Our game is called ***19-Fortuna***, and is a 3D adventure game developed for the PC, and is compatible with the Xbox, and for players who are in their mid-teens, their ages ranging from 14 to 19. This game will interest both boys and girls alike, especially those who are interested in sci-fi games with a hint of fantasy.

The player has been dispatched to the asteroid known as 19-Fortuna with his/her mech, in order to mine some rare crystals for his/her company. However, upon arriving, the player discovers that the asteroid is also inhabited by hostile rock monsters.

The player must mine the crystals and avoid the monsters in order to complete the game. If the monsters destroy the player’s mech, then game is over.

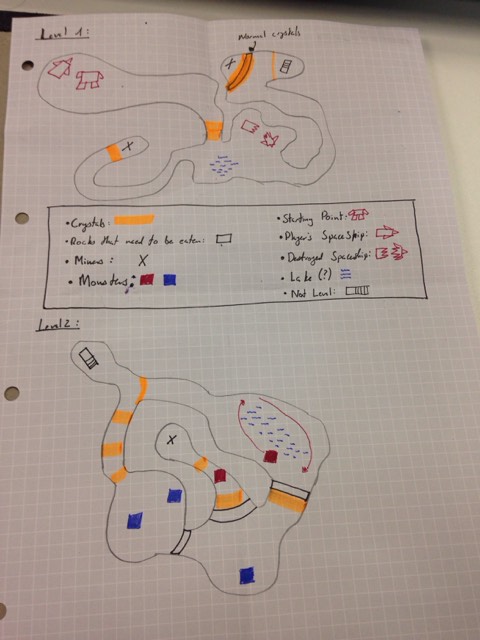
The game was developed on the Unity Game Engine, and was programmed with the C# language.

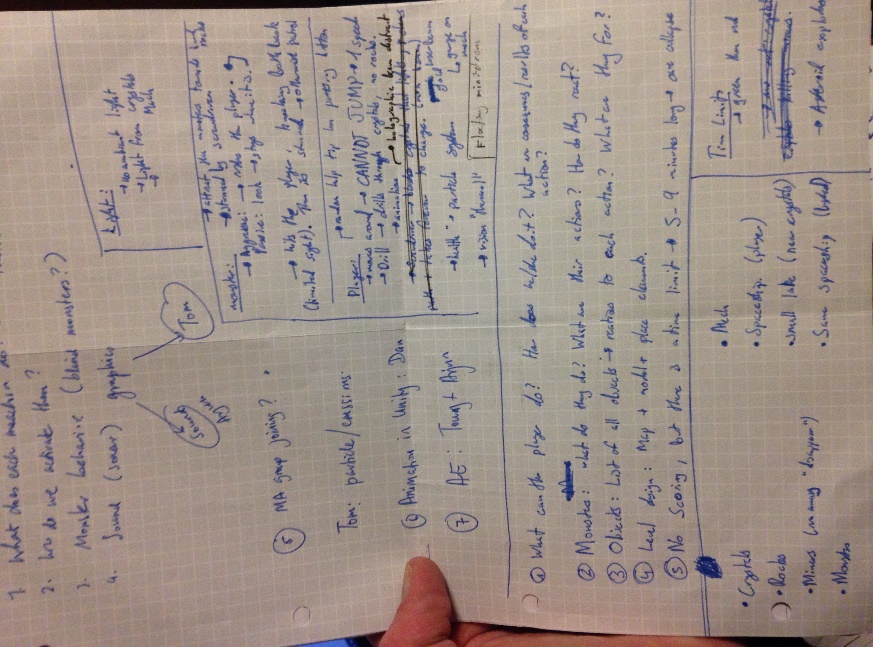
Although I worked primarily as a game programmer, I also participated in the development of other parts of the game, such as the game design and the project management.

When we first came up with the idea, I worked as one of the project managers alongside Arjun Khara. During our meetings, the two of us would take notes of all the ideas, milestones and tasks that needed to be completed, and then we would share them on either Facebook or Github.

This part especially useful when working on the game design and the level design of our game, as it gave all of us a very accurate picture of the game we were trying to build, along with a clear image of the appearances and the behaviors of each game object that would appear in our scenes.

I worked on taking notes, and creating a level design on paper, that I uploaded on the GitHub and Facebook groups.



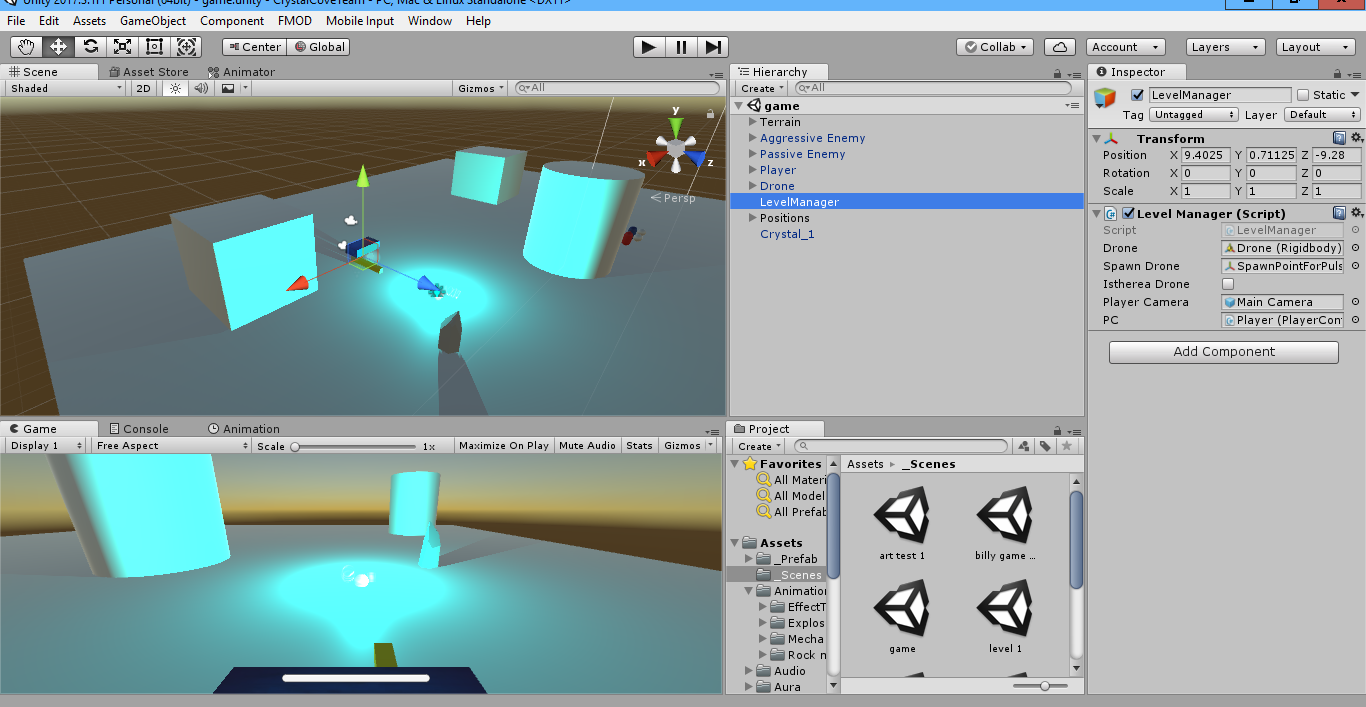


We also did some research in order to have an idea how our game should look. Using the pictures of 3D models and the screenshots of level designs, we all agreed that this game would be a low-poly 3D game that would take place in a dark environment illuminated by a few crystals.

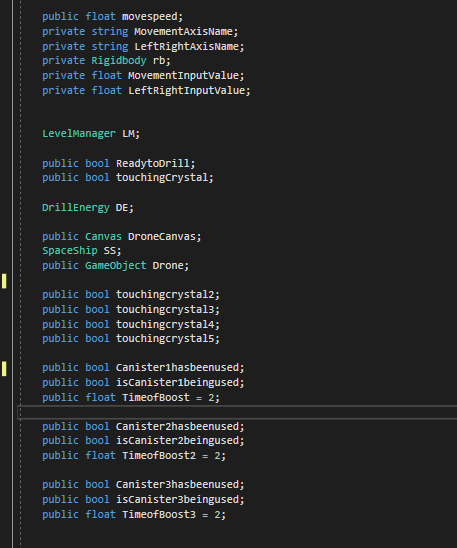
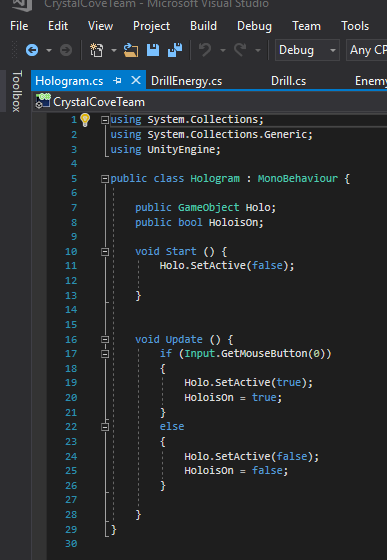
This research gave us a very clear idea about what our game was going to look like, and what it was going to be about.

It also gave me a clear idea what needed to be programmed, and gave me some ideas on how it should be developed.

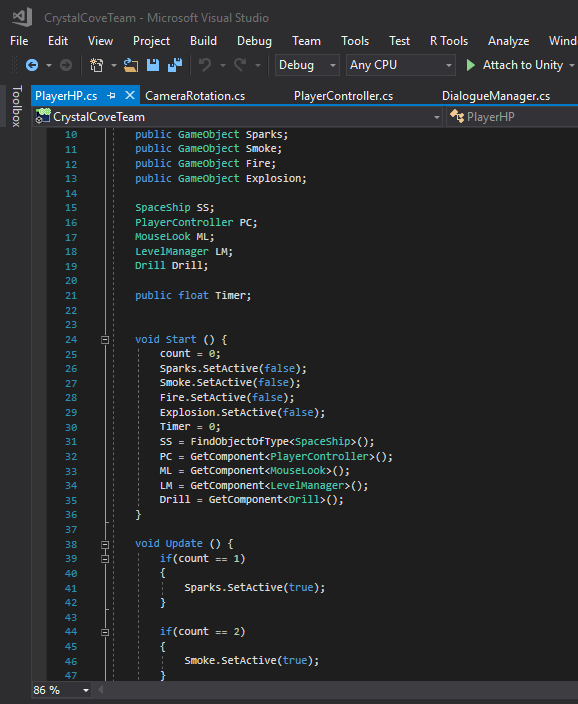
With the concept and design complete, I let Arjun take over the reins of the management, and I began to work as the programmer. I programmed multiple scenes using the Unity Game Engine, and the C# language.

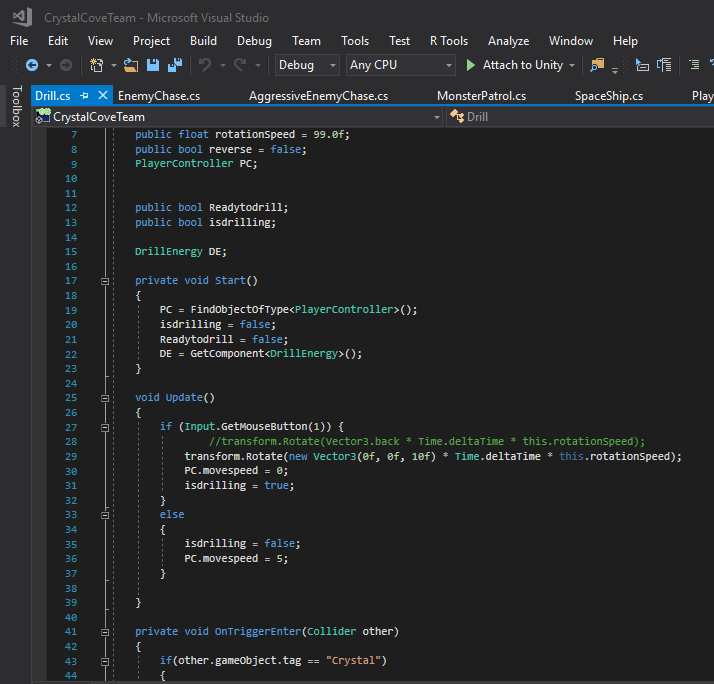
Prior to receiving any of the models, animations, textures, or audios, I worked exclusively with temporary placeholders, using the Unity gameobjects as the cubes and spheres.

In this scene, I worked on all the basic actions that would be taken by the AIs and the players, and that would be implemented in the final game: the player movement, drilling, releasing a hologram, launching a drone and controlling it, looking around with the mouse, the player’s HP, the enemy’s (passive and aggressive) actions.

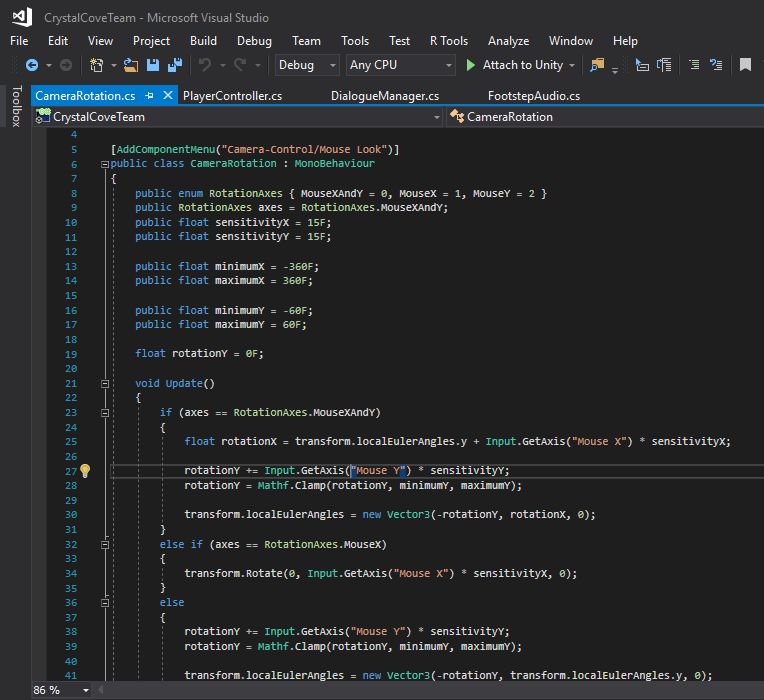
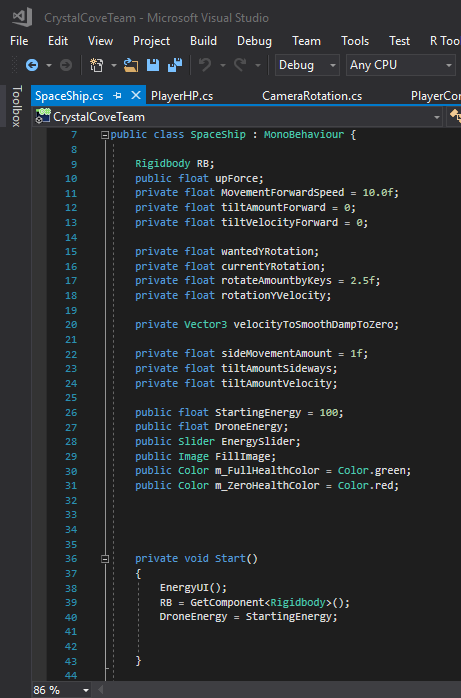
(Player Script on the left – gives the player the ability to walk boost, animations, release the drone.)

(Hologram Script underneath – creates a hologram

that distracts the monsters – the monster chase the hologram.

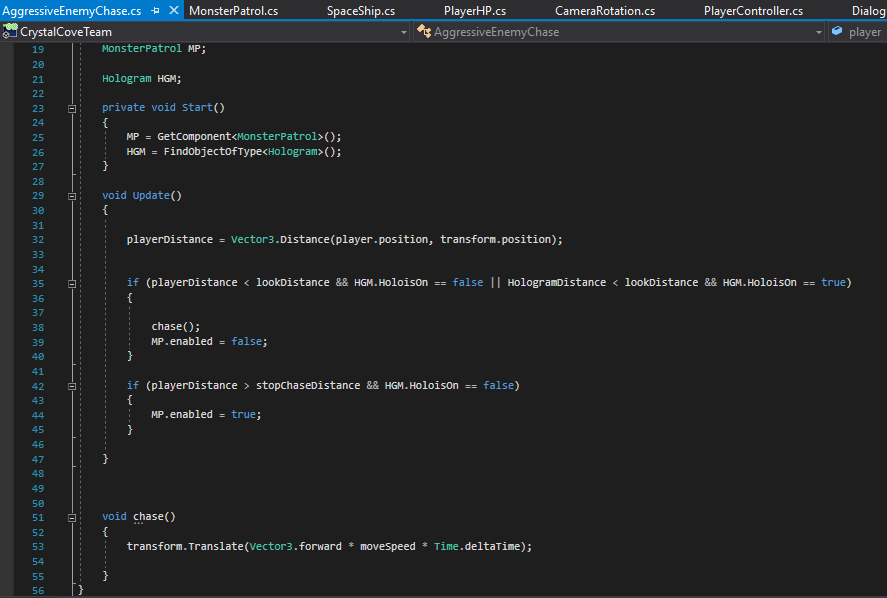


(Drill Script on the left. This causes the drill to spin, and destroy crystals after a few seconds)

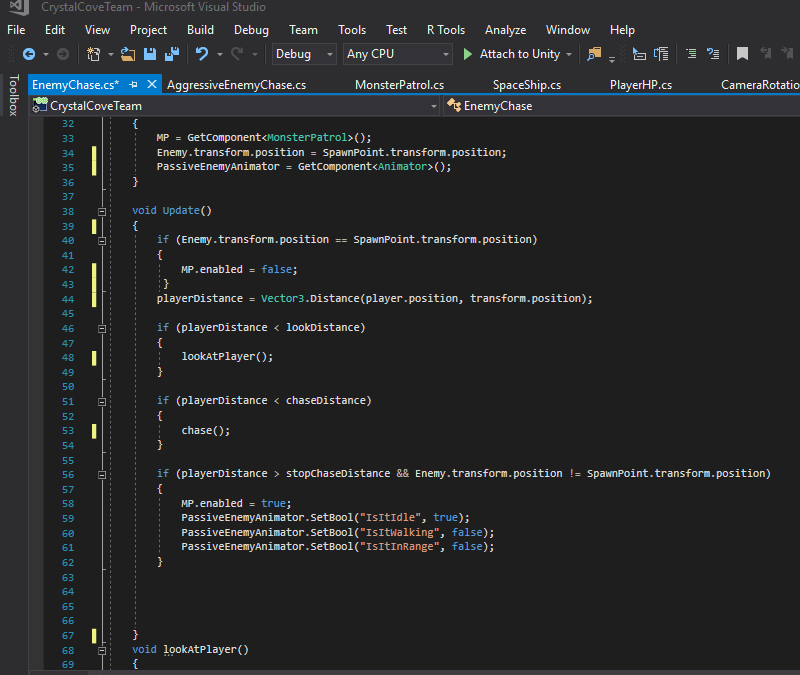
(PlayerHP on the right, whenever the player gets hit by a monster, a new particle system plays.)

(Camera Rotation with mouse on the left)

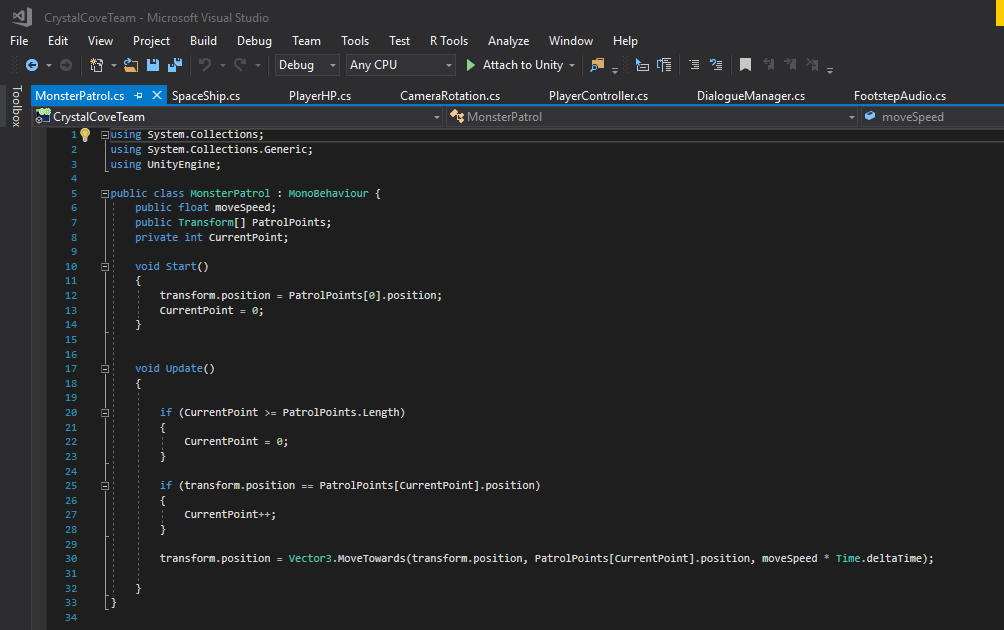
(Drone manipulation on the right. This spawns a drone, and the player can control this drone independently. The mech is static when the drone is flying.)



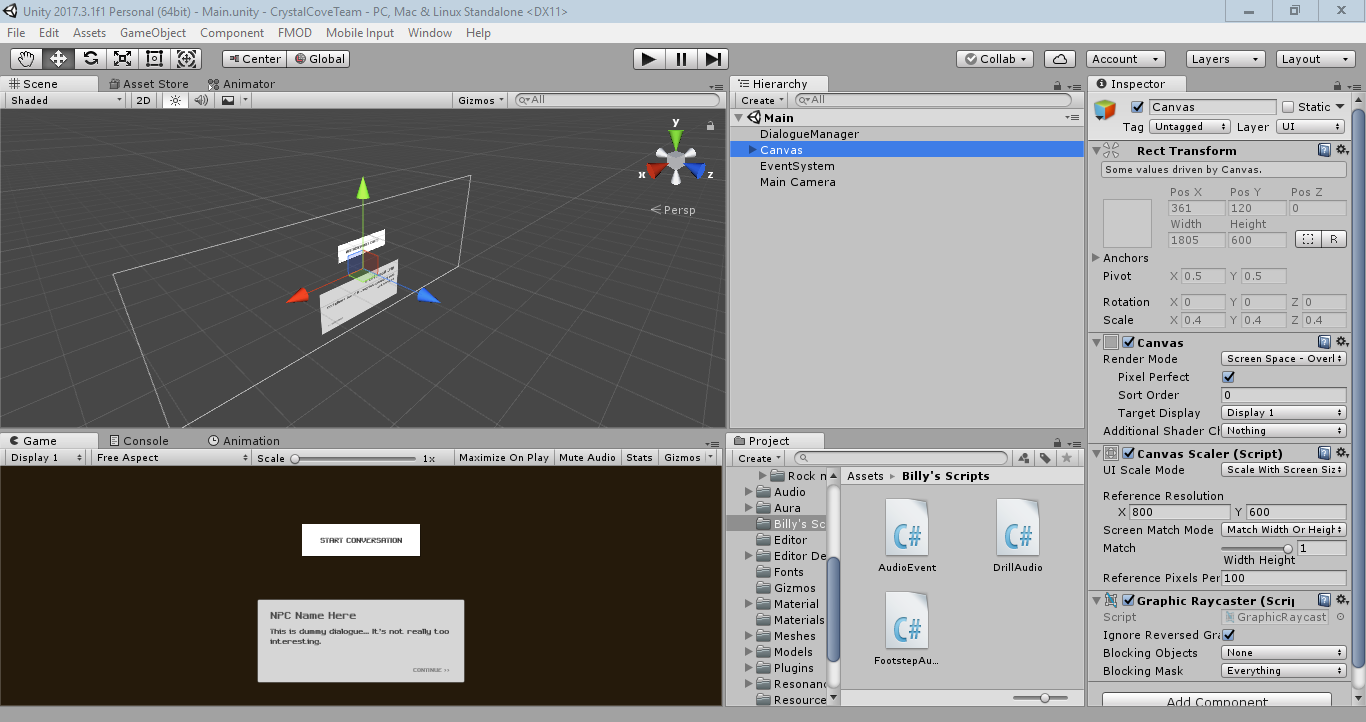
(The aggressive enemy script on the left. This script gives the monster the ability to chase the player)

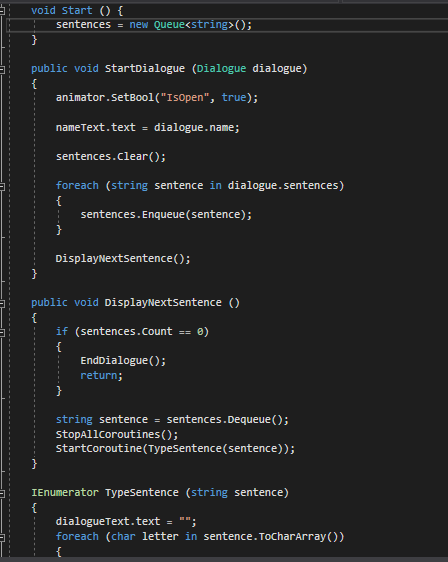


(The passive enemy script on the left. This script gives the monster the ability to observe the player, chase him/her, and the possibility to return to its original starting point when the player leaves its field of view)



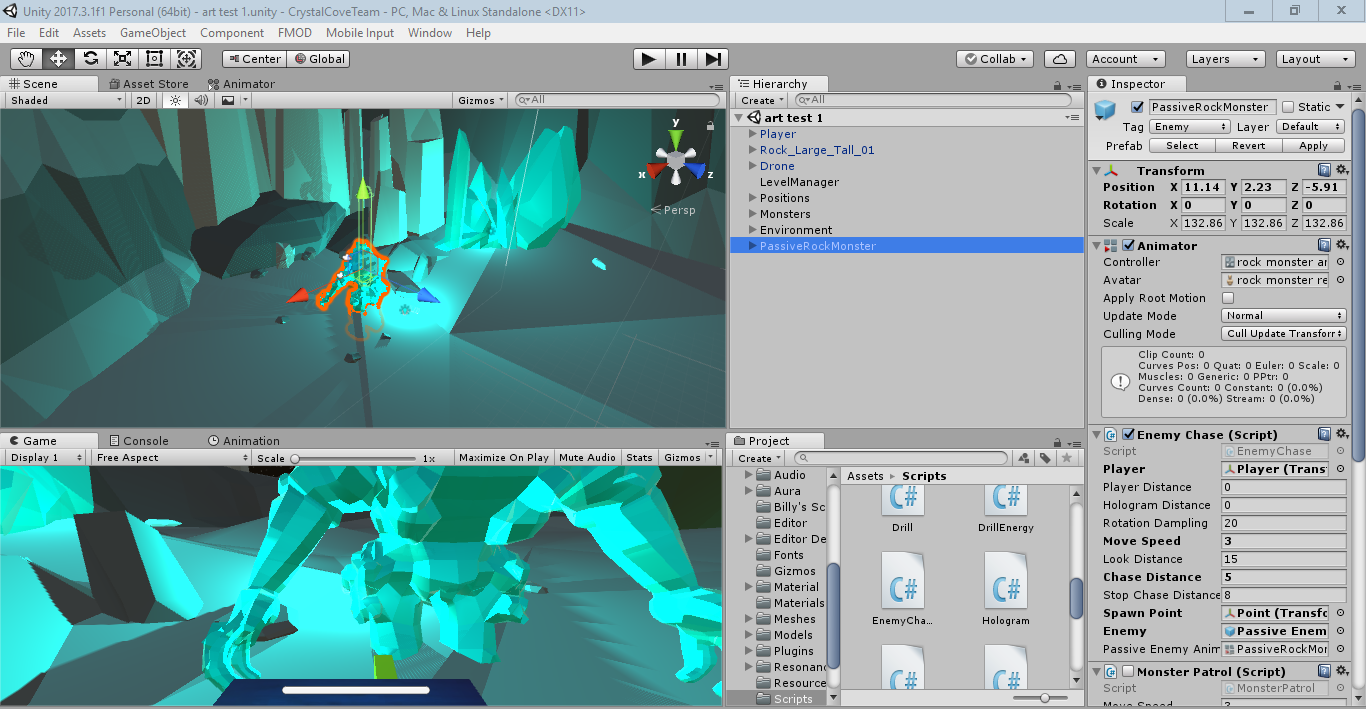
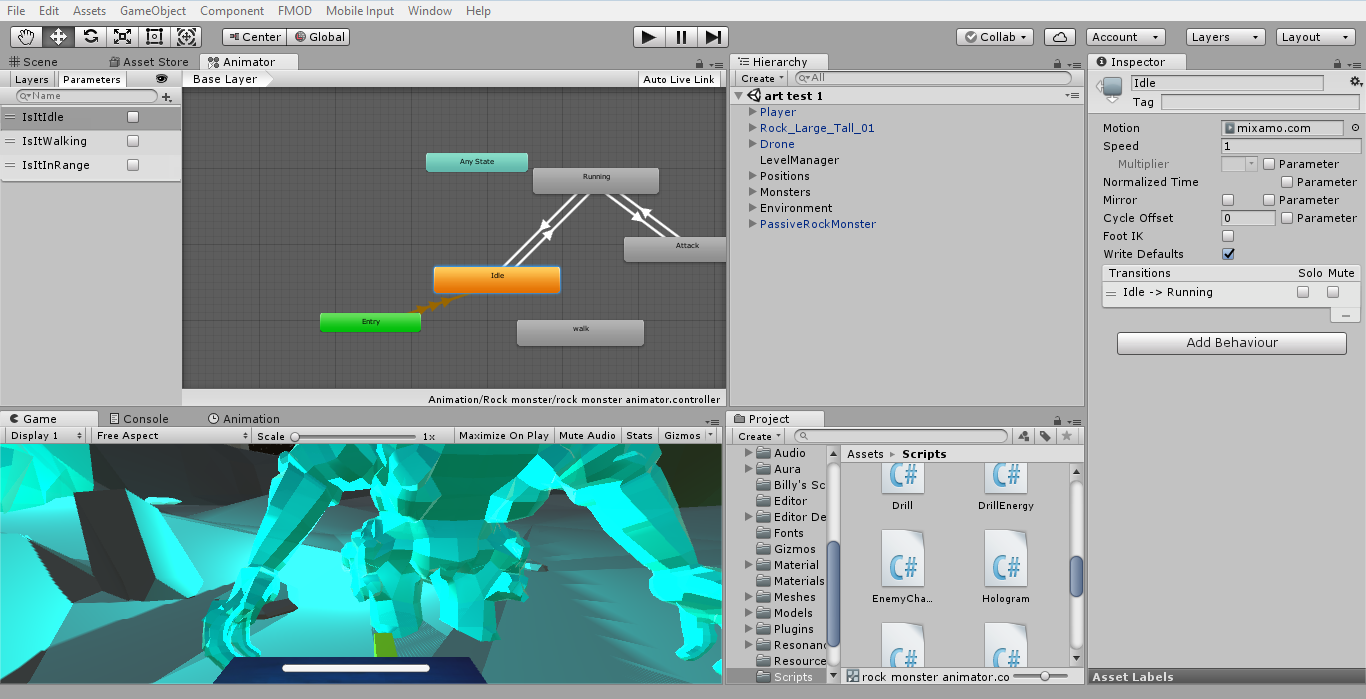
(This script gives the monsters the ability to patrol certain areas of the level)

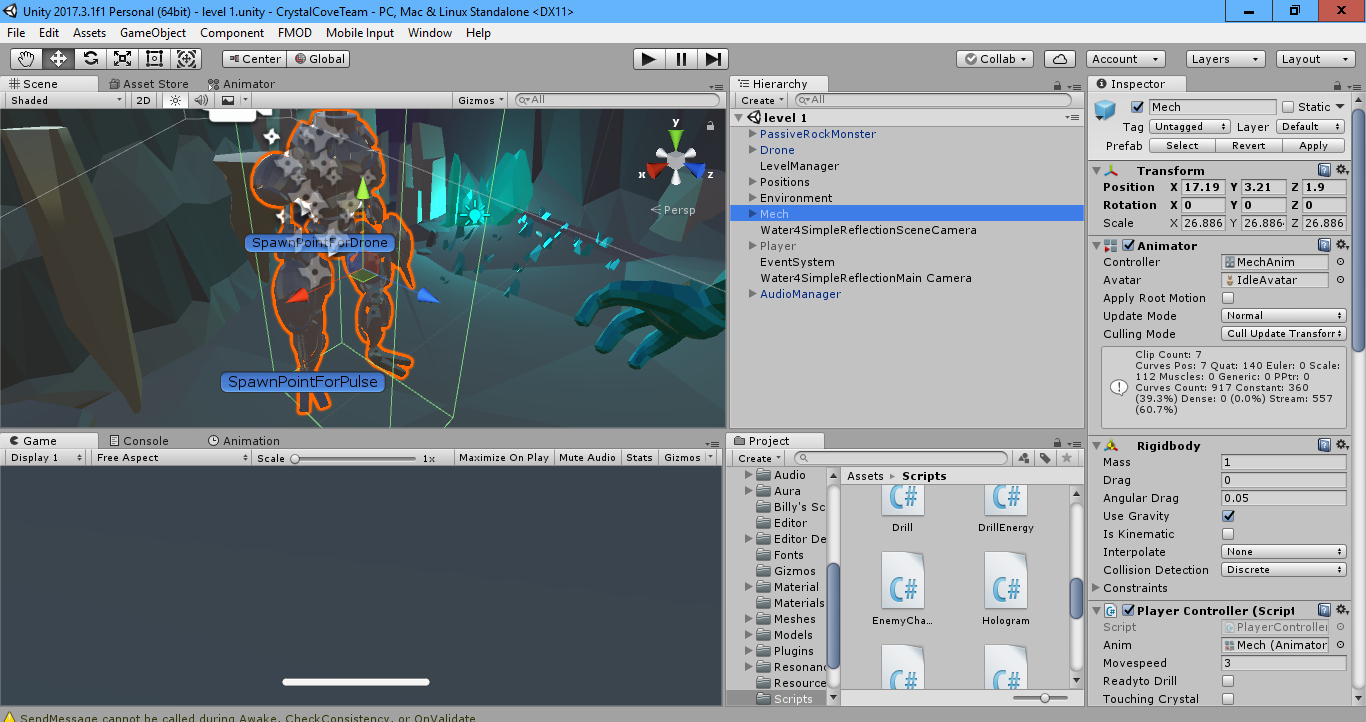
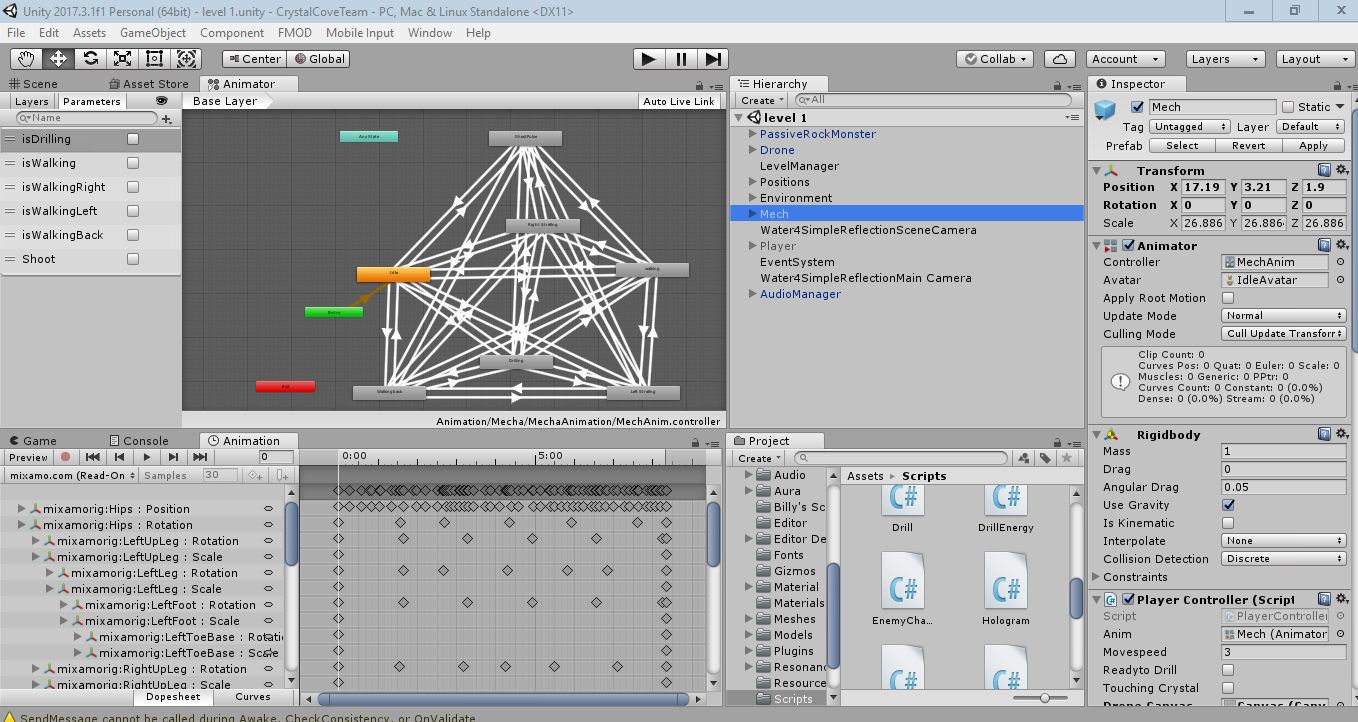
During the development phase, we also thought about creating dialogue box in order to pass on information to the player, similar to the way it is done in the Metroid franchise. We also intended on creating an intro using just a very short dialogue between unknown two characters:

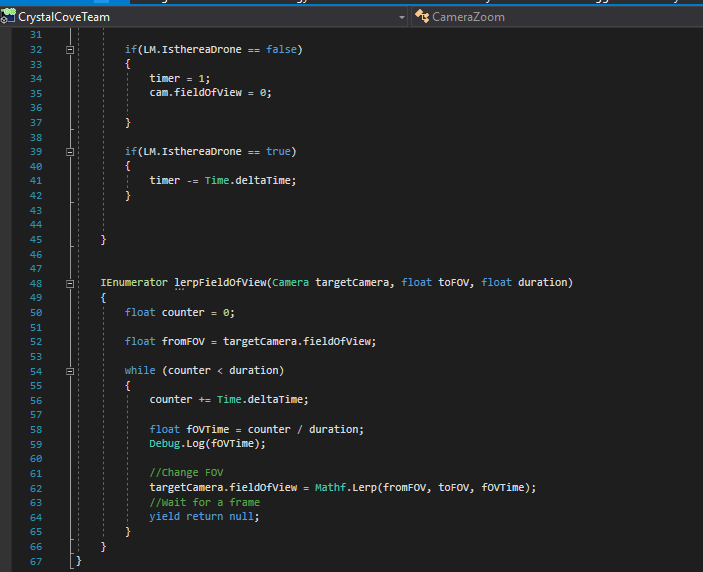
Using the animators and the scripts, I developed a dialogue box that slides the text out of the box once the player presses the ‘continue’ button.

This script is used for the dialogue box.

When the basic 3D environment models were developed, I imported my work into the scene with those models. My job was then to adapt the models and my scripts in order to get them to work with the new models.

I also began to work on implementing the animations on their appropriate models. Underneath is a screenshot of the animations for the rock monster’s animations. I also reprogrammed the scripts in order for the animations to play when they were needed: an idle animation, the running animation and an attacking animation.

Once the monster was properly implemented, I replaced the player placeholders, and replaced them with the player’s final model. I then connected the animations to the mech, and reprogrammed the scripts so that the animations would play in the proper moments.

After some feedback, I also created a script that plays when the player activated the drone. This script gives the impression that the drone is moving out of the mech with the use of optical illusions by playing around with the camera’s field of view, and the distance between the drone and the mech.

When testing the game, I also noticed that there was a mistake with the walking audio: it was always playing, even the player was static on the scene. I checked the scripts that Willy wrote and made the necessary corrections so the walking audio plays only when the mech is walking.

I also added a gauge to the drone, that limits how long it can stay in the air, in order to prevent the player from flying off the game.

Once our game was done, we began to change certain aspects of the game: the hologram was eventually removed, as well as the dialogue box.

I had also developed my own water physics, but we eventually removed those and kept Ece’s water effects.