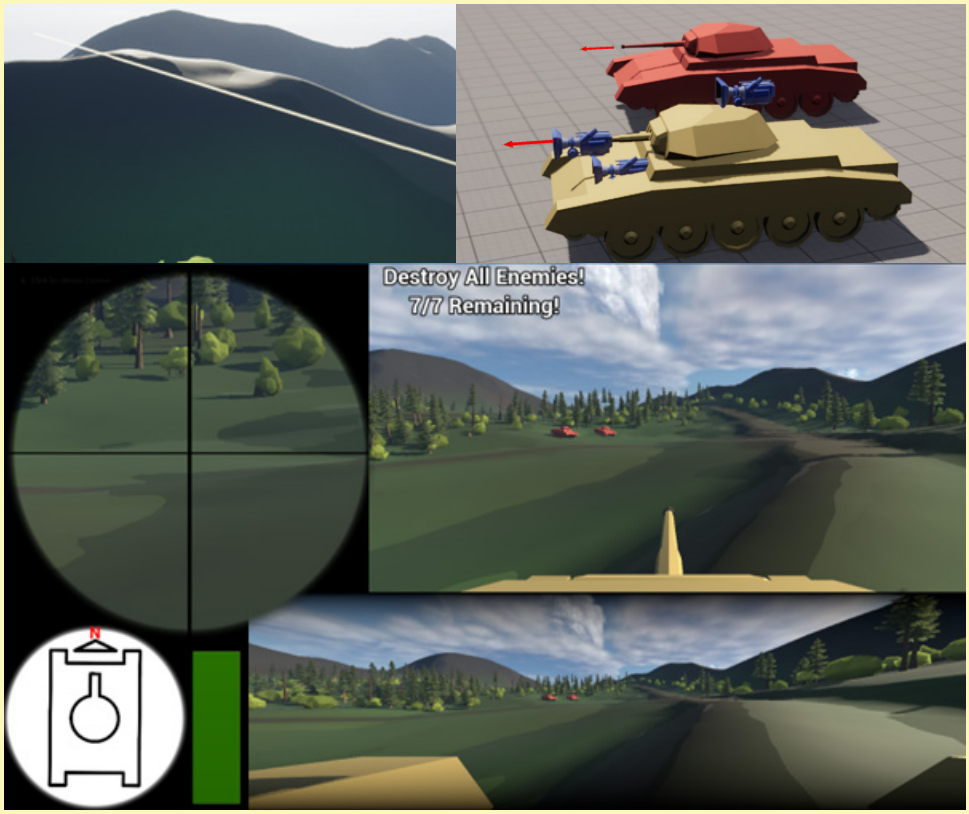


A basic setup for such a UI was created - after some tweaking of the tank model to allow it - and these two parts would be separate actors to allow work on them to not interfere with one another and to save the hassle of multiple players controlling a single pawn.

Full Ahead!

Phase 2 Testing

Testing externally revealed several small issues and improvements. Most notably, players noted they were unable to bring the vehicle to a stop using the driver's unconventional control scheme, that there was little feedback for the gunner, and players would frequently not realise a dead enemy was dead, and continue firing.

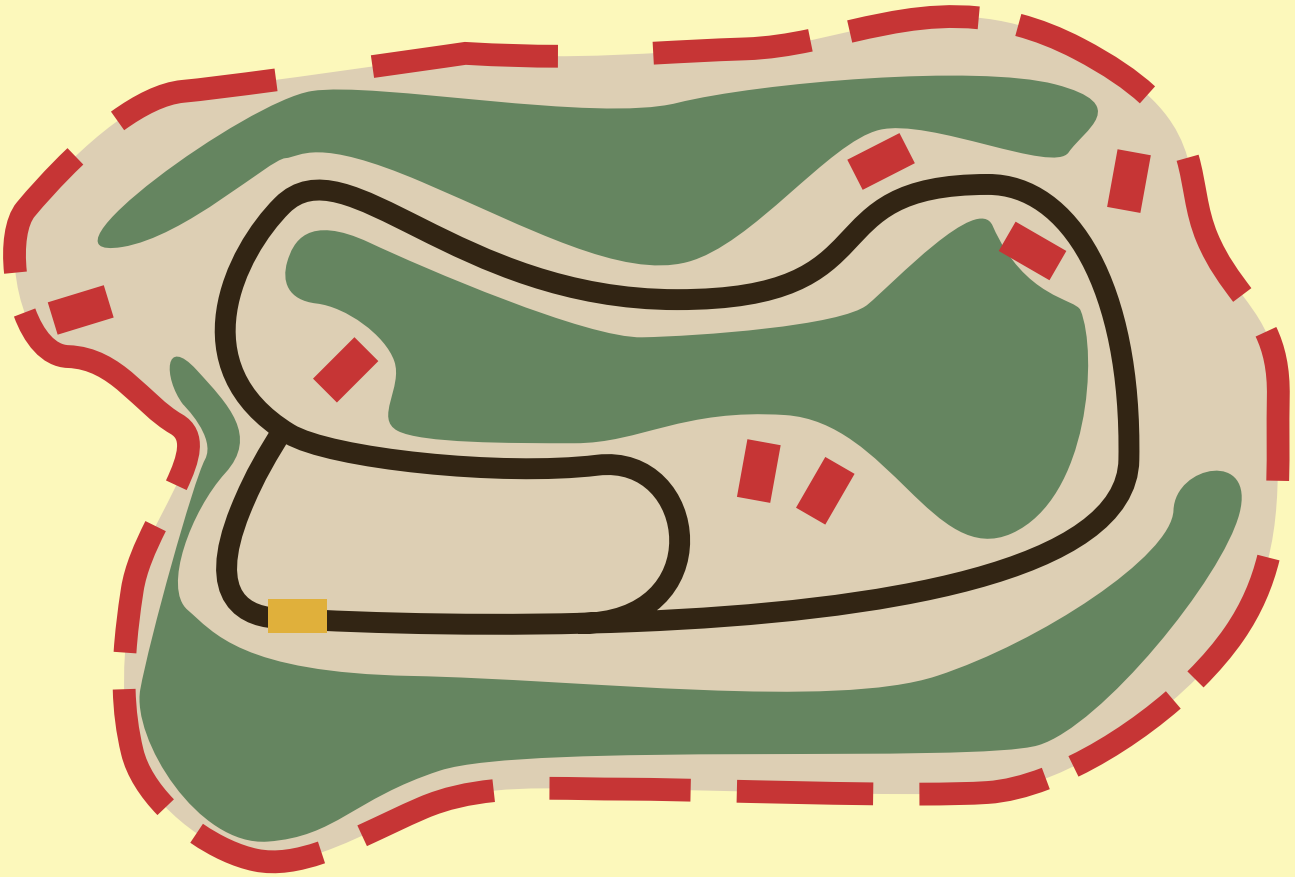


Creating several aspects of feedback for both players (including a muzzle flash, shell trail, sound effects, more vehicle destruction effects, and controller rumble.) served to fix many of these feedback issues.

The UI was created as designed in-engine, a handbrake implemented for the driver, and additional particle effects added to dead vehicles, which now were coloured red to differentiate them.

Phase 2 Level Design

The level was designed fairly quickly using asset packs for the necessary models, to create an environment to further test within.
Tanks were used as ‘breadcrumbs’ to guide the player along the looping path, and enemies early on within vision were included to teach the players to shoot without requiring movement, and later enemies hidden from both the tank hull and turret require the players to co-ordinate and use all three views available to them to work effectively.

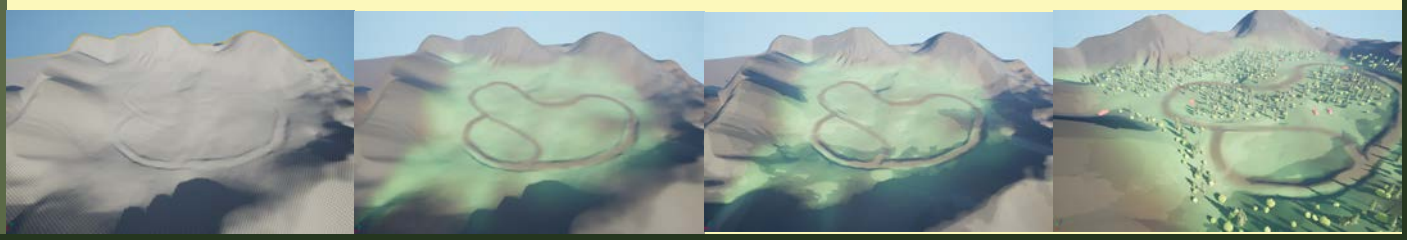


Difficult terrain

Intended path

Enemy/Friendly tank spawns

Level boundaries



Full Ahead!

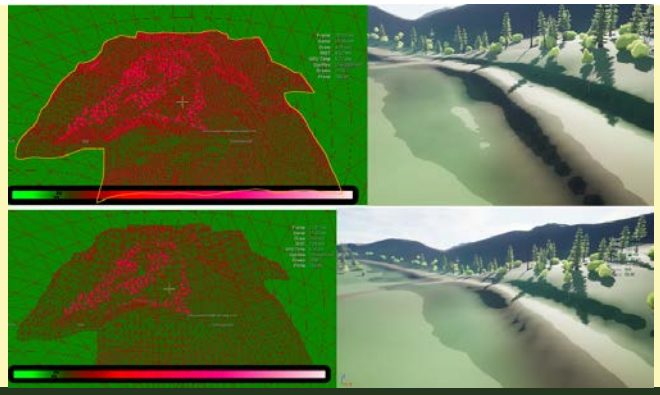
Phase 2 Optimization

At this point, built versions of the project were running particularly poorly on the machine that was being used to develop them. Looking into optimisations issues were found within the ground textures, resolutions, and overall lighting setup, amongst other things.

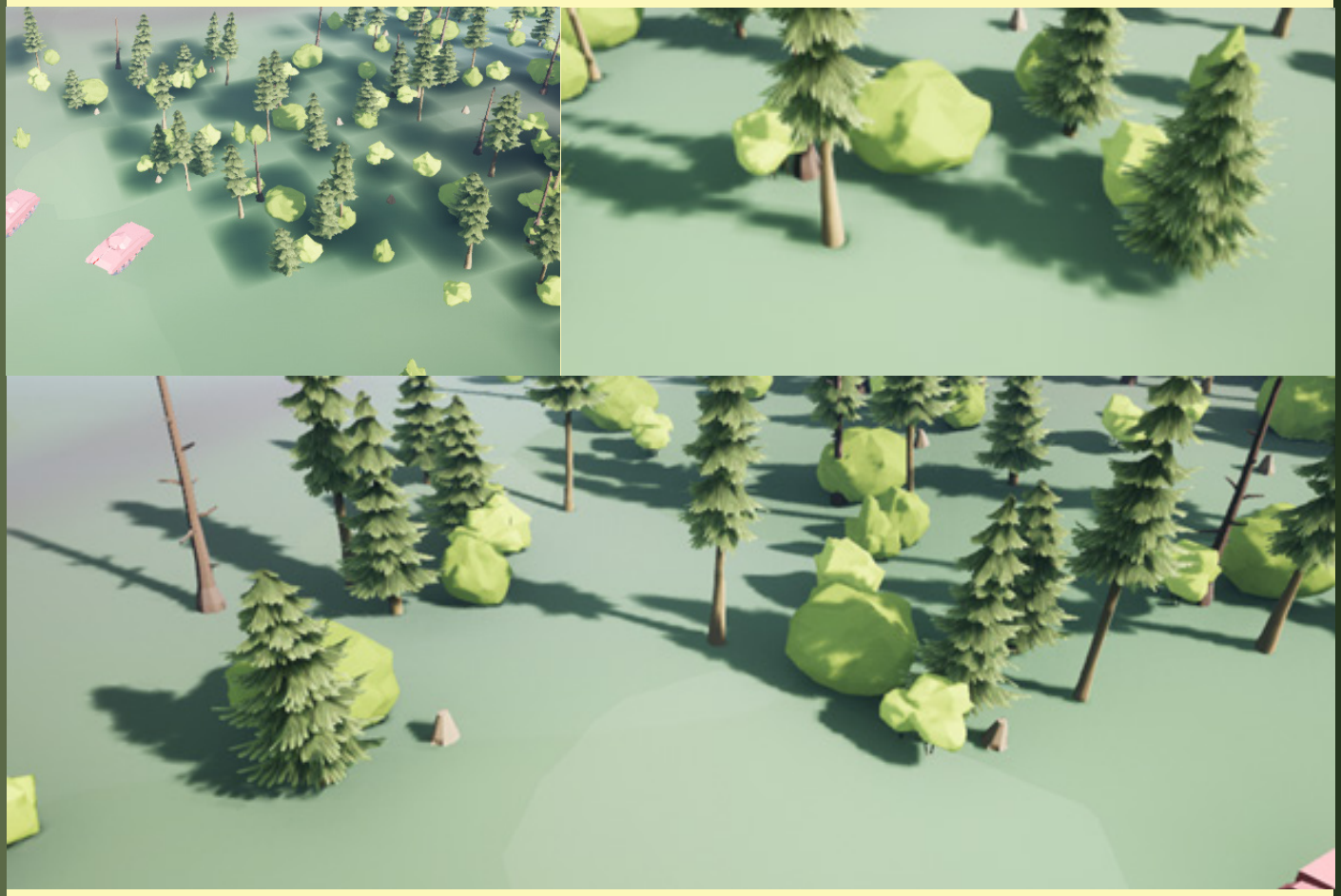
^ Fast noise

^ Voronoi noise

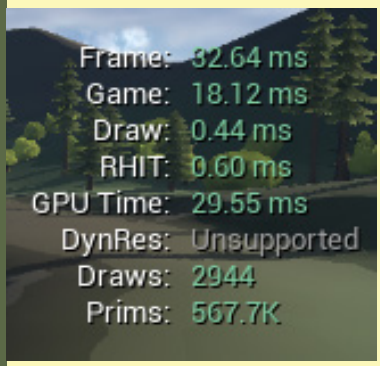
^ Constant scale



Phase 2 Optimization



Lighting particularly was a large area of improvement - rendering four times meant each improvement would have four times the effect.



After substantial tweaks, this ran far better, and posed no further issue.

Phase 2 Testing Cont.

After further informal testing, players still were not recognising burnt-out tanks, and many were unwilling to use the tank aiming controls at anything less than full input.

A newly coloured texture for dead tanks, plus an aiming noise which would scale it's pitch with input guided the players to solve these problems, and improvements were seen.

Phase 2 UI

UI development was short but iterative, creating first simple versions that merely held the necessary controls, and later being updated to a unified style.

Full Ahead!

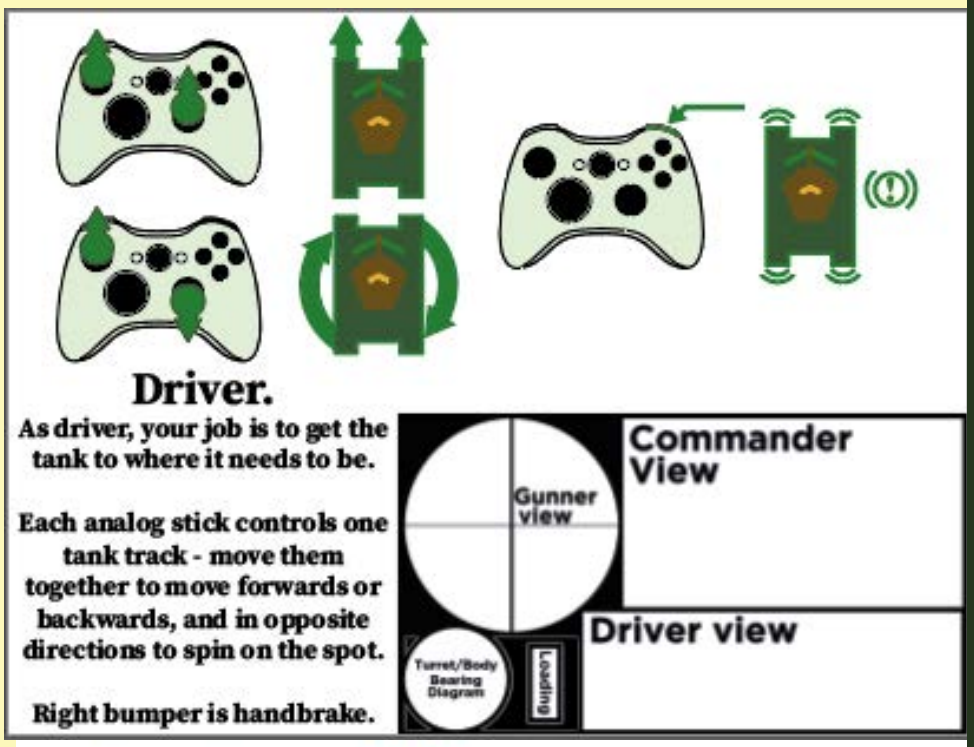
Phase 2 UI

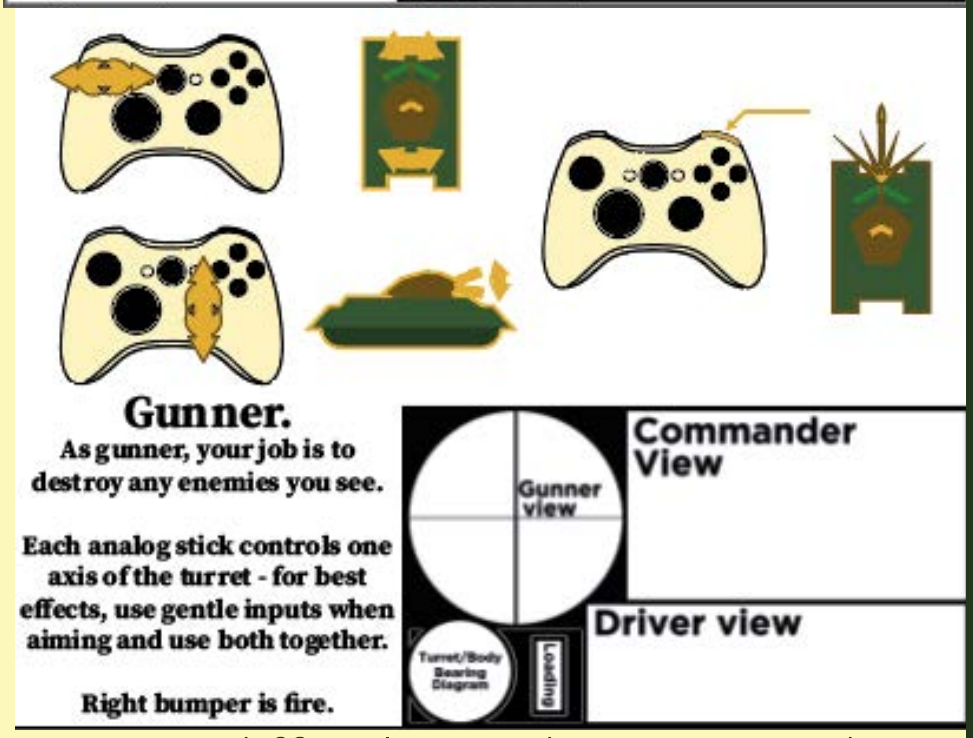
One issue that frequently appeared in playtesting was needing to explain the controls to players in playtesting, or difficulties explaining the specifics. As planned from the start, the UI included a control scheme that explained the basics on the start menu, as well as a more in-depth control scheme for each player in the pause menu.

These panels can be shown/hidden on their respective button presses, and the pause menu control schemes show only information relevant to the player who has paused the game - as shown by the UI highlight colour.

Phase 2 UI

These more in-depth diagrams proved helpful in explaining the control schemes to testers.

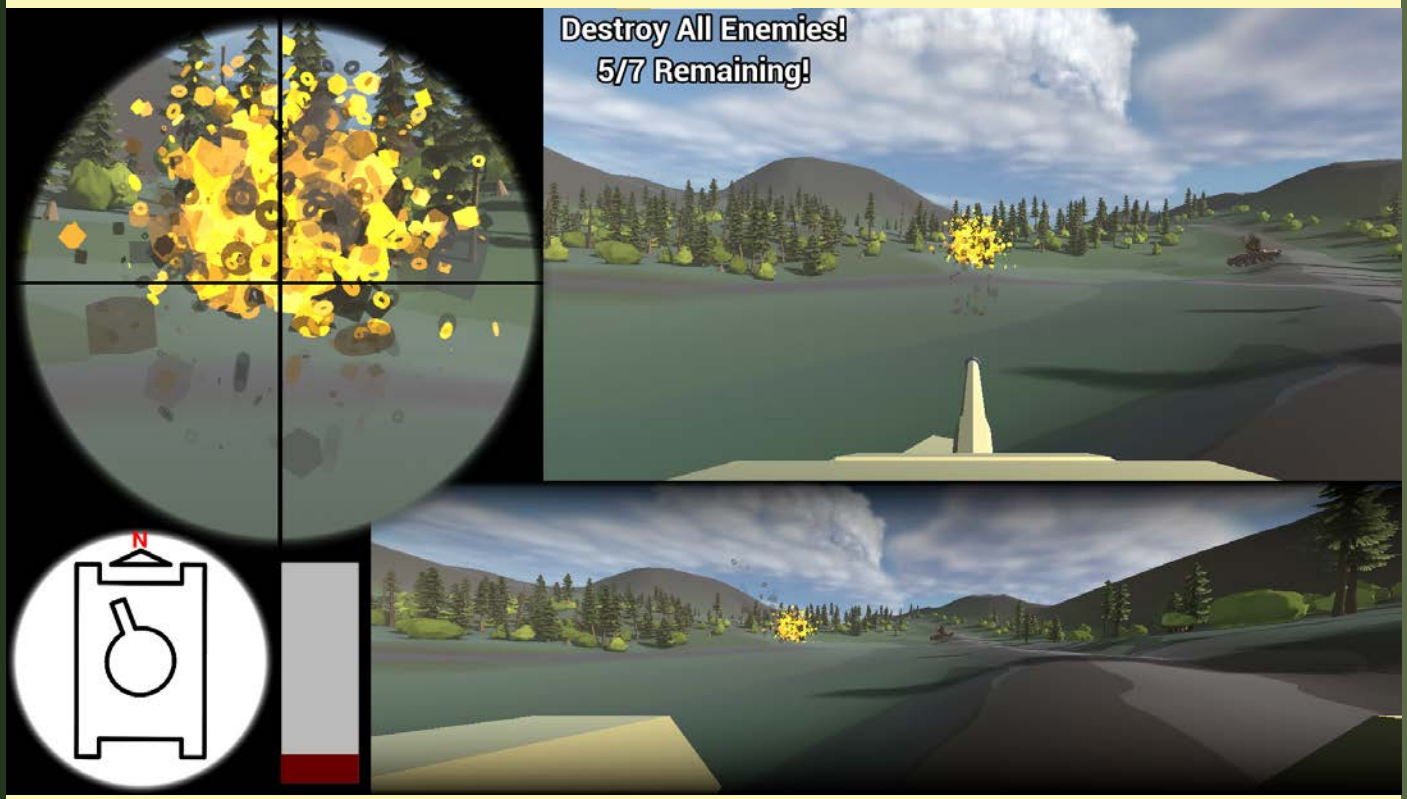




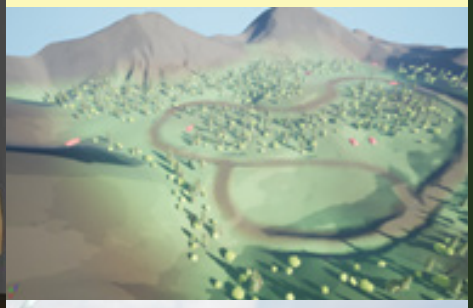
There was some extra difficulty with creating the UI when attempting to make the options highlight as desired. Unreal's built in button highlight only works with mouse events, not controller inputs. The end solution to this was to move the invisible mouse to the center of each button, rather than try to entirely redesign Unreal's UI interaction systems.

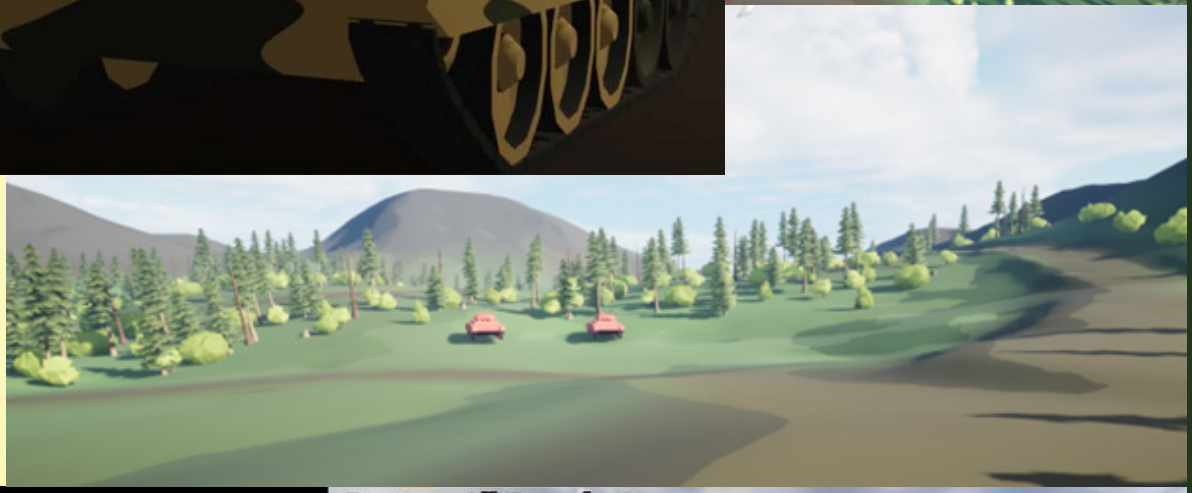
Full Ahead!

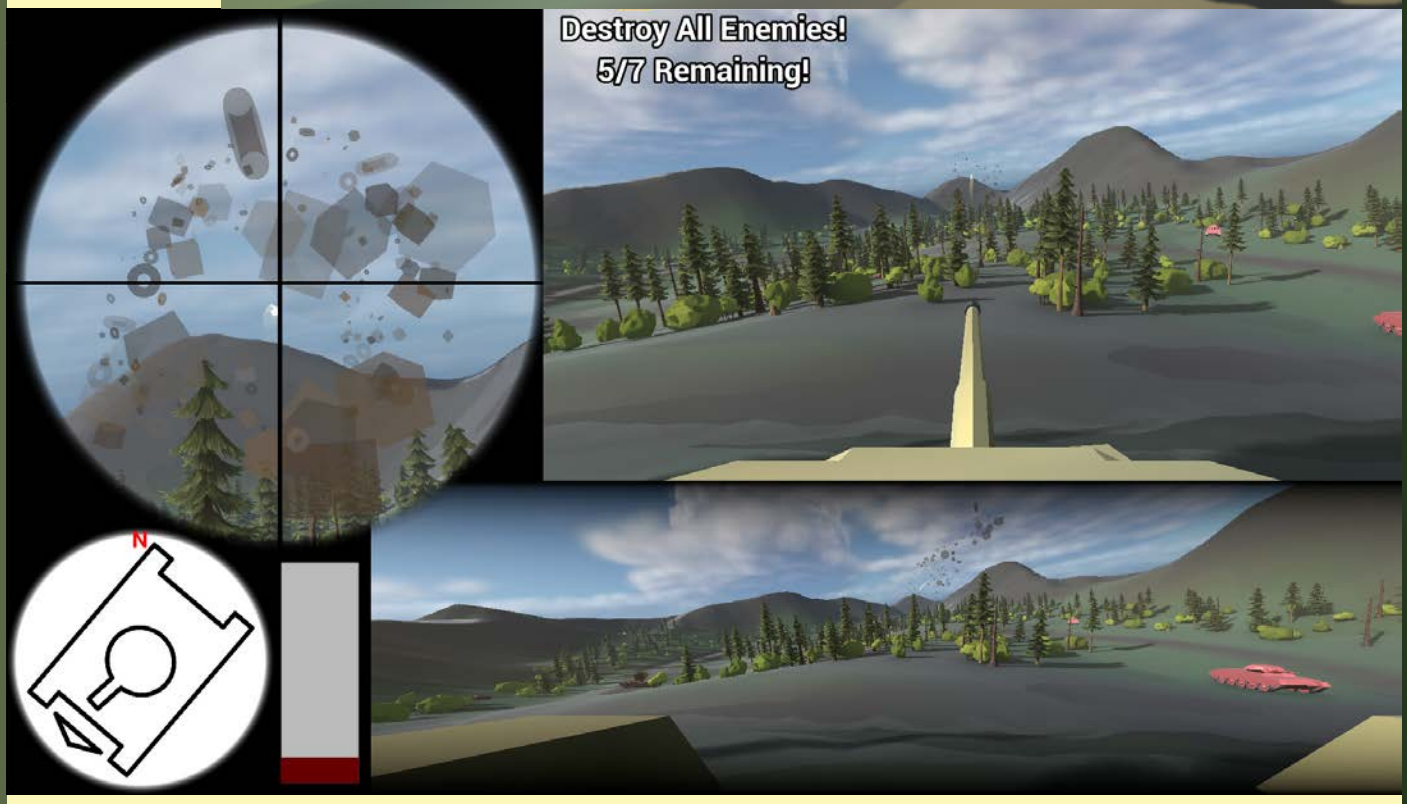
Final Outcome Showcase



Final Outcome Showcase







Full Ahead!

Final Process Reflection

If this project were to be started over from the beginning, the initial, overcomplicated idea which was beyond the scope of a single inexperienced developer would be discarded, outsourced, or else severely trimmed.
Instead, more time would have been spent upon the core systems early on, solidifying the concept into a strong core that could be further polished and developed upon.

With this extra development time gained from underscoping and expanding instead of overscoping, refactoring, and redesigning, many additional aspects could be fully explored.
In the future, the prototype may be expanded to include aspects of PvE/PvP combat, including AI or networked multiplayer functionality. Although AI was at the time categorised as a stretch goal for the project, and as such not something that fell behind, it would enhance the experience greatly.

Ultimately, the setbacks of this project were a combination of inexperience, overconfidence, and several unavoidable issues outside of development limiting the time that could be spent developing core systems early on.
Taking this experience forward, the primary lesson will be to focus on a small, simple core system with room for growth, rather than a complex system that must be refactored.

Final Project Reflection

The basic prototype for the core mechanics of a co-operative multiplayer game in which two players collectively drive a single vehicle has been created.

Although not in keeping with the original proposal, the redesign as created at mid-point review is adhered to, and polished to a degree strong enough that playtesters have had no apparent difficulties with understanding the gameplay or visual aspects of the prototype.

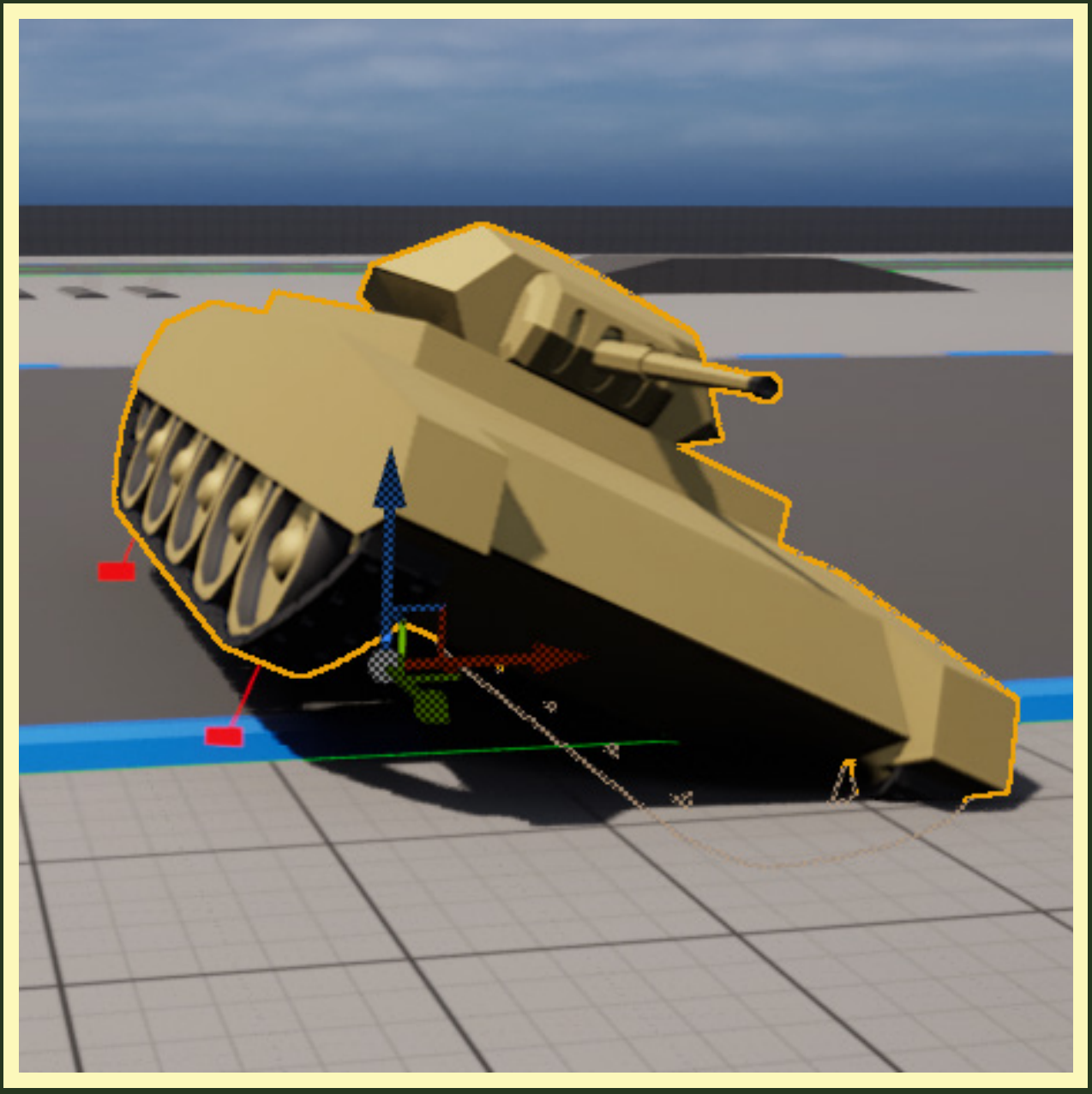
The core mechanics are strong, and enjoyable to play with, both in racing against the clock as in the level that was created, as well as without a goal - just driving around and exploding things for fun.

The art style, while not complex or final, is clear and coherent - both simple to visually understand and consistent throughout. This gives the prototype a level of polish that makes it feel professional.

The assets created are simple but effective, and could easily be polished to meet the rest of the low-poly aesthetic with extra detail, or else replaced by an experienced art team as part of full development, were this developed past prototype.

Ultimately, it is felt that the prototype delivers on the primary goal - the mechanics are fun and could be developed upon into a full game with the knowledge gained here.

Full Ahead!



Anna Catton FYP - 20013485