

LAMPIRAN

Lampiran 1. Source Code

<https://github.com/daslbv/mathlandia>

1. Player Controller

```
using UnityEngine;
using System.Collections;
public class PlayerController : MonoBehaviour
{
    [Header("Movement Settings")]
    public float speed;
    public float jumpingPower;
    public float wallCheckDistance;
    private float defaultSpeed;
    private float horizontal;
    private float horizontal2;
    private bool isFacingRight = true;
    private bool isSprinting;
    private int jumpsRemaining;

    [Header("Components")]
    private Rigidbody2D rb;
    [SerializeField] private Transform groundCheck;
    [SerializeField] private LayerMask groundLayer;
    [SerializeField] private Transform wallCheck;
    [SerializeField] private LayerMask wallLayer;
    AudioSource footstepAudioSource;
    [SerializeField] private AudioClip[] footstepSounds;
    [SerializeField] FixedJoystick joystick;

    [Header("Script Reference")]
    private PlayerController playerController;
    private PlayerStatus playerStatus;

    [Header("Audio Clip")]
    public AudioClip jumpSound;
    public AudioClip landingSound;
    private Animator anim;
    private bool isPlayingFootstep;

    void Awake()
    {
        playerStatus = GetComponent<PlayerStatus>();
        playerController = GetComponent<PlayerController>();
    }
    private void Start()
    {
        defaultSpeed = speed;
        jumpsRemaining = 2;
        rb = GetComponent<Rigidbody2D>();
        anim = GetComponent<Animator>();
        footstepAudioSource = GetComponent<AudioSource>();
        isPlayingFootstep = false;
    }
}
```

2. Player Status

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SocialPlatforms.Impl;
public class PlayerStatus : MonoBehaviour
{
    [Header("Player Status")]
    public int currHealth;
    private int maxHealth = 1;
    public int totalQuiz;

    [Header("Player Component")]
    private Animator playerAnim;
    public Checkpoint checkpoint;
    [Header("Gameobject Reference")]
    [SerializeField] GameObject finishCanvas;
    [SerializeField] string finishLevel;

    [Header("Script Reference")]

    void Awake()
    {
        currHealth = maxHealth;
        playerAnim = GetComponent<Animator>();
    }

    public void Respawn()
    {
        if (currHealth == 0)
        {
            checkpoint.RespawnCheck();
            currHealth = 1;
        }
    }
}
```

3. Checkpoint

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class Checkpoint : MonoBehaviour
{
    [SerializeField] private AudioClip checkpointSound;
    private Transform currentCheckpoint;
    private int checkpointCounter = 0;

    [Header("Player Component")]
    public Animator playerAnim;
    public Transform playerPos;
    private Vector2 initialPos = new Vector2(-5.3f,
0.9650002f);

    [Header("Script Reference")]
    public PlayerController playerController;

    private void Awake()
    {

```

```

        checkpointCounter = 0;
        if (playerPos == null)
        {
            GameObject player =
GameObject.FindGameObjectWithTag("Player");
            if (player != null)
            {
                playerPos = player.transform;
                playerAnim =
player.GetComponent<Animator>();
                playerController =
player.GetComponent<PlayerController>();
            }
        }
    }
}

```

4. Save Manager

```

using System;
using System.IO;
using System.Runtime.Serialization.Formatters.Binary;
using UnityEngine;
using UnityEngine.Assertions.Must;
public class SaveManager : MonoBehaviour
{
    public static SaveManager instance { get; private set; }

    [Header("Timer")]
    public float timerString1;
    public float timerString2;
    public float timerString3;
    public float timerString4;
    public float timerString5;
    public float timerAccumulate;

    [Header("Wrong Answer Quiz")]
    public int totalWrong1;
    public int totalWrong2;
    public int totalWrong3;
    public int totalWrong4;
    public int totalWrong5;
    public int wrongAccumulate;

    [Header("Login System")]
    public string playerName;
    public string playerAbsen;

    [Header("Player Check")]
    public bool isLevel1Complete;
    public bool isLevel2Complete;
    public bool isLevel3Complete;
    public bool isLevel4Complete;
    public bool isLevel5Complete;
    public bool isLevelDone;
    public bool isLogin;

    private void Awake()
    {

```

```

        if (instance != null && instance != this)
            Destroy(gameObject);
        else
            instance = this;

        DontDestroyOnLoad(gameObject);
        Load();
    }
}

```

5. Level Manager

```

using UnityEngine;
using TMPro;
using UnityEngine.UI;
using UnityEngine.SceneManagement;
public class LevelManager : MonoBehaviour
{
    [Header("Score System Settings")]
    public TextMeshProUGUI scoreText;
    private int score;

    [Header("Paused Settings")]
    [SerializeField] GameObject pausedPanel;
    [SerializeField] Button pauseButton;
    [SerializeField] Button resumeButton;
    [SerializeField] Button quitButton;

    [Header("Level Settings")]
    [SerializeField] string nextLevelName;
    [SerializeField] GameObject finishCanvas;

    [Header("Script Reference")]
    [SerializeField] PlayerStatus playerStatus;

    [Header("Audio Reference")]
    [SerializeField] AudioClip clickSFX;
}

```

6. Audio Manager

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
public class AudioManager : MonoBehaviour
{
    public static AudioManager instance { get; private set; }

    private AudioSource soundSource;
    private AudioSource musicSource;

    private void Awake()
    {
        soundSource = GetComponent();
        musicSource =
transform.Find("MusicSource").GetComponent();
        if (musicSource == null)
        {

```

```

        Debug.LogError("No AudioSource found in children
named 'MusicSource'.");
    }
    if (instance == null)
    {
        instance = this;
        DontDestroyOnLoad(gameObject);
    }
    else if (instance != null && instance != this)
    {
        Destroy(gameObject);
    }

    if (!PlayerPrefs.HasKey("musicVolume"))
    {
        PlayerPrefs.SetFloat("musicVolume", 0.5f);
    }
    if (!PlayerPrefs.HasKey("soundVolume"))
    {
        PlayerPrefs.SetFloat("soundVolume", 0.5f);
    }
    ChangeMusicVolume(PlayerPrefs.GetFloat("musicVolume"));
    ChangeSoundVolume(PlayerPrefs.GetFloat("soundVolume"));
}
}

```

7. Home Manager

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class HomeManager : MonoBehaviour
{
    [Header("Sound Settings")]
    [SerializeField] GameObject soundPanel;
    [SerializeField] GameObject soundPanelBox;
    [SerializeField] Button soundButton;
    [SerializeField] Button closeSoundButton;

    [Header("Audio Reference")]
    [SerializeField] AudioClip bgmSong;
    [SerializeField] AudioClip startSFX;
    [SerializeField] AudioClip clickSFX;

    private void Awake()
    {
        InitialChecking();
    }

    void Start()
    {
        AudioManager.instance.PlayMusic(bgmSong, true);
    }
}

```

8. Quiz

```

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

```

```

using UnityEngine.UI;
public class Quiz : MonoBehaviour
{
    [Header("Script Reference")]
    public PlayerController playerController;

    [Header("Quiz Component")]
    public GameObject quizPanel;
    [SerializeField] private GameObject buttonInteract;
    [SerializeField] GameObject joystick;
    private bool playerInRange = false;

    [Header("Audio Reference")]
    [SerializeField] AudioClip clickSFX;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Player"))
        {
            if (buttonInteract != null)
            {
                buttonInteract.SetActive(true);
                LeanTween.scale(buttonInteract, new
Vector3(1, 1, 1), 1f).setEase(LeanTweenType.easeOutBack);
            }
            playerInRange = true;
        }
    }
}

```

9. Quiz System

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;
[System.Serializable]
public class QuizData
{
    public string question;
    public string answer;
}

public class QuizSystem : MonoBehaviour
{
    [Header("Quiz System")]
    public GameObject quizPanel;
    public GameObject quizTrigger;
    public GameObject rightAnswerImage;
    public GameObject wrongAnswerImage;
    [SerializeField] Button submitButton;
    public float textDuration = 1;
    public LeanTweenType easingType;

    [Header("Wrong Counter")]
    [SerializeField] int wrongCounter;
    [SerializeField] string wrongCounterLevel;
    [Header("Inputfield Settings")]

```

```

[SerializeField] TMP_InputField inputField;
[SerializeField] TextMeshProUGUI answerText

[Header("Script Reference")]
public PlayerController playerController;
public Quiz quizScript;
public Timer timer;

[Header("UI Settings")]
[SerializeField] GameObject joystick;

[Header("Audio Clip")]
[SerializeField] AudioClip rightAnswerClip;
[SerializeField] AudioClip wrongAnswerClip;

[Header("Quiz List")]
public List<QuizData> quizList = new List<QuizData>();

private QuizData currentQuiz;
private bool isButtonPressed = false;

void Start()
{
    SetRandomQuiz();
    TimerSet();
}

```

10. Timer

```

using UnityEngine;
using TMPro;
public class Timer : MonoBehaviour
{
    [Header("Timer Settings")]
    public float elapsedTime = 0f;
    public bool isRunning = false;
    public TextMeshProUGUI timerText;
    [SerializeField] string timerLevel;

    void Update()
    {
        if (isRunning)
        {
            elapsedTime += Time.deltaTime;
            UpdateTimerDisplay(elapsedTime);
            TimerSave();
        }
    }
    void UpdateTimerDisplay(float timeToDisplay)
    {
        int minutes = Mathf.FloorToInt(timeToDisplay / 60);
        int seconds = Mathf.FloorToInt(timeToDisplay % 60);
        timerText.text = string.Format("{0:00}:{1:00}",
minutes, seconds);
    }
}

```

11. Menu

```

using TMPro;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
public class Menu : MonoBehaviour
{
    [Header("Level Setting")]
    [SerializeField] Button level1Button;
    [SerializeField] Button level2Button;
    [SerializeField] Button level3Button;
    [SerializeField] Button level4Button;
    [SerializeField] Button level5Button;

    [Header("Login Settings")]
    [SerializeField] TMP_InputField namaInput;
    [SerializeField] TMP_InputField absenInput;

    [Header("Gameobject Canvas")]
    [SerializeField] GameObject levelCanvas;
    [SerializeField] GameObject loginCanvas;
    [SerializeField] GameObject kodeCanvas;
    [SerializeField] GameObject tinjauanCanvas;

    [Header("Display Tinjauan")]
    [SerializeField] TextMeshProUGUI playerNamaText;
    [SerializeField] TMP_InputField kodeInputField

    [Header("Audio Reference")]
    [SerializeField] AudioClip clickSFX;

    private bool sudahTampilkanKodeCanvas = false;

    private void Start()
    {
        SaveManager.instance.Load();
        level2Button.interactable = false;
        level3Button.interactable = false;
        level4Button.interactable = false;
        level5Button.interactable = false;

        OpenLevelCanvas();
        CekLevelUnlock();
        CekTampilkanKodeCanvas();
    }
}

```

12. Tinjauan

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.SceneManagement;
public class Tinjauan : MonoBehaviour
{
    [Header("Review 1-5")]
    [SerializeField] TextMeshProUGUI review1Text;

```

```

[SerializeField] TextMeshProUGUI review2Text;
[SerializeField] TextMeshProUGUI review3Text;
[SerializeField] TextMeshProUGUI review4Text;
[SerializeField] TextMeshProUGUI review5Text;

[Header("Timer Accumulate")]
[SerializeField] float timerAccumulate;
[SerializeField] TextMeshProUGUI timerAccumulateText;

[Header("Canvas Reference")]
[SerializeField] GameObject canvasMenu;
[SerializeField] GameObject canvasTinjauan;
[SerializeField] GameObject canvasLogin;

private void Start()
{
    timerAccumulate =
        SaveManager.instance.timerString1 +
        SaveManager.instance.timerString2 +
        SaveManager.instance.timerString3 +
        SaveManager.instance.timerString4 +
        SaveManager.instance.timerString5;
    ShowReview();
    TimerAccumulate();
}
}

```

Lampiran 2. Dokumentasi



Lampiran 3. Screenshot Hasil Game



rizhan maulana - 17
Total Waktu: 07:04

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
0 salah, 09 waktu	2 salah, 01:22 waktu	1 salah, 02:30 waktu	0 salah, 01:08 waktu

alfiansyah - 15
Total Waktu: 08:49

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
0 salah, 01:35 waktu	0 salah, 01:12 waktu	1 salah, 01:45 waktu	1 salah, 01:08 waktu

alfiansyah - 15
Total Waktu: 08:49

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
0 salah, 01:12 waktu	1 salah, 01:45 waktu	1 salah, 01:23 waktu	5 salah, 02:51 waktu

azka - 6
Total Waktu: 15:34

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
1 salah, 01:17 waktu	1 salah, 02:20 waktu	6 salah, 07:35 waktu	1 salah, 02:08 waktu

azka - 6
Total Waktu: 15:34

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
1 salah, 02:20 waktu	6 salah, 07:35 waktu	1 salah, 02:30 waktu	1 salah, 01:50 waktu

azki - 7

Total Waktu: 10:11

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
1 salah, 01:53 waktu	1 salah, 00:55 waktu	3 salah, 02:28 waktu	4 salah, 02:58 waktu

azki - 7

Total Waktu: 10:11

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
1 salah, 01:55 waktu	3 salah, 02:28 waktu	4 salah, 02:58 waktu	1 salah, 01:55 waktu

arsya - 14

Total Waktu: 10:59

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
1 salah, 01:34 waktu	2 salah, 02:23 waktu	0 salah, 02:41 waktu	1 salah, 01:34 waktu

arsya - 14

Total Waktu: 10:59

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
2 salah, 02:23 waktu	0 salah, 02:41 waktu	1 salah, 01:26 waktu	5 salah, 02:54 waktu

satria - 23

Total Waktu: 04:32

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
0 salah, 00:38 waktu	0 salah, 00:28 waktu	1 salah, 01:40 waktu	1 salah, 00:38 waktu

satria - 23

Total Waktu: 04:32

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
0 lah, :28 ktu	1 salah, 01:40 waktu	1 salah, 00:58 waktu	0 salah, 00:46 waktu

rayhan - 16

Total Waktu: 11:51

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
1 salah, 01:28 waktu	1 salah, 02:55 waktu	1 salah, 02:47 waktu	2 salah, 02:47 waktu

rayhan - 16

Total Waktu: 11:51

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
1 salah, 02:55 waktu	1 salah, 02:47 waktu	2 salah, 02:27 waktu	1 salah, 02:12 waktu

rangga - 10

Total Waktu: 15:17

LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
1 salah, 02:36 waktu	1 salah, 02:29 waktu	2 salah, 04:36 waktu	2 salah, 04:36 waktu

rangga - 10

Total Waktu: 15:17

LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
1 salah, 02:29 waktu	2 salah, 04:36 waktu	2 salah, 04:11 waktu	1 salah, 01:24 waktu

Lampiran 4. Data Nilai Siswa

NO	NAMA	SEMESTER 1																	
		LINGKUP MATERI 1				LINGKUP MATERI 2				LINGKUP MATERI 3				LINGKUP MATERI 4				LINGKUP MATERI 5	
		FM	SM	FM	FM	SM	FM	FM	SM	FM	FM	SM	FM	FM	SM	FM	FM	SM	
1	Zidan	80	83	75	80	100	90	80	85	100	85	100	82	80	100	80	82	100	
2	Algi	80	78	71	80	70	78	78	85	100	82	80	100	78	80	100	78	80	
3	Rizhan	72	80	72	72	70	70	70	75	75	71	72	78	75	78	75	78	78	
4	Alfiansyah	80	75	78	80	70	75	73	70	70	71	72	70	73	70	75	73	70	
5	Azka	80	80	75	80	72	80	84	85	100	83	85	100	90	90	77	82	82	
6	Azki	80	80	78	80	70	80	75	75	75	75	72	100	70	90	77	77	90	
7	Asya	80	82	75	80	74	82	70	70	100	77	78	78	80	100	78	80	100	
8	Satria	80	80	73	80	72	70	77	78	78	75	75	75	70	70	70	70	70	
9	Alfan	100	83	78	100	76	83	80	85	100	80	82	100	80	100	82	100	100	
10	Angele	80	80	70	80	84	80	78	80	80	70	70	70	72	80	100	80	100	
11	Malia	75	85	78	75	77	85	78	85	100	86	80	100	90	80	100	80	100	
12	Almiera	80	83	78	80	70	70	78	85	90	79	70	100	78	80	100	78	80	
13	Aqlia	90	87	80	90	85	78	78	85	100	84	74	100	78	80	100	80	100	
14	Fildan	80	81	77	80	74	81	77	75	75	80	100	72	79	80	100	78	80	
15	Aulia	80	85	90	80	83	85	78	85	100	85	80	100	78	80	100	78	100	
16	Rayhan	80	80	75	80	74	80	74	85	78	75	81	81	74	76	78	78	78	
17	Rangga	70	77	70	70	77	75	70	70	70	70	77	80	72	77	78	77	78	
Total		1367	1379	1293	1367	1293	1344	1303	1358	1491	1327	1316	1532	1308	1333	1516			

Lampiran 5. Data Nilai Implementasi

1. Game

NO	NAMA	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	WAKTU
1	Zidan	1	0	1	1	1	7,11
2	Algi	0	1	1	2	1	12,03
3	Rizhan	0	0	2	1	0	7,04
4	Alfiansyah	0	0	1	1	5	8,49
5	Azka	1	1	6	1	1	15,34
6	Azki	1	1	3	4	1	10,11
7	Arsya	1	2	0	1	5	10,59
8	Satria	0	0	1	1	0	4,32
9	Afrian	1	1	0	0	1	2,43
10	Angelo	0	1	2	1	1	17
11	Maira	0	0	2	4	3	10,45
12	Almeera	1	1	1	1	2	15,39
13	Aqila	0	1	1	1	3	14,19
14	Fildan	1	1	1	0	2	7,39
15	Aulia	0	1	1	0	0	11,39
16	Rayhan	1	1	1	2	1	11,51
17	Rangga	1	1	2	2	1	15,17
Total		9	13	26	23	28	179,95
Rata-Rata		0,5294117647	0,7647058824	1,529411765	1,352941176	1,647058824	10,58529412

2. Konvesional

NO	NAMA	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5	WAKTU
1	Zidan	0	0	1	1	1	6,07
2	Afrian	0	0	1	0	1	6,26
3	Rizhan	0	0	0	1	2	9,46
4	Satria	0	0	0	2	3	10,15
5	Aulia	0	0	0	1	2	11,22
6	Algi	0	1	2	3	1	11,48
7	Angelo	0	2	0	5	2	11,58
8	Maira	0	3	0	3	1	13,24
9	Fildan	0	5	0	5	2	15,3
10	Aqila	0	1	0	1	0	16,22
11	Rayhan	4	3	3	2	0	16,48
12	Azka	1	1	1	4	1	19,46
13	Almeera	3	0	2	1	0	23,27
14	Putra (Azki)	2	5	5	0	0	24,58
15	Rangga	3	5	1	5	3	25,08
16	Alfiansyah	5	5	1	0	1	29,19
17	Arsya	0	0	0	1	2	32,34
Total		36,52941176	57,76470588	70,52941176	82,35294118	79,64705882	281,38
Rata-Rata		1,058823529	1,604575163	1,959150327	2,287581699	2,212418301	16,55176471