Deep Space:

Setting: The year is 2666, human kind has conquered the stars building an empire which spans over a few galaxies. While the war between the human empire and the federation comprised of a number of plant based lifeforms is a few million light years from the peaceful research facility on Zetra V. Or so it was assumed until what we currently assume is an artificial planet entered the solar system throwing the entire gravitational balance of the system into disarray. The scientists on the station calculate that the new course of your planet will lead it into the sun is little more than 48 hours. The evacuation of the planet begins and the first emergency shuttles take of the planet alien fighter ships attack killing hundreds in a flash. As a senior elite ship pilot you step in to the simulation room from there you will control your remotely your vessel. It is up to you to destroy the enemy, evacuate the facility and warn the empire of this new threat.

Description: Deep Space is a MOBA style spaceflight simulator where you pilot a spaceship in pursuit of destroying the enemy base while keeping your own base safe. every interval of time each side spawns a few fighter ships that travel the distance to the other teams base, attacking any enemies they meet on sight. Since each teams base is protected by a cannon for self defense a single ship would hardly have time to damage it so attacking it in large quantities is the only way to dent it. The player plays an elites hip, a faster healthier ship outfitted with four unique abilities which the player can activate easily from the simulation room. There are two additional modes for capture the flag and a loop game where you must go as fast as you can through loops at a certain angle to receive maximum number of points.

Controls: Trigger and Grip are the two basic buttons each VR kit contains, when referring to these buttons in this document we will use these symbols trigger **T** and grip **G**. the third button will be marked **0** and the following are the matching button in each kit:

Vive + Mixed Reality: both hands use the trackpad.

Valve Index: both hands A button.

Oculus: left hand – X button, right hand – A button.

Set up:

First step - Adjusting the button sensors: each button in the game is associated with at least one invisible cube. To use the button the player will place at least one hand inside the cube and press on one of the inputs explained in controls. There’s a cube around the player whose outer limits compromise of these sensor cubes and the player can use the button, which from now on will be referred to as utility, named sensor orb to adjust these cubes to represent the utilities that match his playstyle. The sensor orb utility is a small gray ball, with hand on ball pressing either **T** or **G** will make the sensor cubes glow with colors indicating which Utilities pressing or releasing an input inside of them will activate. The colors are empty – black, ability1 – yellow, ability2 – blue, item1 – red, item2- green, recall – brown, self-destruct – pink, joystick-aim – orange, joystick-fire – emerald, sensor-orb - gray. The ball has 7 different colors each representing the settings the orb currently changing, to switch between them the player will press **T** and **G** in any order to go into color changing mode, different sensor cubes will light up according to which color presides currently. Releasing the **T** button and then pressing on it again while holding the **G** button will iterate through the colors forward, using the same method with the buttons reversed will iterate through the colors backwards. The colors: Gray: no changes, no glowing sensor cubes.

Red: **T** changes distance to left wall sensor cubes, **G** changes distance to right wall sensor cubes, glows middle partition of both walls.

Green: **T** changes distance to front sensor cubes, **G** changes front sensor cubes height., glows front sensor cubes.

Blue: **T** changes size of joystick sensor cubes, **G** changes transparency of all sensor cubes, glows joysticks and sensor orb.

Purple: **T** changes height of ceiling sensor cubes, **G** changes walls sensor cubes top partition, glows top walls partition and ceiling sensor cubes.

Brown: **T** changes height of floor sensor cubes, **G** changes walls sensor cubes bottom partition, glows bottom wall partitions and floor sensor cubes.

Orange: **T** changes shape of all sensor cubes, **G** changes window sizes of all sensor cubes, glows all sensor cubes.

After choosing desired color release both **T** and **G** in any order to reset the device. Press either **T** and **G** for desired setting and move hand either left or right and you will immediately see the change take effect no calibrate the setting to desired form and release the held input. If while tampering with the setting the player decides to cancel his actions he could either press the other input so that both **T** and **G** are pressed and then release both buttons to reset the device or he could press the **0** input a short press, cancelling the changes, resetting the device and leaving the sensor cubes still visible. Leaving the sensor cubes visible is an option advised to new player until they grasp the locations of each utility, if one chooses to use this option than the orange setting offers way to make the sensor cubes less diverting and lowering the transparency of the sensor cubes could prove useful as well. A long press on the **0** input will reset the device and make the sensor cube follow your hand until input **0** is released leaving the sensor cube at hand location. This mechanic is only true for both joysticks and sensor orb.

Second step – choosing abilities and items: first the player will need to open the menu, the player will hold his left arm horizontally in front of his body and look down on the base of his hand, than the menu will open above his hand on a black board with buttons. To press the buttons push your hand down on the button until it begins to glow than press the **T** input. Press the powers than either the ability or item button to get to a menu screen with relevant pool of choices. Press on the ability you want to change first and after press on the desired ability, when player respawns next or when a different map is loaded the new ability will come to play instead of the old.

out of each respective pool the player can choose 2 abilities and 2 items:

Abilities

Mass repair: heals all ally ships including the player, using this ability incurs a cooldown during which the ability cannot be used.

Invisibility: renders the elite ship invisible, enemies wont target you but you can still be hit, using this ability incurs a cooldown during which the ability cannot be used.

Directional shield: when used deploys a shield on a side of the ship corresponding to where the utility button is placed: Top right – above, top left – below, front right – in front, front left – behind, left wall – to the left, right wall – to the right. The shield has a limited amount of health, when it expires the shield will shut down, while shield is down or deactivated shield recharges his health. To keep shield activated use **0** with a short press.

Items

Mine dispenser:

Wormhole:

Barrier:

All players receive two additional utilities: recall – after 6 seconds teleports player to base with full health points and self destruction – destroys player ship and damages nearby enemies. These and the item and ability utilities are all interchangeable between six designated locations. By pressing on **0** input longer than a second the utility will follow the player hand, if the player releases the **0** input while the hand is inside one of the outer sensor cubes the utility will be assigned to the corresponding designated location. If the player desires a “fresh start” the player can open the menu press the resets than the reset utilities buttons and instead of the joysticks two utilities will be assigned, each time the player places a utility at a designated location a new utility will appear in the joystick location until all utilities have been assigned and the joystick return to their place. Picking up a utility with a long **0** press will reveal the allocation of the utility: ability1 – AB1, ability2 – AB2, item1 – IT1, item2 – IT2, recall – REC, self destruction – SD, joystick fire – FIRE, joystick aim – AIM, sensor orb – ORB.

Third step – game settings: open menu and press the settings button, the screen opened will contain two red and two blue arrows and a human and alien button, if the players team is currently set to human the human button would be red while the alien button would be default white, if alien team is set than the alien would be blue and the human button would be default white. Under the arrow button there are two numbers presented, the numbers represent how many fighter ship to spawn each wave for the MOBA mode, alien fighter ship under the blue buttons and the human under the red. The player will use the arrow buttons according to their direction to adjust the number. Return to default menu screen and proceed to press the play button after which press the button that starts the desired mode.

Gameplay:

General: inside the simulation cockpit there are two joysticks, aim and fire, to grab a joystick and twist it use the **G** input and the joystick will follow the direction of your hand until you release the input. When using the trigger on the fire joystick the ship cannon will open fire, aim joystick does not have a secondary action. The axes of joystick movement are forward backwards and sideways. Aim joystick forward propels the ship forward, sideways axis turns ship nose sideways. Fire joystick forward axis makes ship either dive or soar and sideways axis rotates the ship on its side. With a short press of **0** input if the ship was using the cannon or moving the joysticks then those actions would persist even if the player removes his hand or releases the inputs. To cease the utility’s actions the press of either **T** or **G** would suffice.

MOBA: The player will kill enemy ships with his abilities items or cannon while keeping as many of his own side fighter ship alive as fodder while he positions and takes down the enemy base. Going ahead of the fighter ships to take the focus of the enemies would be a viable tactic to a skilled player. The enemy base spin around his planet defended by a turret which is difficult to evade, to win inflict enough damage on the enemy base.

CTF: no bases, after a short duration flags on both sides will spawn, any enemy ship touching the flag will pick him up carrying it until it either dies or returns to its base and places it on top of its flag. If the flag of both teams is stolen neither can score a point until the enemy ship carrying its flag dies and a friendly unit pass over it to pick it up. If enemy ships pick up a flag that was dropped due to death the flag will attach to it and she becomes the flag bearer. Fighter ships do not spawn in waves in this mode, there’s an initial spawn after which every time a fighter ship dies it respawns.

LOOP: go through loops at the right angle, each loop generates an arrow pointing in a random direction, enter the loop by flying the ship so its roll rotation matches that of the direction of the arrow, finishing early awards additional points. Improve your flying skills and break your friends records.

Code:

Building the world: opening the game starts the player on the wait map, there he can adjust game settings and switch abilities. Maps: wait, MOBA, CTF, LOOP. Except for loop map where the loop trail is set in the level itself, all objects generated by the worldbuilder actor are autonomous and do not need initialization, the data they require is stored inside the game instance and each object knows how to behave according to the map it is spawned into. Parent classes use enums set by their children to set important variables, for example to change the HP of all ships change the global variable of HP. To change the HP of a single ship go into destructible in its constructor, the information of each objects HP is stored there and you can change the relation to global HP for that specific object and watch easily the HP relation of all other objects in the same time so you could make a more informed decision and will aide in the future when implementing balancing.

Path: a spline designed as some fraction of a circle along which other objects follow, setting the speed of the objects traveling along the spline is determined by the given amount of seconds to complete path.

Destructible: objects sharing this parent class have a team to identify itself and HP to monitor when this objects die, if a destructible does not override the the whereToShootToKillSelf method than enemies will not target it.

Fightership: basic spaceship armed with a simple cannon that travels from ally base to enemy base, targetable, respawns on CTF and dies on MOBA.

Eliteship: equipped by abilities and controlled by the player, otherwise like the fightership, targetable, respawns on death.

Base: spawns minions and on death ends the game, victory goes to the team that kept its base alive also is placed on a planet, targetable, death means destruction.

Barrier: barrier deflects all damage but is only active due to an ability and for a short duration at that, not targetable, no possible death.

Shield: incorporeal rectangle, not targetable, deactivates on death.

Planet: indestructible sphere, not targetable, no possible death.

AI: a state machine starting with following path state, traveling from one base to next, until the fightership encounters an enemy or a flag. Upon meeting an enemy the ship will close the distance between itself and the enemy until it reaches effective weapon distance and transfer to shooting, when engaging an enemy the ship will chase the nearest enemy but when its starting to fire it will target the same foe until he either dies or moves out of weapon range. When all nearby enemies are destroyed the state will change and the ship will return to path at the most convenient location to join it. When the ship encounters the flag it firsts checks with it if an ally is already rushing the flag, if such an ally exists the ship will stand guard at the enemy ship teams base providing cover and preparing to steal the flag if the ally ever drops it. Otherwise the ship will charge at the flag and return with it hastily to the base in attempt to score a point. When moving towards a target every tick the rotation of the ship is adjusted to point at the target and impulse is given as long as speed does not exceed the maximum value.