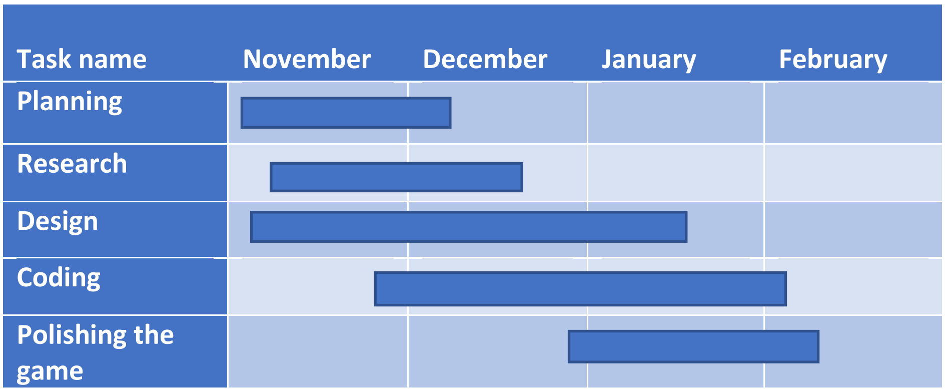
Game Engines

Task 1

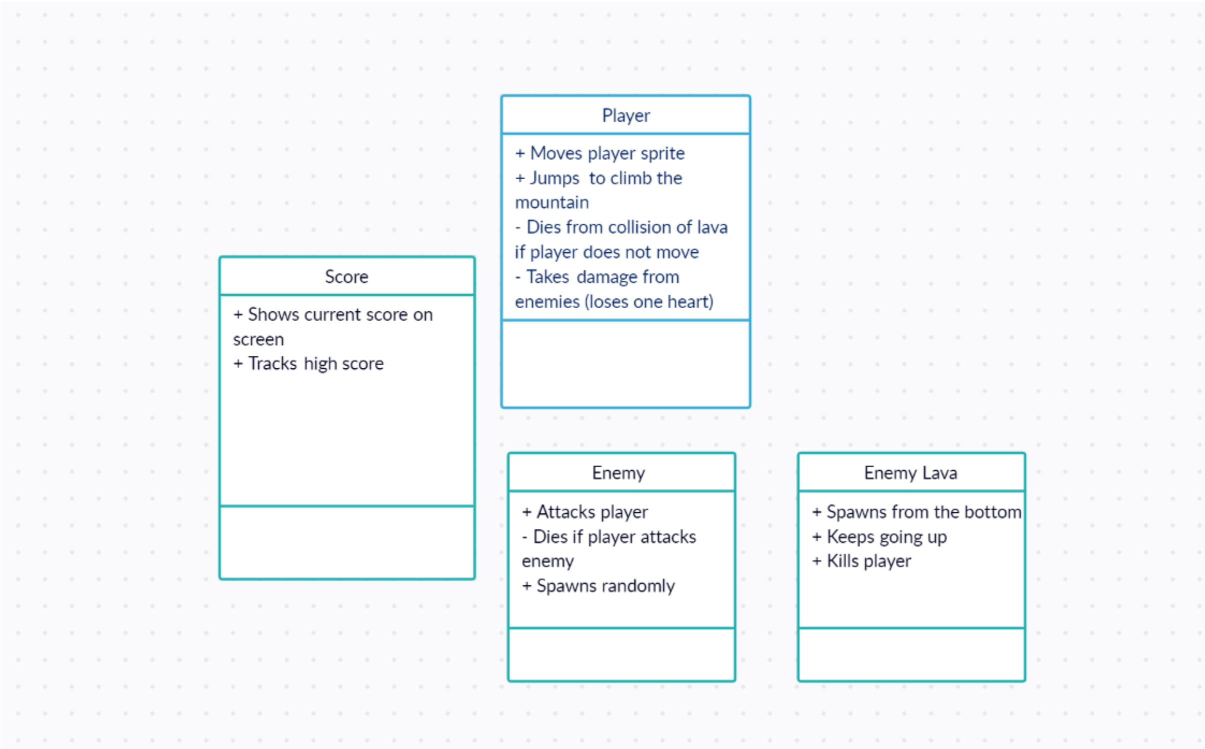
Initially, we had thought of the idea that the game would be a small dinosaur hoping his way through the world. But it was far too boring to do, as the game felt like a reskin of a typical runner game. at this time, we maintained the original design of our character and incorporated some adjusts to the game to what we have now.

With a name like panic, it is not left to up too much imagination to what you have to do, the main point of the game is to run away from the lava and in this game, you play the character named Pan (because the Greek god Pan is the representation of panic and our title is called 'Panic! ...now with lava') the small stegosaurus that is forced to running up the mountain to avoid enemies and the lava below. Here are some of the concepts that we worked on to create Pan. And here as of now is the final concept of the stegosaurus.

Time Schedule using the Gantt Chart.



CRC Cards



Player

+ Moves player sprite
+ Jumps to climb the mountain
- Dies from collision of lava if player does not move
- Takes damage from enemies (loses one heart)

Score

+ Shows current score on screen
+ Tracks high score

Enemy

+ Attacks player
- Dies if player attacks enemy
+ Spawns randomly

Enemy Lava

+ Spawns from the bottom
+ Keeps going up
+ Kills player

After we looked into the code that we would be using when making the game;

Character Movement (Controller)

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

public class Example : MonoBehaviour
{
    private CharacterController controller;
    private Vector3 playerVelocity;
    private bool groundedPlayer;
    private float playerSpeed = 2.0f;
    private float jumpHeight = 1.0f;
    private float gravityValue = -9.81f;

    private void Start()
    {
        controller = gameObject.AddComponent<CharacterController>();
    }

    void Update()
    {
        groundedPlayer = controller.isGrounded;
        if (groundedPlayer && playerVelocity.y < 0)
        {
            playerVelocity.y = 0f;
        }

        Vector3 move = new Vector3(Input.GetAxis("Horizontal"), 0,
Input.GetAxis("Vertical"));
        controller.Move(move * Time.deltaTime * playerSpeed);

        if (move != Vector3.zero)
        {
            gameObject.transform.forward = move;
        }

        // Changes the height position of the player..
        if (Input.GetButtonDown("Jump") && groundedPlayer)
        {
            playerVelocity.y += Mathf.Sqrt(jumpHeight * -3.0f * gravityValue);
        }

        playerVelocity.y += gravityValue * Time.deltaTime;
        controller.Move(playerVelocity * Time.deltaTime);
    }
}
```

(Technologies, 2020)

Parallax scrolling

```
using UnityEngine;


/// <summary>
/// Parallax scrolling script that should be assigned to a layer
/// </summary>
public class ScrollingScript : MonoBehaviour
{
  /// <summary>
  /// Scrolling speed
  /// </summary>
  public Vector2 speed = new Vector2(2, 2);

  /// <summary>
  /// Moving direction
  /// </summary>
  public Vector2 direction = new Vector2(-1, 0);

  /// <summary>
  /// Movement should be applied to camera
  /// </summary>
  public bool isLinkedToCamera = false;

  void Update()
  {
    // Movement
    Vector3 movement = new Vector3(
      speed.x * direction.x,
      speed.y * direction.y,
      0);

    movement *= Time.deltaTime;
    transform.Translate(movement);

    // Move the camera
    if (isLinkedToCamera)
    {
      Camera.main.transform.Translate(movement);
    }
  }
}
```

(Parallax scrolling — Pixelnest Studio, 2013)

Animation

```
using UnityEngine;
using System.Collections;

public class ExampleClass : MonoBehaviour
{
    public Animation anim;

    void Start()
    {
        anim = GetComponent<Animation>();
        foreach (AnimationState state in anim)
        {
            state.speed = 0.5F;
        }
    }
}
```

(Script to animation and movement of 2d player - Unity Answers, 2016)

POWER UPS in Unity

```
Yes 1,3,2. Is the correct order = https://www.youtube.com/watch?v=CLSiRf_OrBk
simple Examples: Public float health = 100f
PART 1
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class player : MonoBehaviour
{

    void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("player"))
        {
            Pickup();
        }
    }

    void Pickup()
    {
        Debug.log ("Power up pick")
    }
}
```

PART 3

```
public class player : MonoBehaviour
{
    public gameObject pickupEffect;
    void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("player"))
        {
            Pickup();
        }
    }

    void Pickup()
    {
        // spawn a cool effect
        Instantiate(pickupEffect, transform.position, transform.rotation);
        // apply effect to the player

        // remove power up object
        Destroy(gameObject);
    }
}
```

PART 2

```
public class player : MonoBehaviour
{
    public gameObject pickupEffect;
public float multiplier = 1.4f;
public float duration = 4f;

    void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("player"))
        {
            Pickup(other);
        }
    }

    void Pickup(Collider player)
    {
        // spawn a cool effect
        Instantiate(pickupEffect, transform.position, transform.rotation);
        // apply effect to the player
```

Health up

```
// apply effect to the player
PlayerStats stats = player.GetComponent<PlayerStats>();
                           stats.health *= multiplier
```

Size change

```
// apply effect to the player
                          player.transform.localScale *= multiplier

        // remove power up object
        Destroy(gameObject);
    }
}
```

COMPLETE CODE

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

public class player : MonoBehaviour
{

    public gameObject pickupEffect;
    public float multiplier = 1.4f;
    public float duration = 4f;
    void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("player"))
        {
            StartCoroutine( Pickup(other) );
        }
    }

    IEnumerator Pickup(Collider player)
    {
        // spawn a cool effect
        Instantiate(pickupEffect, transform.position, transform.rotation);
        // apply effect to the player
        PlayerStats stats = player.GetComponent<PlayerStats>();
        stats.health * multiplier

        GetComponent<MeshRenderer>().enabled = false;
        GetComponent<collider>().enabled = false;

        // wait x amount of seconds
        yield return new WaitForSeconds>(duration);
        // reverse the effect on player
        stats.health /= multiplier;

        // remove power up object
        Destroy(gameObject);
```

(Brackeys, 2017)

Health Bar in Unity

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

public class Health : MonoBehaviour
{
    public int curHealth = 0;
    public int maxHealth = 100;

    public HealthBar healthBar;

    // Start is called before the first frame update
    void Start()
    {
        curHealth = maxHealth;
    }

    // Update is called once per frame
    void Update()
    {
        if( Input.GetKeyDown( KeyCode.Space ) )
        {
            DamagePlayer(10);
        }
    }
    public void DamagePlayer( int damage )
    {
        curHealth -= damage;
        healthBar.SetHealth( curHealth );
    }
}
```

(Polidario, 2020)

VFXSpawnerState spawnCount

```
state.spawnCount += currentRate * state.deltaTime
```

(Technologies, 2020)

HIGH SCORE

```
using UnityEngine;
using UnityEngine.UI;

public class dice : MonoBehaviour{
    public Text score;
    public Text highScore;

    viod Start()
    {
        highScore.text = PlayerPrefs.GetInt("HighScore",0).ToString();
    }

    public void RollDice ()
    {
        int number = Random.Range(1,7);
        score.text = number.ToString();

        if (number > PlayerPrefs.GetInt("HighScore",0))
        {
            PlayerPrefs.SetInt("HighScore", number);
            highScore.text = number.ToString()
        }
        PlayerPrefs.SetInt("HighScore", number);
    }


    public viod reset ()
    {
        PlayerPrefs.DeletaKey("HighScore");
        highScore.text = "0";
    }


}
```

(Brackeys, 2017)

Bibliography:

- Answers.unity.com. 2016. Script To Animation And Movement Of 2D Player - Unity Answers. [online] Available at: <https://answers.unity.com/questions/1235653/how-i-do-fix-the-script-for-2d-player-movement.html> [Accessed 12 December 2020].
- Brackeys, 2017. How To Make A HIGH SCORE In Unity. [online] Youtube.com. Available at: <https://www.youtube.com/watch?v=vZU51tbgMXk&list=PLopDeIkRdGINk1ZQ1Lo6Mz8S0djilTIqp&index=3> [Accessed 12 December 2020].
- Brackeys, 2017. POWER UPS In Unity. [online] Youtube.com. Available at: <https://www.youtube.com/watch?v=CLSiRf_OrBk> [Accessed 12 December 2020].
- Code Monkey, 2019. [online] Youtube.com. Available at: <https://www.youtube.com/channel/UCFK6NCbuCIVzA6Yj1G_ZqCg> [Accessed 12 December 2020].
- Pixelnest Studio. 2013. Parallax Scrolling — Pixelnest Studio. [online] Available at: <https://pixelnest.io/tutorials/2d-game-unity/parallax-scrolling/> [Accessed 12 December 2020].
- Polidario, B., 2020. Unity Tutorials: How To Make A Health Bar In Unity! - Weeklyhow. [online] WeeklyHow. Available at: <https://weeklyhow.com/how-to-make-a-health-bar-in-unity/> [Accessed 12 December 2020].
- Technologies, U., 2020. Unity - Scripting API: Charactercontroller.Move. [online] Docs.unity3d.com. Available at: <https://docs.unity3d.com/ScriptReference/CharacterController.Move.html> [Accessed 12 December 2020].
- Technologies, U., 2020. Unity - Scripting API: VFX.Vfxspawnerstate.Spawncount. [online] Docs.unity3d.com. Available at: <https://docs.unity3d.com/ScriptReference/VFX.VFXSpawnerState-spawnCount.html> [Accessed 12 December 2020].