

Dokumentasi Game Kalimantan Wildlife

1.1 Ide

Pada penelitian ini dibangun sebuah Game dengan genre Platformer dimana pemain melewati rintangan dan mengalahkan musuh yang ada. Latar belakang dari game ini yaitu untuk pengenalan flora dan fauna kalimantan Indonesia. Konsep dari permainan ini yaitu pemain akan mengendalikan sebuah karakter dimana untuk beralih ke level berikutnya pemain harus mencari jalan dan mengalahkan musuh yang ada. Untuk batasan umur game ini dapat dimainkan di segala kalangan dengan tidak memperlihatkan adegan kekerasan dan efek yang tidak wajar seperti darah. Game akan memiliki 6 level, dimana ada 2 level yang berbeda yaitu level khusus bonus, dan level khusus melawan musuh terakhir.

Game ini dibangun dengan menggunakan game engine Unity dengan target perangkatnya yaitu perangkat smartphone android. Adapun beberapa fitur yang terdapat pada game ini yaitu :

1. Game Single Player.
2. Game dapat menyimpan progress pemain
3. Game memiliki kontrol sentuh.
4. Game memiliki fitur galeri dimana terdapat info flora dan fauna Kalimantan Indonesia.
5. Pemain dapat mengatur volume suara pada game.

1.2 Produksi

Pada tahap ini akan dilakukan tahap perancangan awal game, seperti membuat cerita, aturan game, menentukan software yang akan digunakan, membuat dan mencari aset, membuat desain level, tampilan game, item dalam game, dan tombol dalam game.

1. Draft story dari game ini yaitu sebagai berikut :

Story Game Kalimantan WildLife

Karakter utama yaitu seorang polisi hutan bernama Agus, diperintahkan atasannya yang bernama “Pak Fery” untuk mengatasi masalah penebang hutan liar yang ada.

Agus kemudian diberi 3 perintah utama oleh pak Fery yaitu, hancurkan mesin kayu, hancurkan bulldozer, dan kalahkan pimpinan penebang hutan liar.

Setelah itu Agus pun mulai melakukan tugasnya yang pertama yaitu menghancurkan mesin kayu, sembari di perjalanannya Agus melihat hewan seperti burung enggang, dan pesut. Lalu Agus melihat mesin kayu dan menghancurkannya.

Setelah menghancurkan mesin kayu, Agus melanjutkan tugasnya yaitu mencari bulldozer dan menghancurkan. Lalu Agus menemukan mesin Bulldozer dan kemudian menghancurkannya.

Akhirnya tugas Agus samapai kepada yang terakhir yaitu mengalahkan pimpinan penebang kayu, lalu Agus bertemu pimpinan penebang liar yang bernama Jono, dan Agus pun mengalahkannya. Akhirnya hutan pun terhindar dari ancaman penebang liar dan selesai.

2. Peraturan game :

- a) Jika HP pemain habis maka permainan berakhir.
- b) Jika pemain jatuh ke air (tidak termasuk di level bonus) maka permainan berakhir.
- c) Pemain dapat menuju ke level berikutnya ketika pemain mengalahkan boss atau mencapai titik selesai level.
- d) Pemain akan mendapatkan 1 medkit guna untuk mengisi HP apabila pemain mengumpulkan 10 koin.

3. Software yang akan digunakan :

1. Game Engine yang akan digunakan yaitu Unity 2020.01.b8.
2. Software gambar yang akan digunakan sebagai pembuatan asset yaitu Aseprite.

4. Asset gambar :

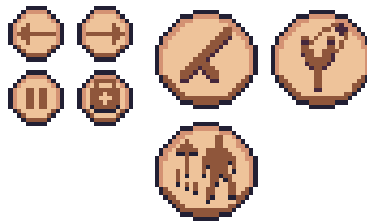
a) TileSet

Asset ini merupakan asset yang akan digunakan untuk pembuatan level yang berada di game ini.



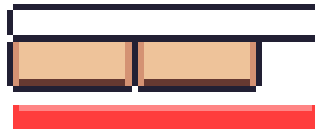
b) Control UI

Asset ini merupakan asset yang akan digunakan untuk pembuatan tombol kontrol UI pemain.



c) Status UI

Asset ini merupakan asset yang akan digunakan untuk pembuatan status UI player dan musuh di game ini seperti UI health pemain dan boss.



d) Menu UI

Asset ini merupakan asset yang akan digunakan untuk pembuatan UI menu utama pada game ini.



e) Gallery Icon UI

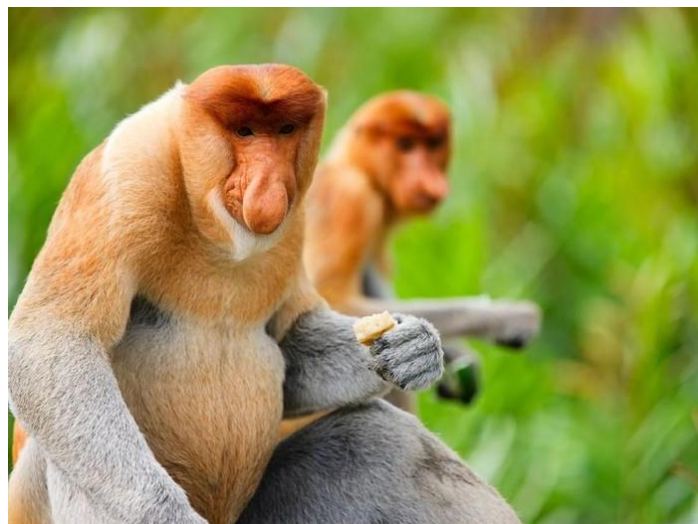
Asset ini merupakan asset yang akan digunakan untuk pembuatan Menu Gallery pada game ini.



Gambar Pesut yang digunakan untuk deskripsi galeri.



Gambar Burung enggang yang digunakan untuk deskripsi galeri.



Gambar Bekantan yang digunakan untuk deskripsi galeri.



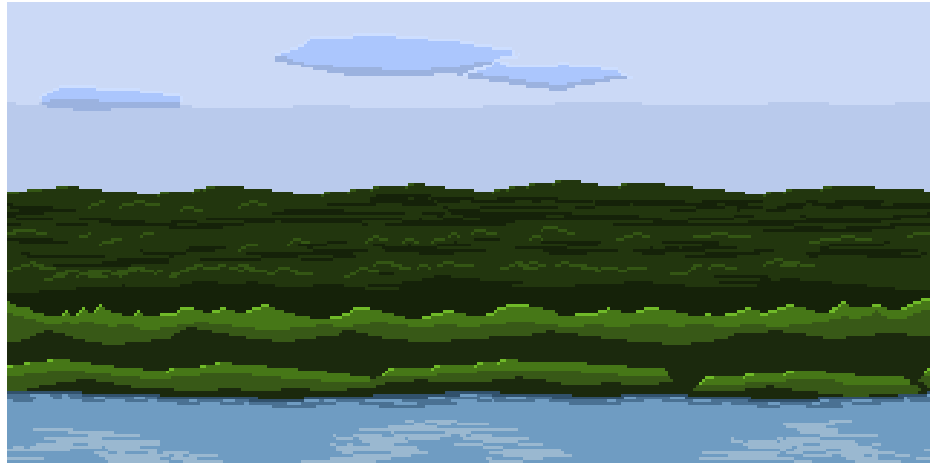
Gambar Anggrek Hitam Kalimantan yang digunakan untuk deskripsi galeri.



Gambar pohon ulin yang digunakan untuk deskripsi galeri.

f) Title UI

Asset ini merupakan asset yang akan digunakan untuk pembuatan title dan background menu pada game ini.

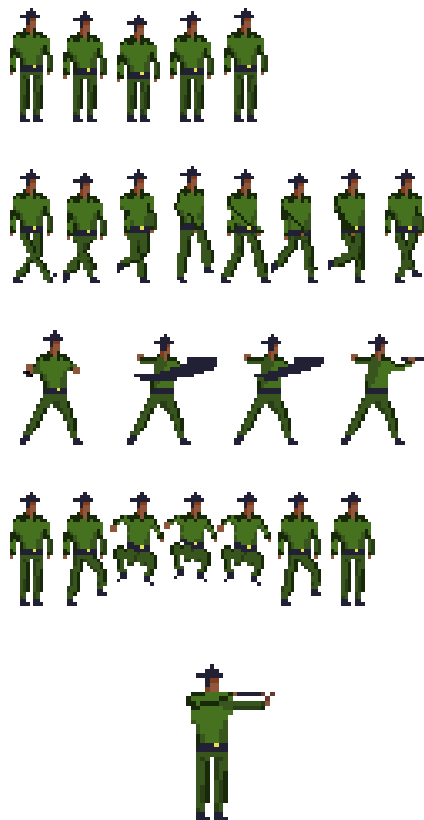


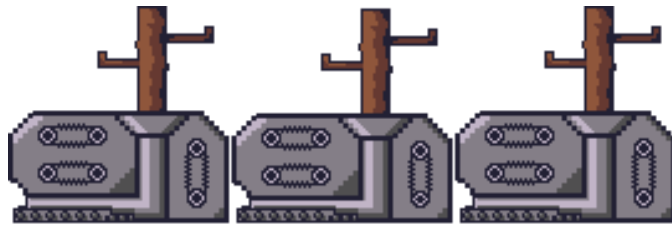
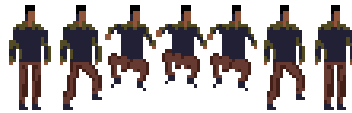
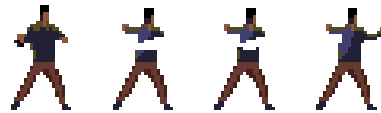
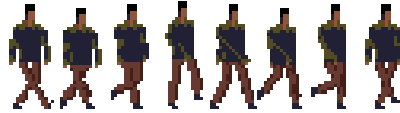
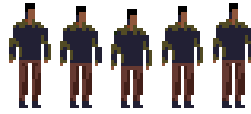
**KALIMANTAN
WILDLIFE**

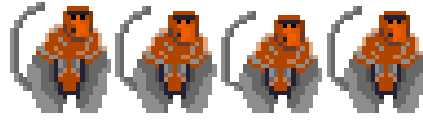


g) Animation spritesheet

Asset ini merupakan asset yang akan digunakan untuk pembuatan animasi karakter, musuh, dan entitas yang berada di game ini.

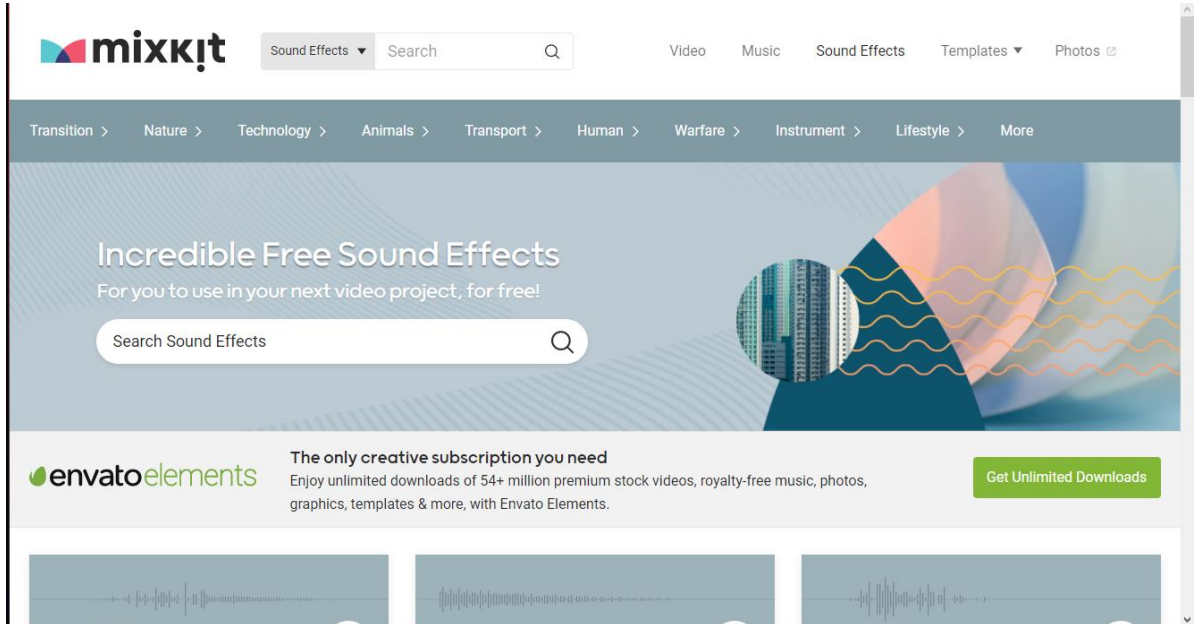




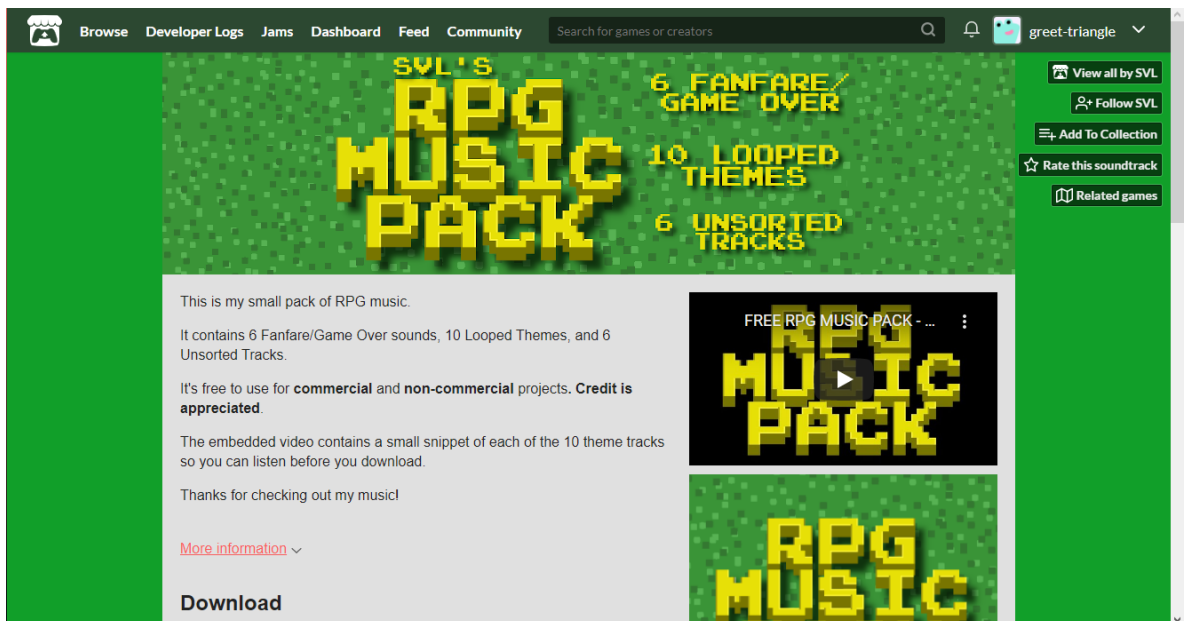


5. Asset audio diperoleh dari :

1. Mixkit <https://mixkit.co/free-sound-effects/>



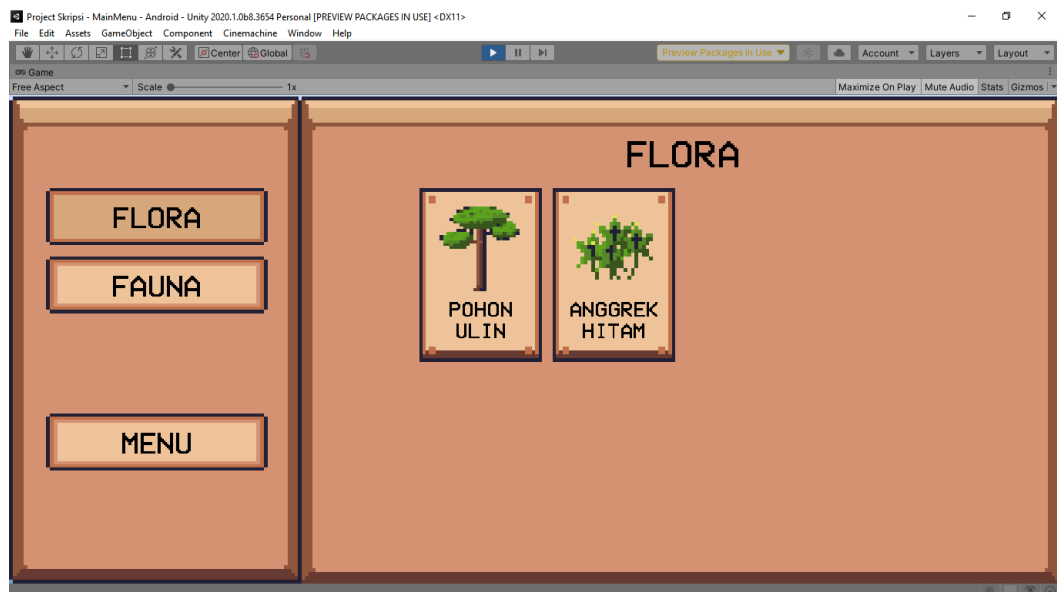
2. RPG music pack <https://svl.itch.io/rpg-music-pack-svl>



6. Tampilan desain game :



Pada main menu akan menampilkan beberapa pilihan seperti play untuk memainkan game dari awal, continue untuk melanjutkan progres permainan, gallery untuk menampilkan menu gallery, settings untuk menampilkan menu konfigurasi, dan exit untuk menutup aplikasi game.

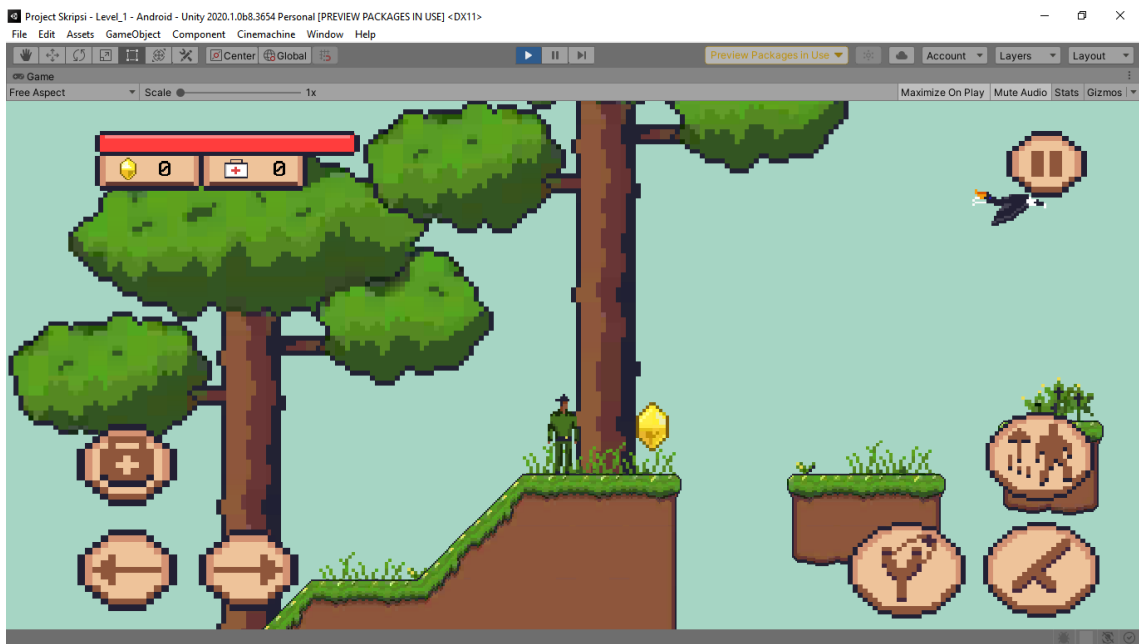




Pada menu gallery pemain dapat melihat informasi flora dan fauna yang berada di game ini seperti deskripsi dan gambar asli dari flora dan fauna tersebut.



Pada menu settings pemain dapat mengatur volume dan suara yang ada di dalam game.



Tampilan gameplay permainan yang terdapat beberapa UI seperti informasi HP pemain, koin dan medkit, dan beberapa UI kontrol Mobile player seperti tombol gerak ke kanan dan ke kiri, tombol melompat, tombol ketapel, tombol medkit, tombol pause, dan tombol tongkat untuk memukul.



Tampilan pause pada saat pemain menekan tombol pause.

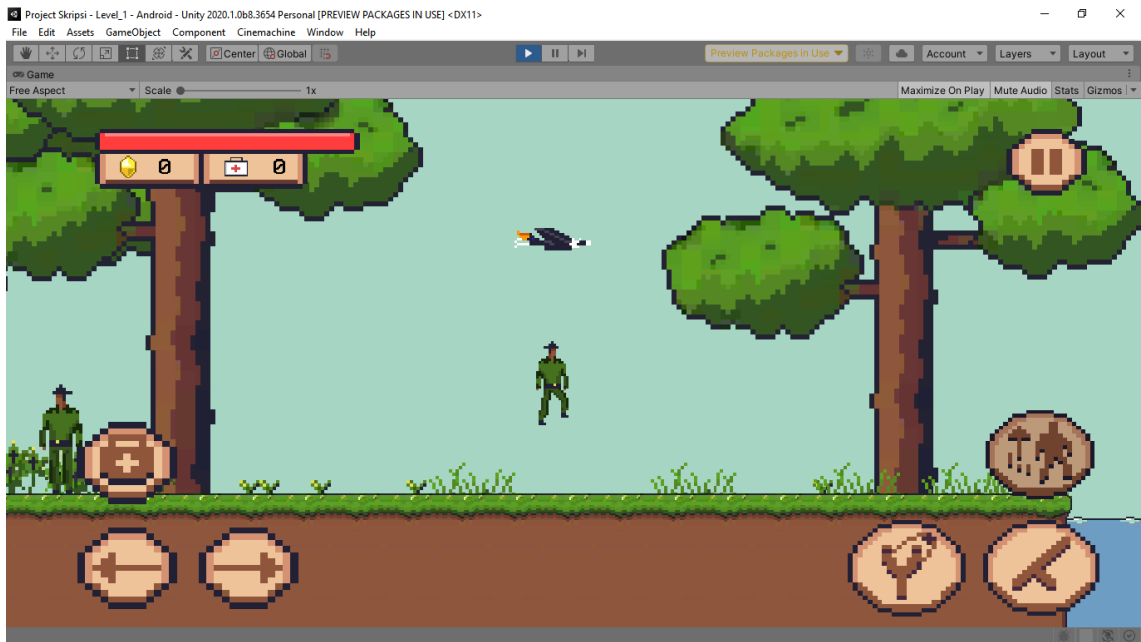


Tampilan gameover pada saat HP pemain menyentuh 0 atau pada saat pemain jatuh ke air.

7. Item dalam game :

1. Pembuatan Sistem Game :







Perilaku dasar karakter yang bisa dilakukan oleh pemain yaitu memukul, melompat, menggunakan ketapel, bergerak ke kanan dan ke kiri, menggunakan ketapel, darah berkurang, menyimpan data karakter di buat dalam 3 script yaitu PlayerBehaviour, PlayerProjectileBehaviour, dan PlayerScriptable.

```

using System.Collections;
using UnityEngine;
using TMPro;

public class PlayerBehaviour : MonoBehaviour
{
    // Variabel GameManager
    private GameManager gameManager;

    // Variabel Data
    [Header("PlayerData")]
    public PlayerScriptable playerData;
    public SoundManager soundManager;

    // Variabel Fisik
    [Header("Player Physic")]
    public Rigidbody2D rb;
    public float speed;
    public float jumpForce;
    private bool isMoving = false;
    private bool onceJump = true;
    public LayerMask layerGround;
    public GameObject groundCheck;
    private Animator animator;

    private bool isGrounded;
    [HideInInspector] public bool facingRight;

    // Variabel Attack
    [Header("Player Melee Attack")]

```

```

    // Variabel Attack
    [Header("Player Melee Attack")]

    [SerializeField] private float attackInterval;
    private float attackTime;
    private bool onceAttack = true;
    private bool isAttacking = false;
    private float attackAnimationTime;
    [SerializeField] private Transform attackPoint;
    [SerializeField] [Range(0.1f, 0.5f)] private float attackRadius;
    [SerializeField] private LayerMask enemyLayerMask;

    // Variabel Ranged Attack
    [Header("Player Ranged Attack")]

    [SerializeField] private float rangedAttackInterval;
    private bool onceRanged = true;
    private bool isRangedAttacking;
    private float rangedAttackTime;
    private float rangedAttackAnimationTime;
    public GameObject projectile;
    [SerializeField] private Transform rangedAttackPoint;

    // Variabel runtime data
    [HideInInspector] public int health;
    [HideInInspector] public int coin = 0;
    [HideInInspector] public int med = 0;
    [HideInInspector] public bool die = false;

    // variabel Damaged
    [Header("Player Color Data")]

```

```

[SerializeField] private float attackInterval;
private float attackTime;
private bool onceAttack = true;
private bool isAttacking = false;
private float attackAnimationTime;
[SerializeField] private Transform attackPoint;
[SerializeField] [Range(0.1f, 0.5f)] private float attackRadius;
[SerializeField] private LayerMask enemyLayerMask;

// Variabel Ranged Attack
[Header("Player Ranged Attack")]

[SerializeField] private float rangedAttackInterval;
private bool onceRanged = true;
private bool isRangedAttacking;
private float rangedAttackTime;
private float rangedAttackAnimationTime;
public GameObject projectile;
[SerializeField] private Transform rangedAttackPoint;

// Variabel runtime data
[HideInInspector] public int health;
[HideInInspector] public int coin = 0;
[HideInInspector] public int med = 0;
[HideInInspector] public bool die = false;

// variabel Damaged
[Header("Player Color Data")]

public Renderer spriteRenderer;
public Color defaultSpriteColor;

```

```
public Color damagedSpriteColor;
// Variabel UI
[Header("Player UI")]

public HealthUIScript healthBar;
public GameObject healthBarBoss;
public TextMeshProUGUI coinCountUI;
public TextMeshProUGUI medCountGUI;

// Variabel UI mobile control
private float moveDir;
private bool hitAttackButton = false;
private bool hitRangedAttackButton = false;

[Header("Animator Effect")]
public Animator medAnimator;
```

```

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

public class PlayerProjectileBehaviour : MonoBehaviour
{
    private Rigidbody2D rb;
    public float projectileSpeed;
    [HideInInspector] public float xDirection = 0;
    [HideInInspector] public bool move = true;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        rb.AddForce(new Vector2(xDirection * projectileSpeed, 0));
    }

    private void Move(float xDirection)
    {
        rb.AddForce(new Vector2(xDirection * projectileSpeed, 0));
        move = false;
    }

    void OnCollisionEnter2D(Collision2D col)
    {
        if (col.gameObject.tag.Equals("Enemy"))
        {
            if(col.gameObject.GetComponent<EnemyBehaviour>() !=
null)

```

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

[CreateAssetMenu(fileName = "PlayerData", menuName = "PlayerData",
order = 1)]
public class PlayerScriptable : ScriptableObject
{
    public int health;
    public int med;
    public int coin;

    public int itemGalleryUnlocked;
    public List<GalleryScriptable> galleryData;
    public int indexRespawn;

    [Header("Sound Settings")]
    [Range(0, 1f)] public float soundVolume;
    public int toggleSound;
}
```




Pada game ini cerita akan digunakan dalam sistem dialog, adapun sistem dialog di game ini memiliki beberapa bagian yaitu avatar pembicara, text dialog, nama pembicara, kontrol dialog dan trigger dialog. Sistem dialog ini dibuat dengan menggunakan 3 script yaitu DialogScriptable, DialogManager, dan TriggerDialog.

```
using UnityEngine;

[CreateAssetMenu(fileName = "DialogData", menuName = "DialogData",
order = 3)]
public class DialogScriptable : ScriptableObject
{
    [Header("Dialog Data")]

    public bool startDialog;

    public string[] dialogAvatarName;

    public Sprite[] dialogAvatarSprite;

    [TextArea]
    public string[] dialogText;
}
```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class DialogManager : MonoBehaviour
{
    [Header("Helper")]
    private int indexDialogData;
    public Animator dialogAnimator;
    private GameManager gameManager;

    [Header("Dialog Data")]
    public DialogScriptable[] dialogData;

    [Header("Dialog Helper Object")]
    public Image avatarSpriteUI;
    public TextMeshProUGUI avatarnameTextUI;
    public TextMeshProUGUI dialogTextUI;

    private bool startDialog;
    private Queue<Sprite> avatarSprite;
    private Queue<string> avatarName;
    private Queue<string> dialogueText;

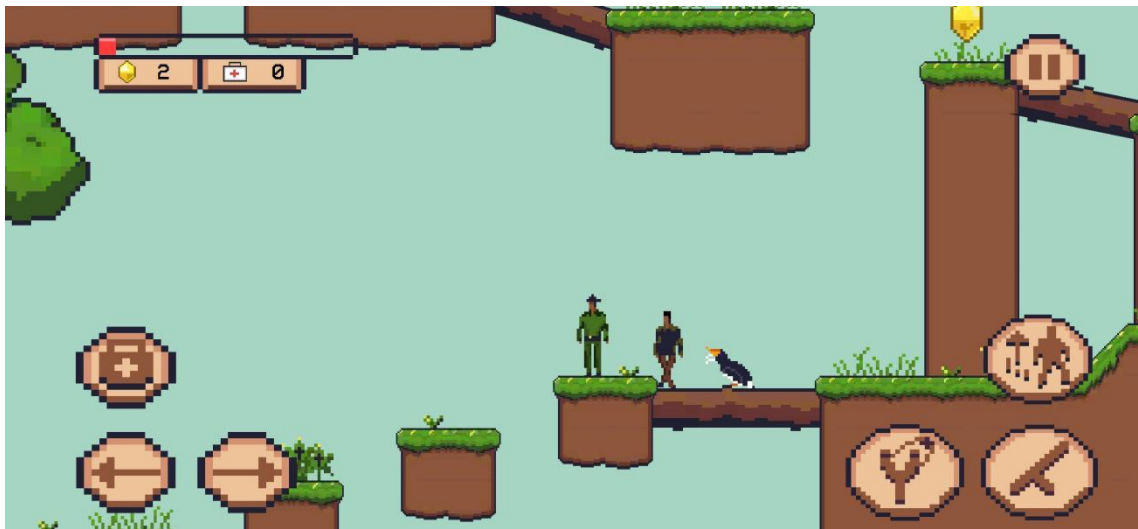
    [Header("Assign the value for the last level")]
    public SceneManagement sceneManagement;

```

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

public class TriggerDialog : MonoBehaviour
{
    private bool alreadyTriggerred = false;
    public string targetGameObjectName;
    public DialogManager dialogManager;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.name == targetGameObjectName &&
alreadyTriggerred == false)
        {
            dialogManager.GetComponent<DialogManager>().triggerNextDialog();
            alreadyTriggerred = true;
        }
    }
}
```



Game ini memiliki musuh yang dapat menyerang pemain dan dapat berjalan mendatangi pemain, untuk sistem musuh seperti ini dibuat menggunakan 4 script yaitu EnemyAttribute, EnemyBehaviour, EnemyMovement_1, PreventObjectOnPosition.

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

[CreateAssetMenu(fileName = "EnemyData", menuName = "EnemyData",
order = 2)]
public class EnemyAttribute : ScriptableObject
{
    [Header("Attribute")]
    public int health;
    [Range(0f, 5f)]
    public float movementSpeed;
    [Range(0.1f, 5f)]
    public float lookRadius;

    public bool isBoss;
}
```

```

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

public class EnemyBehaviour : MonoBehaviour
{
    [Header("Enemy Attribute")]
    public EnemyAttribute enemyAttribute;
    public SoundManager soundManager;
    public GameManager gameManager;

    // Attribute
    [HideInInspector] public bool die = false;
    private bool isBoss;
    [HideInInspector] public int health;
    public Renderer sprite;

    // Animation
    [Header("Animation")]
    public Animator animator;

    // UI
    [Header("Boss UI")]
    public HealthUIScript healthBar;
    public GameObject healthBarUI;

    // color data
    [Header("Enemy Color Data")]
    public Color defaultSpriteColor;
    public Color damagedSpriteColor;
}

```

```

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

public class EnemyMovement_1 : MonoBehaviour
{
    [Header("Enemy Attribute")]
    public SoundManager soundManager;
    public EnemyAttribute enemyAttribute;
    public LayerMask targetLayerMask;
    public EnemyBehaviour enemyBehaviour;

    // attribute
    private float speed;
    private float lookRadius;
    public Rigidbody2D rb;
    public Transform attackPoint;

    // player variabel
    public Transform playerTransform;

    // variabel
    [HideInInspector] public bool facingRight;
    [HideInInspector]
    public bool isHitted = false;

    // Animation
    [Header("Animation")]
    public Animator animator;

```




Untuk Boss yang ada di game ini memiliki keunikan dan perilaku masing-masing, dimana boss seperti mesin kayu tidak bisa bergerak namun dapat mengeluarkan kayu, boss bulldozer yang dapat bergerak cepat ke kanan dan ke kiri, dan boss terakhir Jono yang dapat memanggil bala bantuan. Untuk pembuatan boss diperlukan penggunaan script musuh dan beberapa tambahan script sebanyak 5 yaitu 2 script untuk boss mesin kayu, 2 script untuk boss bulldozer, dan 1 script untuk boss terakhir Jono.

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

public class ProjectileBehaviour : MonoBehaviour
{
    Rigidbody2D rb;
    Vector2 direction;

    public float speed;

    [HideInInspector]
    public bool canBePunched;
    [HideInInspector]
    public bool isPunched;

    // Start is called before the first frame update
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        isPunched = false;
    }
}
```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class BossBehaviour_1 : MonoBehaviour
{
    public EnemyBehaviour enemyBehaviour;

    [Header("Projectile")]
    public GameObject[] projectile;
    public Transform projectileSpawnPoint;
    private int projectileCount;

    [Header("Projectile Time Spawn")]
    public float[] timeIntervalProjectile;
    private float time_1 = 0f;

    void Start()
    {
        projectileCount = 0;
    }

    void Update()
    {
        SpawnProjectile();
    }

    void SpawnProjectile()
    {
        time_1 += Time.deltaTime;
    }
}

```

```

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

public class Boss2Attack : MonoBehaviour
{
    public Boss2Movement boss2Movement;

    private void OnCollisionStay2D(Collision2D col)
    {
        if (col.gameObject.tag == "Player")
        {
            GameObject player = col.gameObject;

            if (player.GetComponent<PlayerBehaviour>().die == false)
            {
                player.GetComponent<PlayerBehaviour>().TakeDamage(10, 0);
                player.GetComponent<Rigidbody2D>().AddForce(new
                Vector2(0f, 100));
            }
            else
            {
                Physics2D.IgnoreCollision(GetComponent<Collider2D>(), col.collider);
            }
        }
    }
}

```

```

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

public class Boss2Movement : MonoBehaviour
{
    [HideInInspector] public int direction = -1;
    public Transform playerTransform;
    public float movementSpeed = 5f;
    private float timerRotate = 0f;
    public float timeRotate = 3f;

    public EnemyBehaviour attribute;

    public BossZone bossZone;
    private bool facingRight = false;
    private bool phaseTwo = false;
    private bool phaseThree = false;

    public GameObject colliderArena2;

    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (attribute.health < 50)

```

```

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

public class Boss3Movement : MonoBehaviour
{
    EnemyBehaviour enemyBehaviour;

    public Transform playerTransform;
    public int movementSpeed;
    public Animator animator;
    public GameObject attackPoint;
    public LayerMask targetLayer;
    public GameManager gameManager;
    public DialogManager dialogManager;

    public GameObject enemyPrefabs;
    public GameObject summonPoint;
    private bool summonPhaseOne = true;
    private bool summonPhaseTwo = true;

    public float attackInterval;
    private float attackTime = 0f;

    private bool isAttacking = false;
    private float direction = -1;
    private bool facingLeft = true;

    private void Awake()
    {

```



Pada game ini terdapat sistem platform yang dapat bergerak sebagai tantangan pemain yang dibuat menggunakan 2 script yaitu FallPlatform, dan PlatformScript.

```
using System.Collections;
using UnityEngine;

public class FallPlatform : MonoBehaviour
{
    private Rigidbody2D rb;

    public float timeFall;

    void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }
}
```

```
void OnCollisionEnter2D(Collision2D col)
{
    if(col.gameObject.name == "Player")
    {
        StartCoroutine(Fall(timeFall));
    }
}

IEnumerator Fall(float time)
{
    yield return new WaitForSeconds(timeFall);
    rb.constraints = RigidbodyConstraints2D.None;
    rb.constraints = RigidbodyConstraints2D.FreezePositionX |
RigidbodyConstraints2D.FreezeRotation;
    rb.gravityScale = 1;
}
}
```



```

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

public class PlatformScript : MonoBehaviour
{
    [Header("Platform Attribute")]
    public bool vertical;
    public float speed;
    public Transform position_1;
    public Transform position_2;

    private bool traverse = false;

    void Start()
    {
    }

    // Update is called once per frame
    void Update()
    {
        if (!traverse)
        {
            MovePlatform(1, vertical);
        }

        if (traverse)
        {
            MovePlatform(-1, vertical);
        }
    }
}

```

Untuk sistem next level, game over, pause, save data, pengaturan audio sound effect, gallery dan yang berkaitan dengan UI diperlukan 11 script yaitu 3 script untuk UI yang diantaranya yaitu GalleryMenu, HealthUiScript, UIControl, dan 8 script untuk game manager yang diantaranya yaitu AddToGallery, AudioAmbienceSettings, AudioSettingsUI, GalleryScriptable, GameManager, SaveManager, SceneManagement, dan SoundManager.

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

public class GalleryMenu : MonoBehaviour
{
    public PlayerScriptable playerData;

    public GameObject[] itemFlora;
    public GameObject[] itemFauna;

    [Header("None text UI")]
    public GameObject noneTextFlora;
    public GameObject noneTextFauna;

    [Header("Desc Item UI")]
    public GameObject descUI;
    public Image itemImage;
    public Image itemImage2;
    public TextMeshProUGUI itemName;
    public TextMeshProUGUI itemDesc;
```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class HealthUIScript : MonoBehaviour
{
    public Slider slider;
    public TextMeshProUGUI bossNameGUI;

    public void SetMaxHealth(int maxHealth)
    {
        slider.maxValue = maxHealth;
        slider.value = maxHealth;
    }

    public void SetHealth(int health)
    {
        slider.value = health;
    }

    public void SetBossName(string name)
    {
        if(bossNameGUI != null)
        {
            bossNameGUI.text = name;
        }
    }
}

```

```
using UnityEngine;

public class AddToGallery : MonoBehaviour
{
    public PlayerScriptable playerData;
    public int unlockGalleryItem;

    private void OnTriggerEnter2D(Collider2D col)
    {
        if (col.gameObject.tag == "Player"
            && playerData.itemGalleryUnlocked < unlockGalleryItem)
        {
            playerData.itemGalleryUnlocked = unlockGalleryItem;
        }
    }
}
```

```

using UnityEngine;

public class AudioAmbienceSettings : MonoBehaviour
{
    public PlayerScriptable playerData;
    public AudioSource[] audios;

    void Update()
    {
        if (playerData.toggleSound == 0)
        {
            SettingsAudios(0);
        }
        else
        {
            SettingsAudios(playerData.soundVolume);
        }
    }

    private void SettingsAudios(float volume)
    {
        foreach(AudioSource audio in audios)
        {
            audio.volume = volume;
        }
    }
}

```

```

using UnityEngine;
using UnityEngine.UI;

public class AudioSettingsUI : MonoBehaviour
{
    public AudioSource audioSource = null;
    public Slider slider = null;
    public Toggle toggle = null;
    public PlayerScriptable playerData;
    public SaveManager saveManager;

    private void Awake()
    {
        playerData = saveManager.LoadGame();

        slider.value = playerData.soundVolume;

        if (playerData.toggleSound == 1)
        {
            toggle.isOn = true;
        }
        else
        {
            toggle.isOn = false;
        }
    }

    public void AdjustVolumeUI()
    {

```

```
using UnityEngine;

[CreateAssetMenu(fileName = "GalleryData", menuName = "GalleryData",
order = 4)]
public class GalleryScriptable : ScriptableObject
{
    public string name;
    public Sprite sprite;
    [TextArea]
    public string description;

    [Header("1.Flora 2.Fauna")]
    [Range(1, 2)]
    public int type;

    public Sprite image;
}
```

```

using UnityEngine;

public class GameManager : MonoBehaviour
{
    private GameObject Player;

    [SerializeField]
    private GameObject gameOverPanel;

    public bool isPaused = false;
    public bool dialoguePopUp = false;
    [HideInInspector] public bool levelComplete = false;

    [SerializeField]
    private GameObject pausePanel;

    void Awake()
    {
        Player = GameObject.Find("Player");
    }

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            Pause();
        }

        if (Player == null)
        {

```



```

using UnityEngine;

public class SaveManager : MonoBehaviour
{
    public PlayerScriptable playerData;
    public AudioSource soundAudioSource;

    public void SaveGame(PlayerScriptable playerData)
    {
        PlayerPrefs.SetInt("indexRespawn", playerData.indexRespawn);
        PlayerPrefs.SetInt("health", playerData.health);
        PlayerPrefs.SetInt("coin", playerData.coin);
        PlayerPrefs.SetInt("med", playerData.med);
        PlayerPrefs.SetFloat("soundVolume", playerData.soundVolume);
        PlayerPrefs.SetInt("toggleSound", playerData.toggleSound);
        PlayerPrefs.SetInt("itemGallery",
playerData.itemGalleryUnlocked);
        PlayerPrefs.Save();
    }

    public PlayerScriptable LoadGame()
    {
        if (CheckData())
        {
            playerData.indexRespawn =
PlayerPrefs.GetInt("indexRespawn");
            playerData.health = PlayerPrefs.GetInt("health");
            playerData.coin = PlayerPrefs.GetInt("coin");
            playerData.med = PlayerPrefs.GetInt("med");
            playerData.soundVolume =

```

```

using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;

public class SceneManagement : MonoBehaviour
{
    [SerializeField] private GameObject levelCompletePanel;
    [SerializeField] private GameObject newGamePropmtUI;

    public GameManager gameManager;
    public SaveManager saveManager;
    public SoundManager soundManager;
    public PlayerScriptable playerData;
    public Animator transitionAnimator;
    private bool once = true;

    [Header("Assign the value for the last level")]
    [SerializeField] private GameObject endGamePanel;

    private void Awake()
    {
        playerData = saveManager.LoadGame();
    }

    public void StartGame()
    {
        if(playerData.indexRespawn > 1)
        {
            newGamePropmtUI.SetActive(true);
        }
    }
}

```

```

using UnityEngine;

public class SoundManager : MonoBehaviour
{
    public AudioClip[] audiosMenuUI = null;
    public AudioClip[] audiosAction = null;
    [HideInInspector] public AudioSource audioSource;

    private void Awake()
    {
        audioSource = GetComponent<AudioSource>();
    }

    public void PlaySound(int indexSoundClip)
    {
        audioSource.clip = audiosMenuUI[indexSoundClip];
        audioSource.Play();
    }

    public void PlayActionSound(int indexSoundActionClip)
    {
        audioSource.clip = audiosAction[indexSoundActionClip];
        audioSource.Play();
    }
}

```



Untuk AI entity Enggang dan pesut mahakan di perlukan 2 script masing-masing yaitu enggang behaviour dan pesut behaviour

```

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

public class EnggangBehaviour : MonoBehaviour
{
    [Header("point")]
    public Transform[] point;

    [Header("Attribute")]
    public float speed;
    public float idleTime;
    public Animator animator;

    public int currentIndexPositionPoint;
    public bool reverse;
    private bool canRotate;
    private bool isIdle;
    private float time;
    private int indexToPoint;

    void Awake()
    {
        isIdle = true;
        time = 0f;

        if (reverse)
        {
            indexToPoint = -1;
            canRotate = true;
        }
    }
}

```

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

public class PesutBehaviour : MonoBehaviour
{
    [Header("point")]
    public Transform[] point;

    [Header("Attribute")]
    public float speed;
    public Animator animator;

    public bool reverse;
    public int currentIndexPositionPoint;

    private bool canRotate;

    void Awake()
    {
        reverse = true;
        currentIndexPositionPoint = point.Length - 1;
    }

    void Update()
    {
        Rotate();
        Move();
    }
}
```