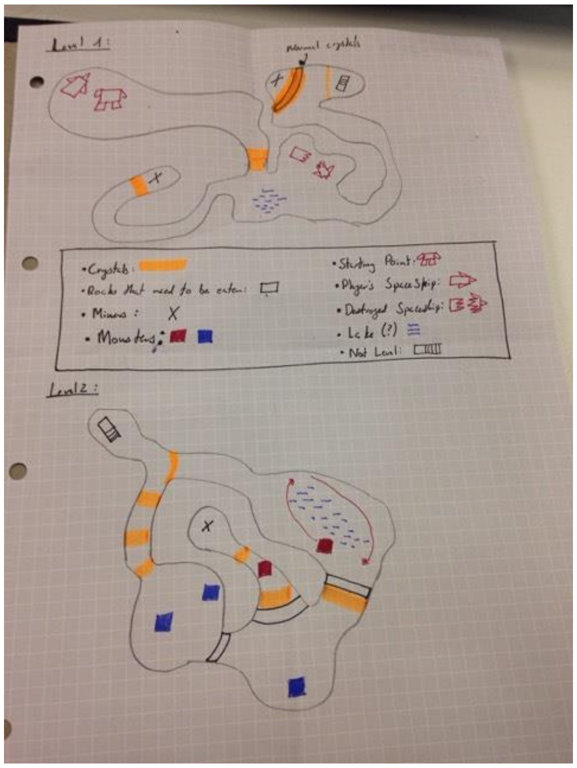
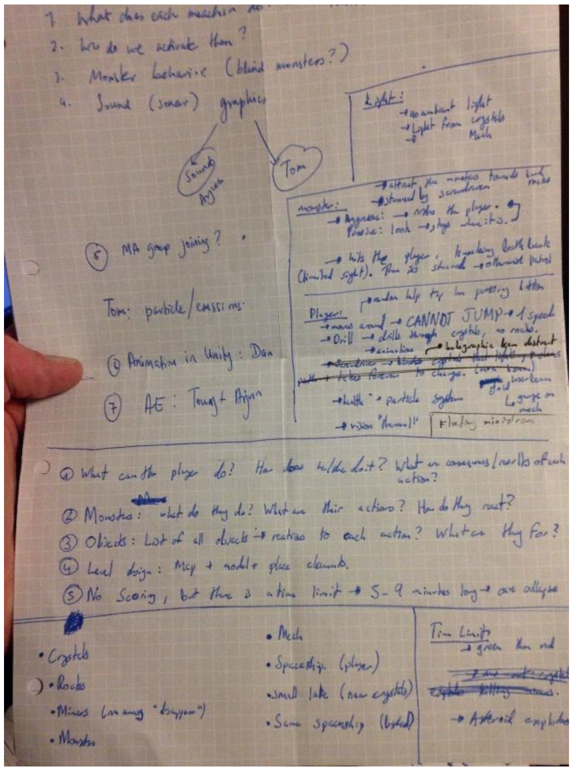
Our game is called ***19-Fortuna***, and is a 3D adventure game developed for the PC, and is compatible with the Xbox, and for players who are in their mid-teens, their ages ranging from 14 to 19. This game will interest both boys and girls alike, especially those who are interested in sci-fi games with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




1. What does each mechanism [illegible]
2. How do we activate them?
3. Monster behavior (blind monsters?)
4. Sound (sonar) graphics

Sound Arjun

Tom

Light:
→ no ambient light
→ Light from crystals Mech
→

Monster:
→ attract the monsters towards [illegible] rocks
→ stunned by [illegible]
→ Aggressive: → [illegible] the player.
Passive: look → [illegible]
→ hits the player, [illegible] back (limited sight). Then [illegible] stunned → [illegible] patrol

⑤ MA group joining?

Tom: particle/emissions

Player:
→ move around → CANNOT JUMP → 1 speed
→ [illegible] help [illegible] positing [illegible]
→ Drill → drills through crystals, no rocks.
→ animation → holographic beam distract
~~→ [illegible] → [illegible] crystals that [illegible]~~
~~[illegible] → takes [illegible]~~ to charge. [illegible]
Drill laser beam
→ health → particle system
Gauge on mech
→ vision "thermal" | Floating [illegible]

⑥ Animation in Unity: Dan

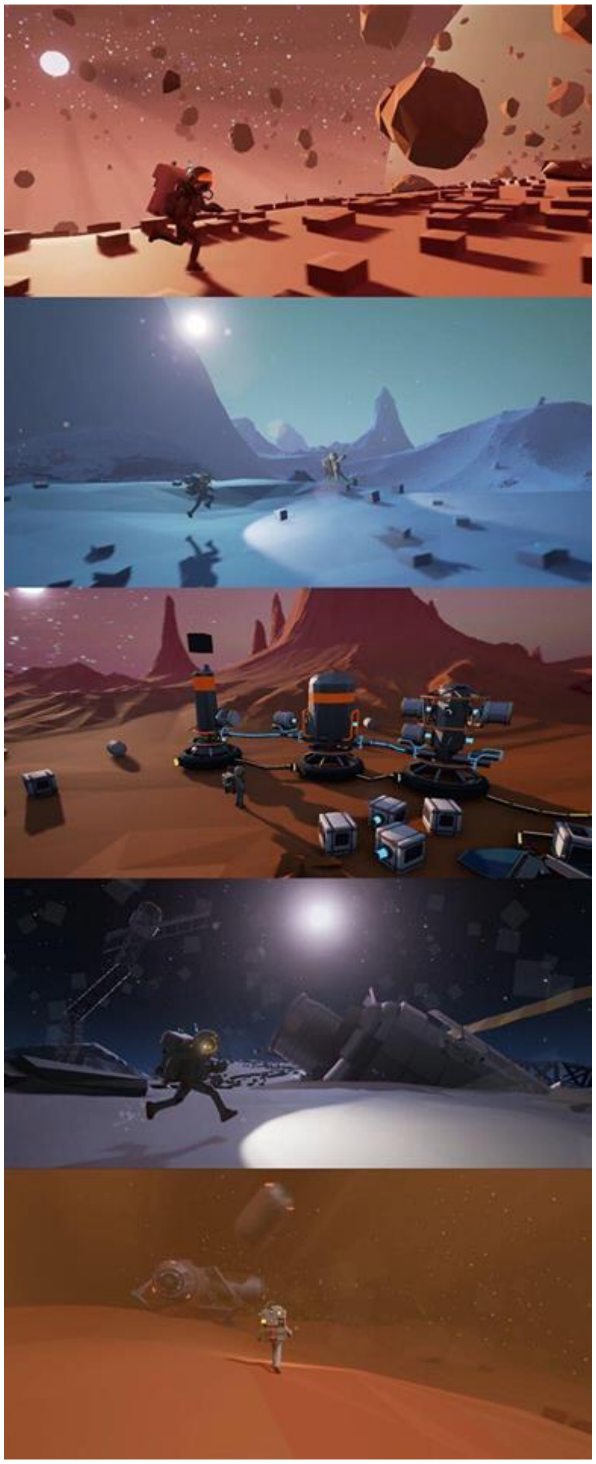
⑦ AE: Taught Arjun

① What can the player do? How does he/she do it? What are consequences/results of each action?
② Monsters: what do they do? What are their actions? How do they react?
③ Objects: List of all objects → reactions to each action? What are they for?
④ Level design: Map + model + place elements.
⑤ No Scoring, but there is a time limit → 5-9 minutes long → cave collapse

• Crystals
• Rocks
• Mines (or many "disappear")
• Monsters
• Mech
• Spaceship (player)
• small lake (new crystals)
• Some spaceship (busted)

Time Limits
→ green then red
~~[illegible]~~
~~[illegible] killing [illegible]~~.
→ Asteroid explodes

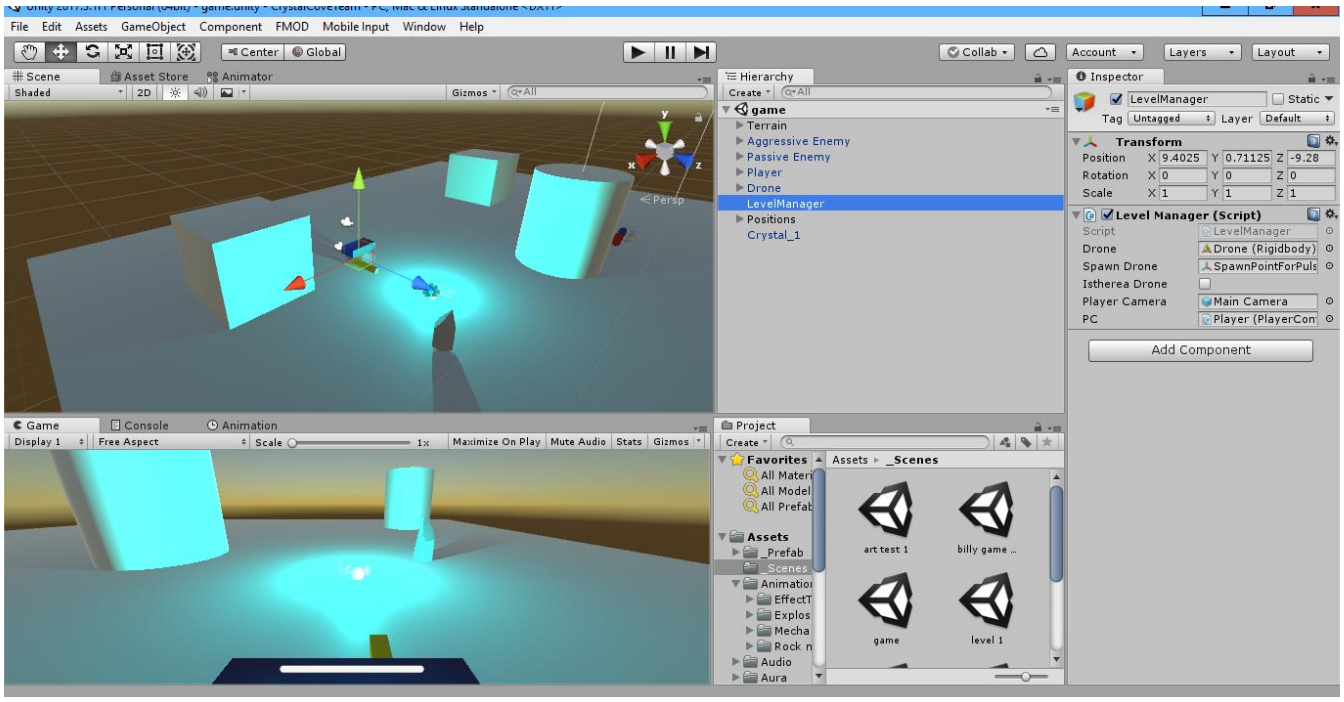
We also did some research in order to have an idea how our game should look. Using the pictures of 3D models and the screenshots of level designs, we all agreed that this game would be a low-poly 3D game that would take place in a dark environment illuminated by a few crystals.



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.



✓ LevelManager						Static
Tag	Untagged		Layer	Default		

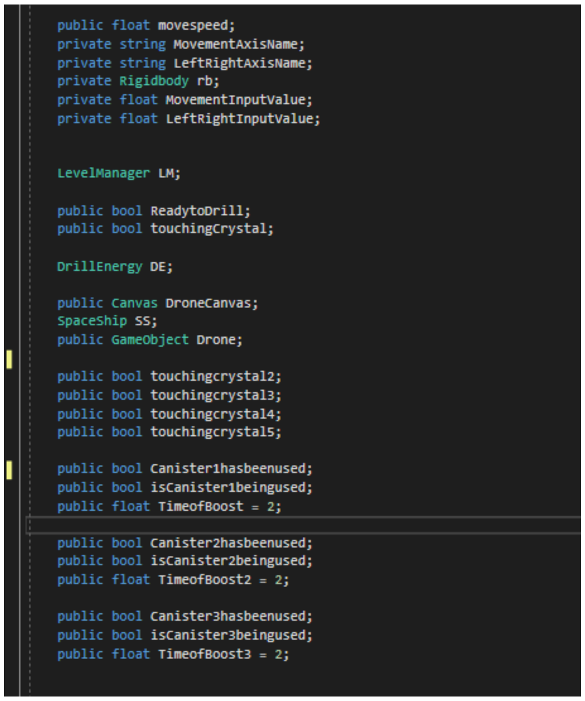
Transform

Position	X	9.4025	Y	0.71125	Z	-9.28
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

✓ Level Manager (Script)

Script	LevelManager
Drone	Drone (Rigidbody)
Spawn Drone	SpawnPointForPuls
Istherea Drone	☐
Player Camera	Main Camera
PC	Player (PlayerCon

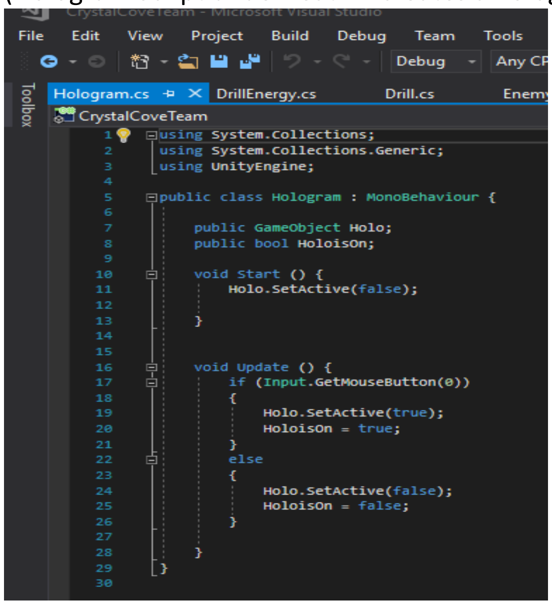
In this scene, I worked on all the basic actions that would be taken by the AIs and the players, and that would be implemented in the final game: the player movement, drilling, releasing a hologram, launching a drone and controlling it, looking around with the mouse, the player's HP, the enemy's (passive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



```
public float movespeed;
private string MovementAxisName;
private string LeftRightAxisName;
private Rigidbody rb;
private float MovementInputValue;
private float LeftRightInputValue;


LevelManager LM;

public bool ReadytoDrill;
public bool touchingCrystal;

DrillEnergy DE;

public Canvas DroneCanvas;
SpaceShip SS;
public GameObject Drone;

public bool touchingcrystal2;
public bool touchingcrystal3;
public bool touchingcrystal4;
public bool touchingcrystal5;

public bool Canister1hasbeenused;
public bool isCanister1beingused;
public float TimeofBoost = 2;

public bool Canister2hasbeenused;
public bool isCanister2beingused;
public float TimeofBoost2 = 2;

public bool Canister3hasbeenused;
public bool isCanister3beingused;
public float TimeofBoost3 = 2;
```

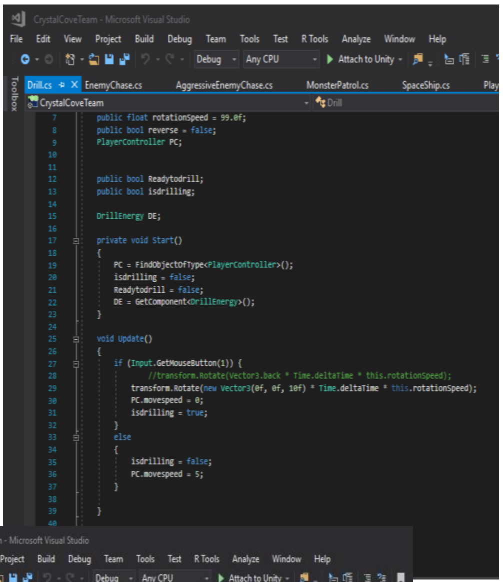
(Player Script on the left – gives the player the ability to walk boost, animations, release the drone.)

(Hologram Script underneath – creates a hologram


CrystalCoveTeam - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools
Debug Any CP
Toolbox
Hologram.cs DrillEnergy.cs Drill.cs Enemy
CrystalCoveTeam

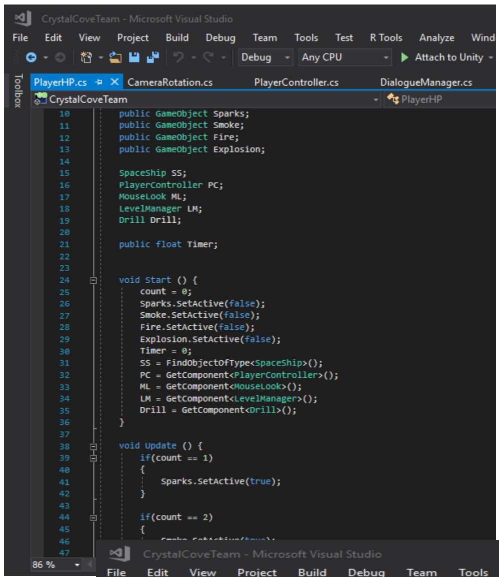
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Hologram : MonoBehaviour {

    public GameObject Holo;
    public bool HoloisOn;

    void Start () {
        Holo.SetActive(false);

    }


    void Update () {
        if (Input.GetMouseButton(0))
        {
            Holo.SetActive(true);
            HoloisOn = true;
        }
        else
        {
            Holo.SetActive(false);
            HoloisOn = false;
        }

    }
}
```

that distracts the monsters – the monster chase the hologram.


CrystalCoveTeam - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test R Tools Analyze Window Help
Debug Any CPU Attach to Unity
Toolbox
Drill.cs EnemyChase.cs AggressiveEnemyChase.cs MonsterPatrol.cs SpaceShip.cs
CrystalCoveTeam Drill

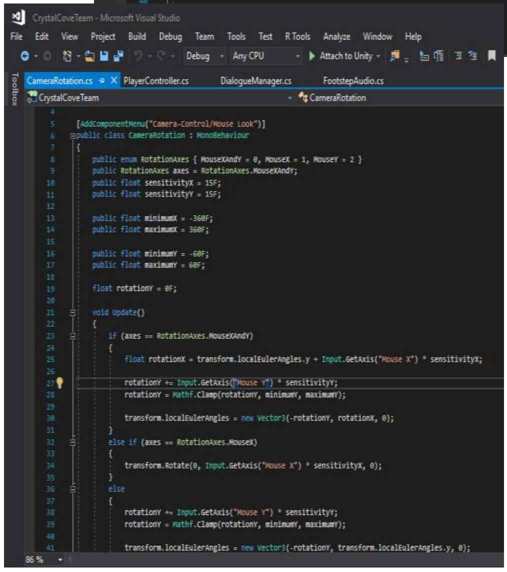
```
    public float rotationSpeed = 99.0f;
    public bool reverse = false;
    PlayerController PC;


    public bool Readytodrill;
    public bool isdrilling;

    DrillEnergy DE;

    private void Start()
    {
        PC = FindObjectOfType<PlayerController>();
        isdrilling = false;
        Readytodrill = false;
        DE = GetComponent<DrillEnergy>();
    }

    void Update()
    {
        if (Input.GetMouseButton(1)) {
            //transform.Rotate(Vector3.back * Time.deltaTime * this.rotationSpeed);
            transform.Rotate(new Vector3(0f, 0f, 10f) * Time.deltaTime * this.rotationSpeed);
            PC.movespeed = 0;
            isdrilling = true;
        }
        else
        {
            isdrilling = false;
            PC.movespeed = 5;
        }

    }
```

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


CrystalCoveTeam - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test R Tools Analyze Wind
Debug Any CPU Attach to Unity
Toolbox
PlayerHP.cs CameraRotation.cs PlayerController.cs DialogueManager.cs
CrystalCoveTeam PlayerHP
86 %

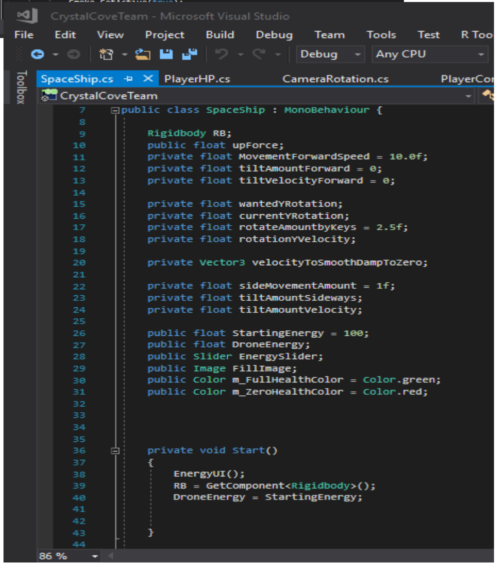
```
    public GameObject Sparks;
    public GameObject Smoke;
    public GameObject Fire;
    public GameObject Explosion;

    SpaceShip SS;
    PlayerController PC;
    MouseLook ML;
    LevelManager LM;
    Drill Drill;

    public float Timer;


    void Start () {
        count = 0;
        Sparks.SetActive(false);
        Smoke.SetActive(false);
        Fire.SetActive(false);
        Explosion.SetActive(false);
        Timer = 0;
        SS = FindObjectOfType<SpaceShip>();
        PC = GetComponent<PlayerController>();
        ML = GetComponent<MouseLook>();
        LM = GetComponent<LevelManager>();
        Drill = GetComponent<Drill>();
    }

    void Update () {
        if(count == 1)
        {
            Sparks.SetActive(true);
        }

        if(count == 2)
        {
```

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


CrystalCoveTeam - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test R Tools Analyze Window Help
Debug Any CPU Attach to Unity
Toolbox
CameraRotation.cs PlayerController.cs DialogueManager.cs FootstepAudio.cs
CrystalCoveTeam CameraRotation
86 %

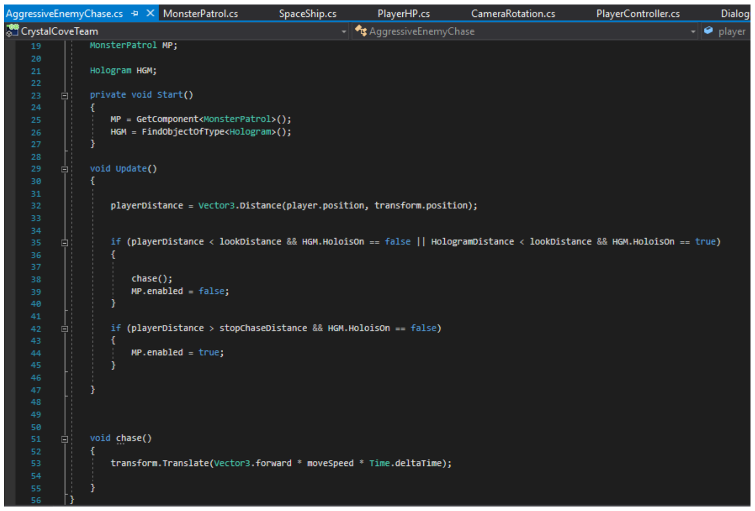
```
[AddComponentMenu("Camera-Control/Mouse Look")]
public class CameraRotation : MonoBehaviour
{
    public enum RotationAxes { MouseXAndY = 0, MouseX = 1, MouseY = 2 }
    public RotationAxes axes = RotationAxes.MouseXAndY;
    public float sensitivityX = 15F;
    public float sensitivityY = 15F;

    public float minimumX = -360F;
    public float maximumX = 360F;

    public float minimumY = -60F;
    public float maximumY = 60F;

    float rotationY = 0F;

    void Update()
    {
        if (axes == RotationAxes.MouseXAndY)
        {
            float rotationX = transform.localEulerAngles.y + Input.GetAxis("Mouse X") * sensitivityX;

            rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
            rotationY = Mathf.Clamp(rotationY, minimumY, maximumY);

            transform.localEulerAngles = new Vector3(-rotationY, rotationX, 0);
        }
        else if (axes == RotationAxes.MouseX)
        {
            transform.Rotate(0, Input.GetAxis("Mouse X") * sensitivityX, 0);
        }
        else
        {
            rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
            rotationY = Mathf.Clamp(rotationY, minimumY, maximumY);

            transform.localEulerAngles = new Vector3(-rotationY, transform.localEulerAngles.y, 0);
```

(Camera Rotation with mouse on the left)


CrystalCoveTeam - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test R Too
Debug Any CPU
Toolbox
SpaceShip.cs PlayerHP.cs CameraRotation.cs PlayerCor
CrystalCoveTeam
86 %

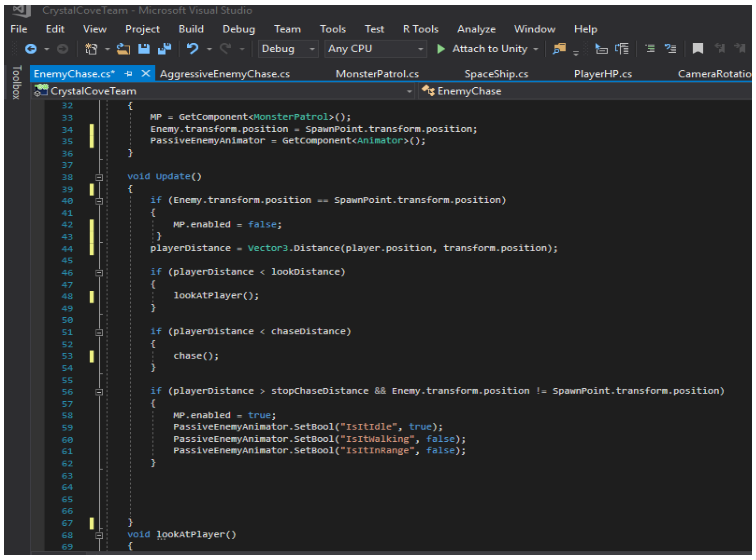
```
public class SpaceShip : MonoBehaviour {

    Rigidbody RB;
    public float upForce;
    private float MovementForwardSpeed = 10.0f;
    private float tiltAmountForward = 0;
    private float tiltVelocityForward = 0;

    private float wantedYRotation;
    private float currentYRotation;
    private float rotateAmountbyKeys = 2.5f;
    private float rotationYVelocity;

    private Vector3 velocityToSmoothDampToZero;

    private float sideMovementAmount = 1f;
    private float tiltAmountSideways;
    private float tiltAmountVelocity;

    public float StartingEnergy = 100;
    public float DroneEnergy;
    public Slider EnergySlider;
    public Image FillImage;
    public Color m_FullHealthColor = Color.green;
    public Color m_ZeroHealthColor = Color.red;



    private void Start()
    {
        EnergyUI();
        RB = GetComponent<Rigidbody>();
        DroneEnergy = StartingEnergy;


    }
```

(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.)



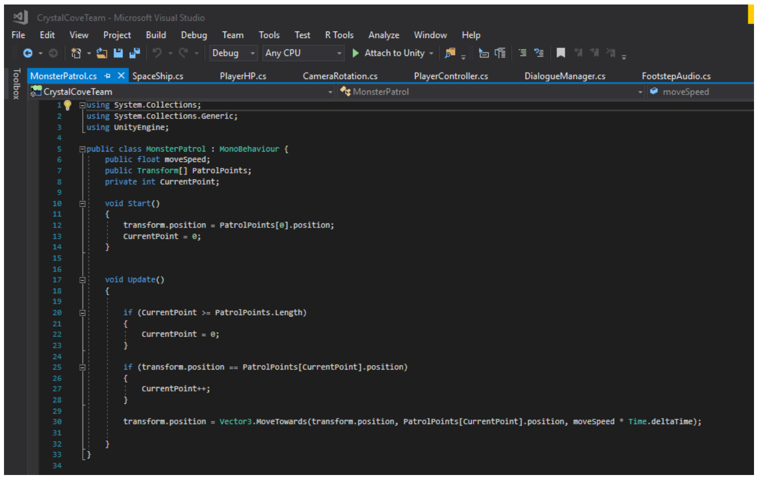
```csharp
    MonsterPatrol MP;

    Hologram HGM;

    private void Start()
    {
        MP = GetComponent<MonsterPatrol>();
        HGM = FindObjectOfType<Hologram>();
    }

    void Update()
    {

        playerDistance = Vector3.Distance(player.position, transform.position);


        if (playerDistance < lookDistance && HGM.HoloisOn == false || HologramDistance < lookDistance && HGM.HoloisOn == true)
        {

            chase();
            MP.enabled = false;
        }

        if (playerDistance > stopChaseDistance && HGM.HoloisOn == false)
        {
            MP.enabled = true;
        }

    }



    void chase()
    {
        transform.Translate(Vector3.forward * moveSpeed * Time.deltaTime);

    }
}
```

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



```csharp
    {
        MP = GetComponent<MonsterPatrol>();
        Enemy.transform.position = SpawnPoint.transform.position;
        PassiveEnemyAnimator = GetComponent<Animator>();
    }

    void Update()
    {
        if (Enemy.transform.position == SpawnPoint.transform.position)
        {
            MP.enabled = false;
        }
        playerDistance = Vector3.Distance(player.position, transform.position);

        if (playerDistance < lookDistance)
        {
            lookAtPlayer();
        }

        if (playerDistance < chaseDistance)
        {
            chase();
        }

        if (playerDistance > stopChaseDistance && Enemy.transform.position != SpawnPoint.transform.position)
        {
            MP.enabled = true;
            PassiveEnemyAnimator.SetBool("IsItIdle", true);
            PassiveEnemyAnimator.SetBool("IsItWalking", false);
            PassiveEnemyAnimator.SetBool("IsItInRange", false);
        }




    }
    void lookAtPlayer()
    {
```

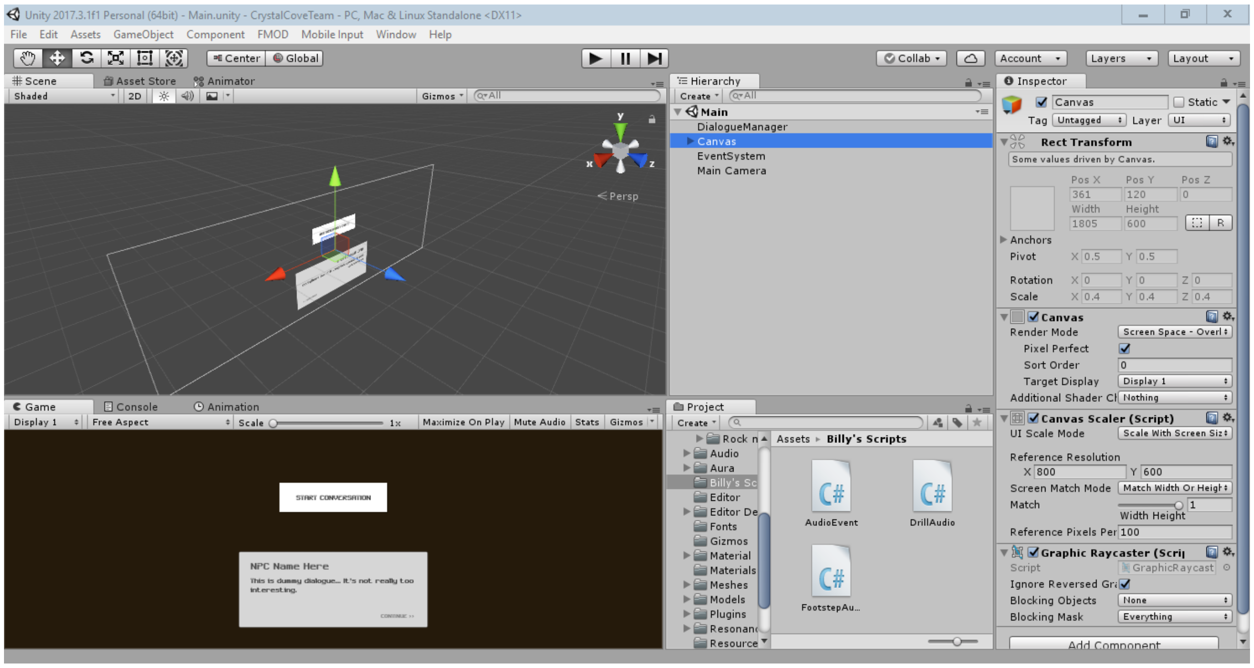
(The passive enemy script on the left. This script gives the monster the ability to observe the player, chase him/her, and the possibility to return to its original starting point when the player leaves its field of view)



```csharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MonsterPatrol : MonoBehaviour {
    public float moveSpeed;
    public Transform[] PatrolPoints;
    private int CurrentPoint;

    void Start()
    {
        transform.position = PatrolPoints[0].position;
        CurrentPoint = 0;
    }


    void Update()
    {

        if (CurrentPoint >= PatrolPoints.Length)
        {
            CurrentPoint = 0;
        }

        if (transform.position == PatrolPoints[CurrentPoint].position)
        {
            CurrentPoint++;
        }

        transform.position = Vector3.MoveTowards(transform.position, PatrolPoints[CurrentPoint].position, moveSpeed * Time.deltaTime);

    }
}
```

(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:



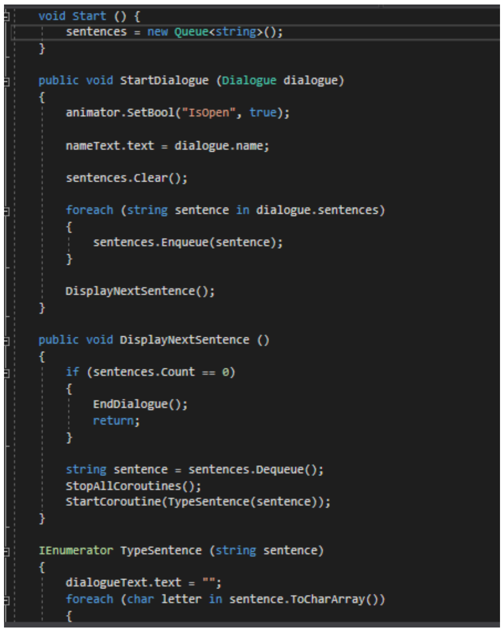
Unity 2017.3.1f1 Personal (64bit) - Main.unity - CrystalCoveTeam - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component FMOD Mobile Input Window Help
Scene
Asset Store
Animator
Hierarchy
Inspector
Game
Console
Animation
Project
START CONVERSATION
NPC Name Here
This is dummy dialogue... It's not really too interesting.
CONTINUE >>

Main
DialogueManager
Canvas
EventSystem
Main Camera

Assets ▸ Billy's Scripts
AudioEvent
DrillAudio
FootstepAu...

Canvas
Tag Untagged   Layer UI
Rect Transform
Some values driven by Canvas.
Pos X 361   Pos Y 120   Pos Z 0
Width 1805   Height 600
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 0.4 Y 0.4 Z 0.4
Canvas
Render Mode Screen Space - Overlay
Pixel Perfect ✓
Sort Order 0
Target Display Display 1
Additional Shader Channels Nothing
Canvas Scaler (Script)
UI Scale Mode Scale With Screen Size
Reference Resolution X 800 Y 600
Screen Match Mode Match Width Or Height
Match 1
Reference Pixels Per Unit 100
Graphic Raycaster (Script)
Script GraphicRaycaster
Ignore Reversed Graphics ✓
Blocking Objects None
Blocking Mask Everything

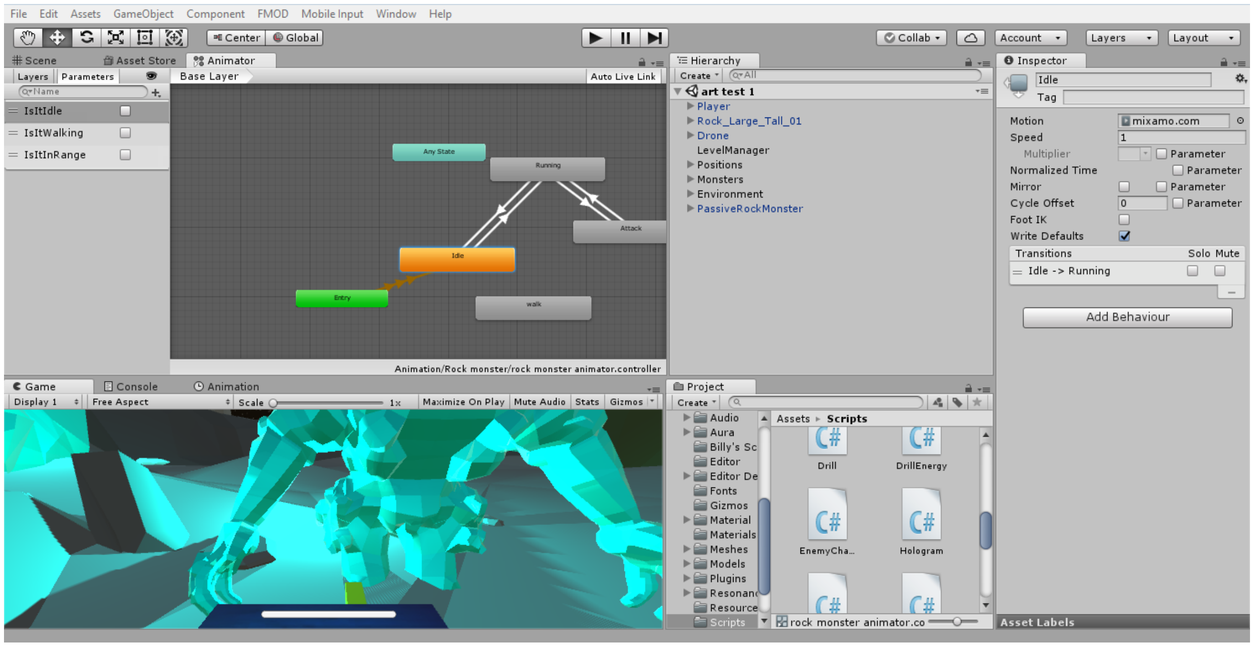
Using the animators and the scripts, I developed a dialogue box that slides the text out of the box once the player presses the 'continue' button.



```
void Start () {
    sentences = new Queue<string>();
}

public void StartDialogue (Dialogue dialogue)
{
    animator.SetBool("IsOpen", true);

    nameText.text = dialogue.name;

    sentences.Clear();

    foreach (string sentence in dialogue.sentences)
    {
        sentences.Enqueue(sentence);
    }

    DisplayNextSentence();
}

public void DisplayNextSentence ()
{
    if (sentences.Count == 0)
    {
        EndDialogue();
        return;
    }

    string sentence = sentences.Dequeue();
    StopAllCoroutines();
    StartCoroutine(TypeSentence(sentence));
}

IEnumerator TypeSentence (string sentence)
{
    dialogueText.text = "";
    foreach (char letter in sentence.ToCharArray())
    {
```

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.



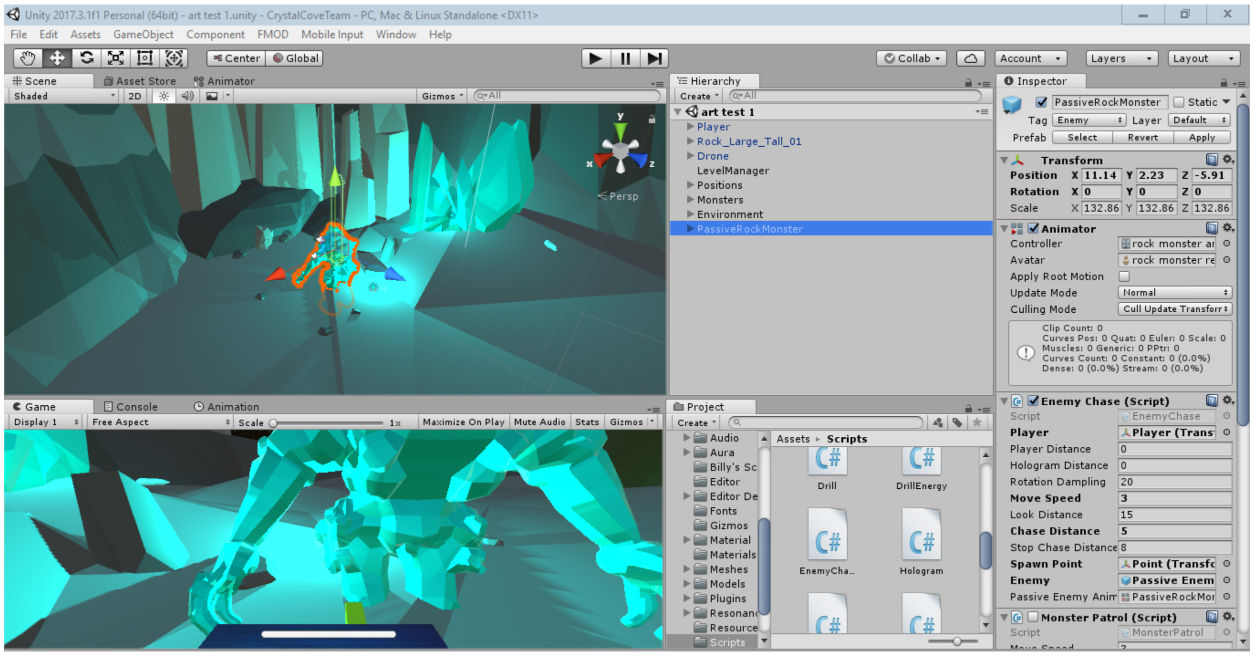
Parameters:
- IsItIdle
- IsItWalking
- IsItInRange

Animation/Rock monster/rock monster animator.controller

Hierarchy: art test 1 — Player, Rock_Large_Tall_01, Drone, LevelManager, Positions, Monsters, Environment, PassiveRockMonster

Inspector: Idle
- Motion: mixamo.com
- Speed: 1
- Multiplier: Parameter
- Normalized Time: Parameter
- Mirror: Parameter
- Cycle Offset: 0, Parameter
- Foot IK
- Write Defaults: ✓
- Transitions: Idle -> Running

Project: Assets ▸ Scripts — Drill, DrillEnergy, EnemyCha.., Hologram; rock monster animator.co

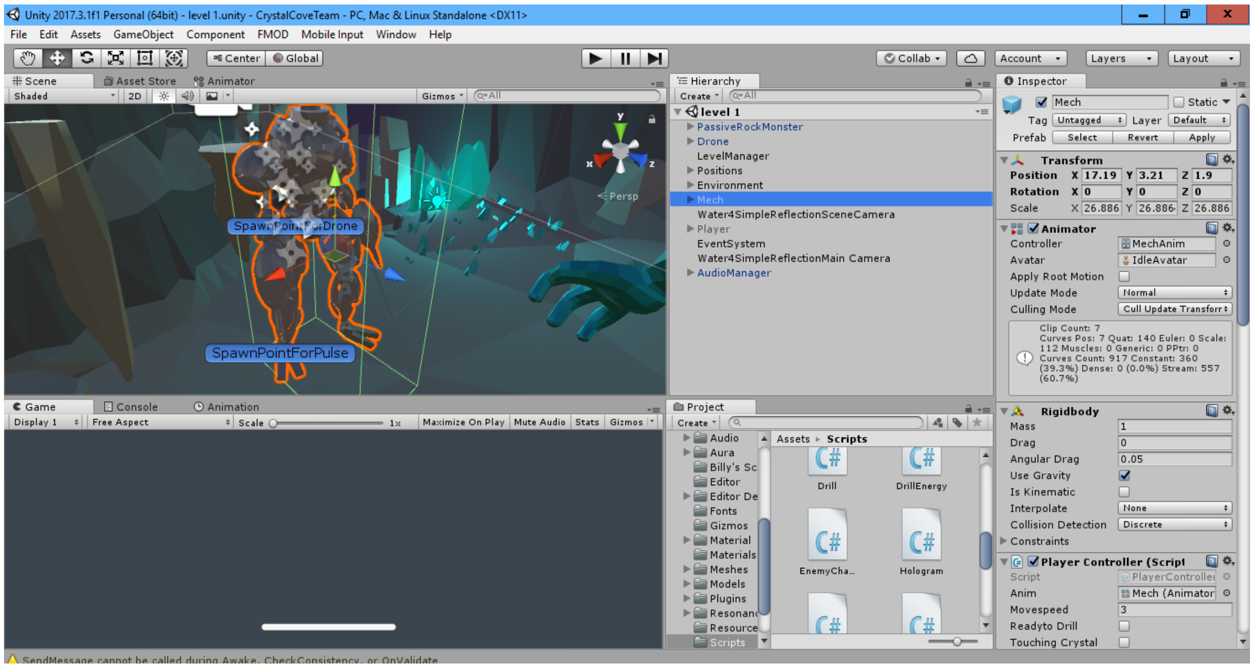


Unity 2017.3.1f1 Personal (64bit) - art test 1.unity - CrystalCoveTeam - PC, Mac & Linux Standalone <DX11>

Hierarchy: art test 1 — Player, Rock_Large_Tall_01, Drone, LevelManager, Positions, Monsters, Environment, PassiveRockMonster

Inspector: PassiveRockMonster
- Tag: Enemy; Layer: Default
- Transform — Position X 11.14, Y 2.23, Z -5.91; Rotation X 0, Y 0, Z 0; Scale X 132.86, Y 132.86, Z 132.86
- Animator — Controller: rock monster ar; Avatar: rock monster re; Apply Root Motion: ☐; Update Mode: Normal; Culling Mode: Cull Update Transform
- Clip Count: 0; Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0; Muscles: 0 Generic: 0 PPtr: 0; Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 0 (0.0%)
- Enemy Chase (Script) — Script: EnemyChase; Player: Player (Trans; Player Distance: 0; Hologram Distance: 0; Rotation Dampling: 20; Move Speed: 3; Look Distance: 15; Chase Distance: 5; Stop Chase Distance: 8; Spawn Point: Point (Transfc; Enemy: Passive Enem; Passive Enemy Anim: PassiveRockMor
- Monster Patrol (Script) — Script: MonsterPatrol

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.

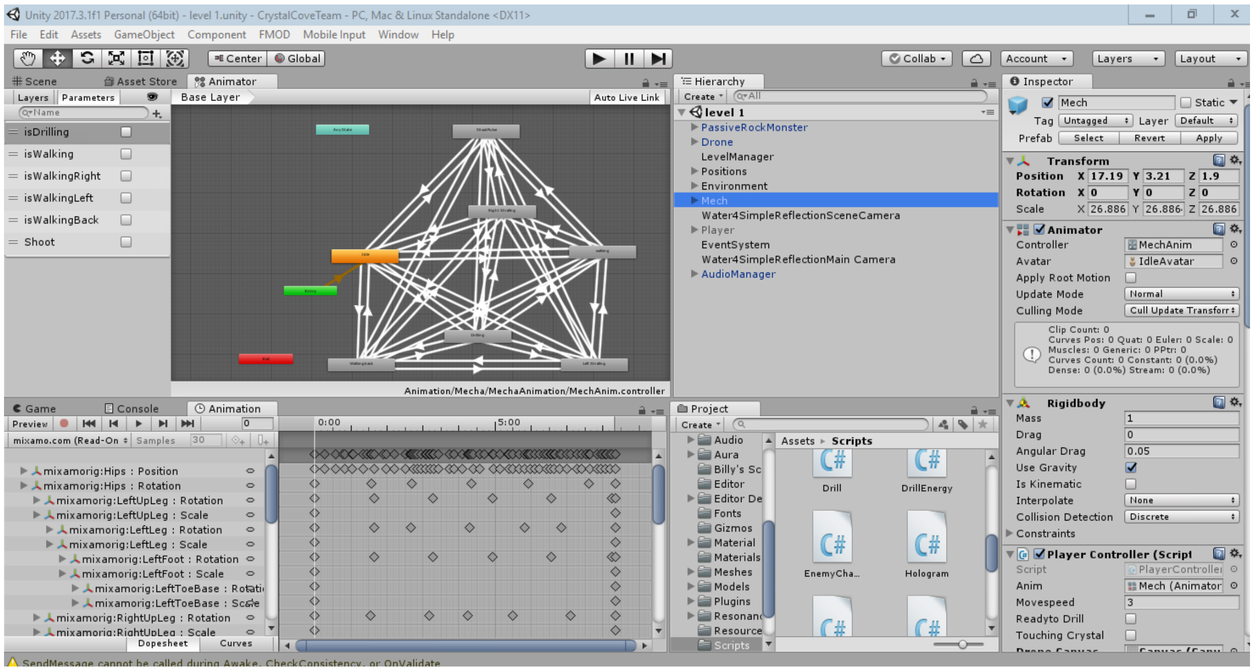

Unity 2017.3.1f1 Personal (64bit) - level 1.unity - CrystalCoveTeam - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component FMOD Mobile Input Window Help
Center Global
Collab Account Layers Layout
Scene Asset Store Animator
Shaded 2D Gizmos
SpawnPointForDrone
SpawnPointForPulse
Persp
Game Console Animation
Display 1 Free Aspect Scale 1x Maximize On Play Mute Audio Stats Gizmos
Hierarchy
Create
Project
Create
Inspector

level 1
PassiveRockMonster
Drone
LevelManager
Positions
Environment
Mech
Water4SimpleReflectionSceneCamera
Player
EventSystem
Water4SimpleReflectionMain Camera
AudioManager

Assets ▸ Scripts
Audio, Aura, Billy's Sc, Editor, Editor De, Fonts, Gizmos, Material, Materials, Meshes, Models, Plugins, Resonan, Resource, Scripts
Drill, DrillEnergy, EnemyCha..., Hologram

Mech Static
Tag Untagged Layer Default
Prefab Select Revert Apply
Transform
Position X 17.19 Y 3.21 Z 1.9
Rotation X 0 Y 0 Z 0
Scale X 26.886 Y 26.886 Z 26.886
Animator
Controller MechAnim
Avatar IdleAvatar
Apply Root Motion
Update Mode Normal
Culling Mode Cull Update Transform
Clip Count: 7
Curves Pos: 7 Quat: 140 Euler: 0 Scale: 112 Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 917 Constant: 360 (39.3%) Dense: 0 (0.0%) Stream: 557 (60.7%)
Rigidbody
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity ✓
Is Kinematic
Interpolate None
Collision Detection Discrete
Constraints
Player Controller (Script)
Script PlayerController
Anim Mech (Animator)
Movespeed 3
Readyto Drill
Touching Crystal

SendMessage cannot be called during Awake, CheckConsistency, or OnValidate


Unity 2017.3.1f1 Personal (64bit) - level 1.unity - CrystalCoveTeam - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component FMOD Mobile Input Window Help
Center Global
Collab Account Layers Layout
Scene Asset Store Animator
Layers Parameters
Base Layer
Auto Live Link
Game Console Animation
Preview
Dopesheet Curves
Hierarchy
Project
Inspector

isDrilling
isWalking
isWalkingRight
isWalkingLeft
isWalkingBack
Shoot

Animation/Mecha/MechaAnimation/MechAnim.controller

mixamo.com (Read-Only) Samples 30
0:00 5:00
mixamorig:Hips : Position
mixamorig:Hips : Rotation
mixamorig:LeftUpLeg : Rotation
mixamorig:LeftUpLeg : Scale
mixamorig:LeftLeg : Rotation
mixamorig:LeftLeg : Scale
mixamorig:LeftFoot : Rotation
mixamorig:LeftFoot : Scale
mixamorig:LeftToeBase : Rotation
mixamorig:LeftToeBase : Scale
mixamorig:RightUpLeg : Rotation
mixamorig:RightUpLeg : Scale

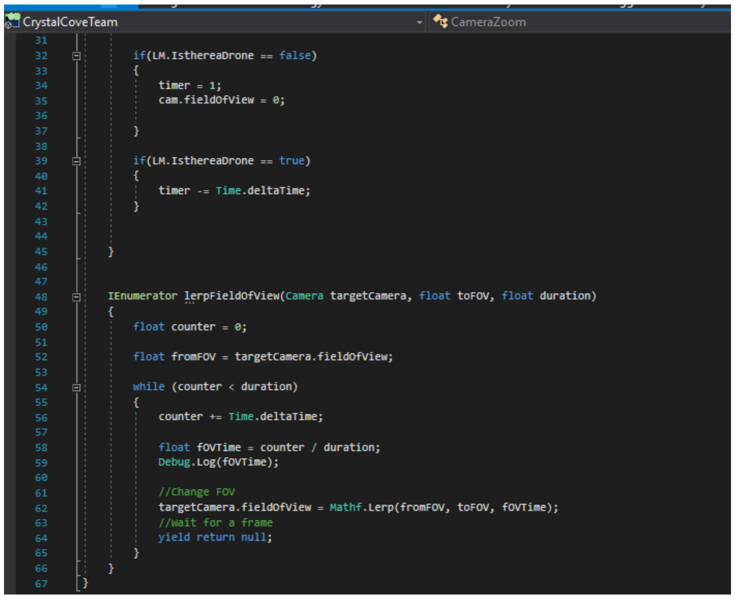
level 1
PassiveRockMonster
Drone
LevelManager
Positions
Environment
Mech
Water4SimpleReflectionSceneCamera
Player
EventSystem
Water4SimpleReflectionMain Camera
AudioManager

Assets ▸ Scripts
Audio, Aura, Billy's Sc, Editor, Editor De, Fonts, Gizmos, Material, Materials, Meshes, Models, Plugins, Resonan, Resource, Scripts
Drill, DrillEnergy, EnemyCha..., Hologram

Mech Static
Tag Untagged Layer Default
Prefab Select Revert Apply
Transform
Position X 17.19 Y 3.21 Z 1.9
Rotation X 0 Y 0 Z 0
Scale X 26.886 Y 26.886 Z 26.886
Animator
Controller MechAnim
Avatar IdleAvatar
Apply Root Motion
Update Mode Normal
Culling Mode Cull Update Transform
Clip Count: 0
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0
Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 0 Constant: 0 (0.0%)
Dense: 0 (0.0%) Stream: 0 (0.0%)
Rigidbody
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity ✓
Is Kinematic
Interpolate None
Collision Detection Discrete
Constraints
Player Controller (Script)
Script PlayerController
Anim Mech (Animator)
Movespeed 3
Readyto Drill
Touching Crystal

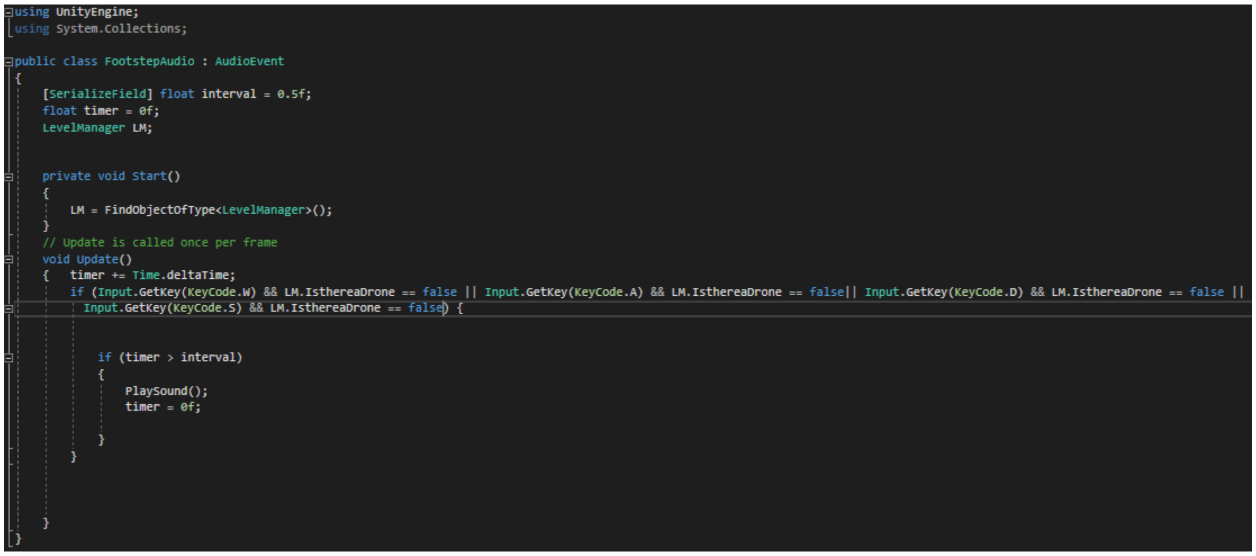
SendMessage cannot be called during Awake, CheckConsistency, or OnValidate

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.



```
            if(LM.IsthereaDrone == false)
            {
                timer = 1;
                cam.fieldOfView = 0;

            }

            if(LM.IsthereaDrone == true)
            {
                timer -= Time.deltaTime;
            }


        }


        IEnumerator lerpFieldOfView(Camera targetCamera, float toFOV, float duration)
        {
            float counter = 0;

            float fromFOV = targetCamera.fieldOfView;

            while (counter < duration)
            {
                counter += Time.deltaTime;

                float fOVTime = counter / duration;
                Debug.Log(fOVTime);

                //Change FOV
                targetCamera.fieldOfView = Mathf.Lerp(fromFOV, toFOV, fOVTime);
                //Wait for a frame
                yield return null;
            }
        }
    }
```

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.



```
using UnityEngine;
using System.Collections;

public class FootstepAudio : AudioEvent
{
    [SerializeField] float interval = 0.5f;
    float timer = 0f;
    LevelManager LM;


    private void Start()
    {
        LM = FindObjectOfType<LevelManager>();
    }
    // Update is called once per frame
    void Update()
    {   timer += Time.deltaTime;
        if (Input.GetKey(KeyCode.W) && LM.IsthereaDrone == false || Input.GetKey(KeyCode.A) && LM.IsthereaDrone == false|| Input.GetKey(KeyCode.D) && LM.IsthereaDrone == false ||
          Input.GetKey(KeyCode.S) && LM.IsthereaDrone == false) {


            if (timer > interval)
            {
                PlaySound();
                timer = 0f;

            }
        }



    }
}
```

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.

POWERPOINT ON THE NEXT PAGE:

Microsoft
PowerPoint Presenta