Mediocre Commander

Alexander Wilson 100886121
Jared Sherwood 101035878
Tao Guo 100968344

Game Idea and Influence

For our game we looked a RTS game released in 2007 called Supreme Commander which takes place in the future. In Supreme Commander the player controls a walking mech which starts off the game building starting stuff like resource harvesters and engineer factories. The game has air, ground, and naval units as well as special “experimental” units which are built by the commander.Our goal is to build a game that has same kind of idea so that player can control multiple object and conquer the entire map by eliminating computer enemy.

Our Game Mechanics

In Mediocre Commander we ended up sticking to the core mechanics. We have two types of units, Air and Ground, and a factory that can spawn more units. The ground units we have are the Commander, who spawns in first, and the Light Armored Vehicle which is a general purpose ground unit that does decent damage. Our air units include the Bomber, which might now be better described as an Air Superiority Fighter due to the fact that it does not drop bombs, and the Gunship which does sustained damage.

Originally, we were planning on giving the Commander the ability to build more units, but that never ended up making it into the final release. We didn’t make engineer units for the same reason.

There is an enemy AI player who shares the same units as you do. Enemy units will move around and search for player units. When they find a player they’ll pursue and attack. Each unit has different attack and health stats.

We were also planning on adding stationary resource extractors as a buildable unit so that we could generate resources to spend on units, but we didn’t get to that point since we never got spot building to work.

Controls

Camera: Move the mouse to the edge of the screen to pan the camera in that direction. You can also move the camera by using the arrow keys.

Units: Units can be selected by clicking on them, or you can select multiple units by drawing a box around them. Once selected, units can be moved by right clicking on the map. All selected units will move towards the last place on the map that is right clicked. Selected units can attack

Factory: your factory can be toggled on/off using Q. 1, 2, 3 keys will change the unit that is being spawned by the factory. 1 will spawn Light Assault Vehicles, 2 will spawn Bombers and 3 will spawn gunships.

How We Met Technical Requirements

The first milestone is all about some basic selection and movement, we added those features into the final submission, you can still select units by drag on the screen, for the last milestone, we added navmesh to the entire map and added a navmeshAgent component for each player unit to accomplish reasonable movement when the unit is trying to cross the map.

Additionally, we have a commander which contains skeletal movement, the player can also control the commander to move around.

Our enemy units have a decent enough quality AI that enables meaningful gameplay. One of the requirements in the final submission is to give semi autonomous actions to a player unit. We couldn't think of a way to add that into the game in a way that would not be frustrating or otherwise hinder the game experience. We could have done something like making the player units attack automatically if an enemy is in range and that wouldn't have been difficult to do at all, but often in RTS games, players don't want their units acting on their own and instead want them to only do exactly as they are commanded.

We also ran into problems with integrating our flocking behaviour. Alexander made a very good flocking prototype that worked great, but he was unable to get it to work in our engine due to the way we designed our architecture.

Additional Features

Top down viewing minimap, player can keep track of the map in a more general view, but the minimap will not allow you to see the entire map, because that is kind of cheating, since we do not have fog of war. The textures were "politely borrowed" from a google image search.

The relatively complex map, terrain and trees are made by ourselves using Unity's terrain editor. We also built all of our unit models ourselves in 3ds Max using Autodesk's free student license.

Things we Learned

The biggest issue we faced during development was our weak implementation of the architecture we designed. Coming from OpenGL, we got a little too comfortable thinking that the component architecture would be a lot easier to deal with. We ended up all going our own ways in terms of architecture at the beginning and things were a little too messy and took forever to implement as the complexity of the game increased. We ended up fixing this in part later on into development but some parts of the game such as unit control and spawning were not quite fixed. If we were to start again we'd focus on building the architecture first to be very modular so that we can have the code segments acting a little more independently than they are now.