L A M P I R A N

Lampiran 1 - Script Aplikasi

AINavMesh.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.AI;

public class AINavMesh :
MonoBehaviour

{

    GameObject destPos;

    NavMeshAgent agent;

    Rigidbody rigid;

    void Start()

    {

        rigid =
GetComponent<Rigidbody>();

        agent =
GetComponent<NavMeshAgent>();

        destPos =
GameObject.Find("RealDestPos");

    }

    void FixedUpdate()

    {

        agent.destination =
destPos.transform.position;

        FreezeRotation();

    }

    void FreezeRotation()

    {

        rigid.angularVelocity =
Vector3.zero;

    }

}
```

AudioManager.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System;



public class AudioManager :
MonoBehaviour

{

    public static AudioManager
Instance;

    public Sound[] musicSounds,
sfxSounds;

    public AudioSource musicSource,
sfxSource;

    private void Awake()

    {

        if (Instance == null)

        {

            Instance = this;


DontDestroyOnLoad(gameObject);

        }

        else

        {

            Destroy(gameObject);

        }

    }

    private void Start()

    {

        PlayMusic("Theme");

    }
```

```
    public void PlayMusic(string
name)

    {

        Sound s =
Array.Find(musicSounds, x => x.name
== name);

        if (s == null)

        {

            Debug.Log("Music not
found");

        }

        else

        {

            musicSource.clip =
s.clip;

            musicSource.Play();

        }

    }

    public void PlaySFX(string name)

    {

        Sound s =
Array.Find(sfxSounds, x => x.name ==
name);

        if (s == null)

        {

            Debug.Log("Sound not
found");

        }

        else

        {


sfxSource.PlayOneShot(s.clip);

        }

    }

    public void ToggleMusic()

    {

        musicSource.mute =
!musicSource.mute;

    }

    public void ToggleSFX()

    {

        sfxSource.mute =
!sfxSource.mute;

    }

    public void MusicVolume(float
volume)

    {

        musicSource.volume = volume;

    }

    public void SFXVolume(float
volume)

    {

        sfxSource.volume = volume;

    }

}
```

backpress.cs

```
using UnityEngine;

using System.Collections;

using UnityEngine.SceneManagement;

public class backpress :
MonoBehaviour {

    public string MainMenu =
"MainMenu";

        // Update is called once per
frame

        void Update()

        {

                if
(Input.GetKeyDown(KeyCode.Escape))

         {
```

```
SceneManager.LoadScene("MainMenu");
        }
       }
    public void MainMenuScene()
    {
SceneManager.LoadScene(MainMenu);
    }
    public void QuitGames()
    {
        Application.Quit();
    }
}
```

BGScroll.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class BGScroll : MonoBehaviour
{
    public float speed = 0.2f;
    Material mat;
    // Start is called before the
first frame update
    void Start()
    {
        MeshRenderer mr =
GetComponent<MeshRenderer>();
        mat = mr.material;
    }
    // Update is called once per
frame
    void Update()
    {
                // P =  P0 + vt
        mat.mainTextureOffset +=
Vector2.down * speed *
Time.deltaTime;
    }
}
```

BlinkText.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
public class BlinkText :
MonoBehaviour
{
    public GameObject readText;
    void Awake()
    {
        readText.SetActive(false);
        StartCoroutine(ShowReady());
    }
    IEnumerator ShowReady()
    {
        int count = 0;
        while (count < 100)
        {
            readText.SetActive(true);
            yield return new
WaitForSeconds(.5f);
```

```
readText.SetActive(false);

            yield return new
WaitForSeconds(.5f);

            count++;

        }

    }

    // Update is called once per
frame

}
```

BounceWall.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BounceWall :
MonoBehaviour

{

    [SerializeField] string
playerTag;

    [SerializeField] float
bounceForce;

    void Start()

    {

    }

    private void
OnCollisionEnter(Collision collision)

    {

        if (collision.transform.tag
== playerTag)

        {

            CollisionSound();

            Rigidbody otherRB =
collision.rigidbody;

            otherRB.velocity = new
Vector3(0, 0, 0);

otherRB.AddForce(Vector3.back *
bounceForce, ForceMode.Impulse);

        }

    }

    // Start is called before the
first frame update

    // Update is called once per
frame

    void Update()

    {

    }

    void CollisionSound()

    {

       AudioSource colsound=
GetComponent<AudioSource>();

       colsound.Play();

    }

}
```

BoxCollect.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using TMPro;

public class BoxCollect :
MonoBehaviour

{

    // Start is called before the
first frame update

    void Start()

    {
PlayerPrefs.SetInt("BoxCollect", 0);

    }
```

```
    // Update is called once per
frame

    void Update()

    {

        GetComponent<TMP_Text>().text
=
PlayerPrefs.GetInt("BoxCollect").ToSt
ring();

    }

}
```

CreateOpened.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;



public class CrateOpened :
MonoBehaviour

{

    public GameObject Treasure;

    public ParticleSystem Particle;

    public Collider CrateCol;

    public GameObject PanelKuis;

    public AudioSource BoxOpenSfx,
FireworkSfx;

    public int ScoreValue;



    void OnTriggerEnter(Collider
TriggerCol)

    {

        if (TriggerCol.gameObject.tag
== "Player")

        {

            BoxOpenSfx.Play();

StartCoroutine(TakeTreasure());

        }

    }

    IEnumerator TakeTreasure()

    {

        yield return new
WaitForSeconds(0.5f);

        Particle.Play();

        FireworkSfx.Play();

        int Collect =
PlayerPrefs.GetInt("BoxCollect") +
ScoreValue;


PlayerPrefs.SetInt("BoxCollect",
Collect);

        StartCoroutine(ShowKuis());

    }



    IEnumerator ShowKuis()

    {

        yield return new
WaitForSeconds(0.2f);

        PanelKuis.SetActive(true);


StartCoroutine(DestroyObject());

    }

    IEnumerator DestroyObject()

    {

        yield return new
WaitForSeconds(0.3f);

        Destroy(Treasure);

    }

}
```

IntroUI.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class IntroUI : MonoBehaviour

{

    public GameObject missionUI;

    public GameObject missionPos;

    Vector3 dir;

    public float speed = 3f;

    // Start is called before the
first frame update

    void Start()

    {

    }

    // Update is called once per
frame

    void Update()

    {

        dir =
missionPos.transform.position -
missionUI.transform.position;

        missionUI.transform.position
+= dir * speed * Time.deltaTime;

    }

}
```

LHS_Background.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class LHS_background :
MonoBehaviour

{

    public float delta = 0.02f;

    public float speed = 20f;

    Vector3 pos;

    // Start is called before the
first frame update

    void Start()

    {

        pos = transform.position;

    }

    // Update is called once per
frame

    void Update()

    {

        Vector3 v = pos;

        v.x += delta *
Mathf.Sin(Time.time * speed);

        transform.position = v;

    }

}
```

LHS_Bar.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;


public class LHS_Bar : MonoBehaviour

{

    //public GameObject player;

    //private GameObject m_goBar;
```

```
    public GameObject bar;

    // Start is called before the
first frame update

    void Start()

    {

        //m_goBar =
GameObject.Find("Canvas/Image");

        bar.SetActive(true);

        bar.transform.position =
Camera.main.WorldToScreenPoint(transf
orm.position + new Vector3(0, 3.35f,
0));

    }

    // Update is called once per
frame

    void FixedUpdate()

    {

        //bar.SetActive(true);

        //transform.position =
Camera.main.WorldToScreenPoint(player
.transform.position + Vector3.up *
2.8f);

        //bar.transform.position =
Camera.main.WorldToScreenPoint(transf
orm.position + new
Vector3(0,0.8f,0));

    }

}
```

LHS_bouncing.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;



public class LHS_bouncing :
MonoBehaviour

{

    public Vector3 MovePos;

    public float _speed = 4f;

    // Start is called before the
first frame update

    void Start()

    {

        MovePos = new Vector3(1f,
1f).normalized;

    }

    // Update is called once per
frame

    void Update()

    {

    }

    private void
OnCollisionEnter(Collision collision)

    {

        if
(collision.gameObject.tag.Equals("Wal
l"))

            transform.position +=
MovePos * _speed * Time.deltaTime;

    }

}
```

LHS_Camera.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;



public class LHS_Camera :
MonoBehaviour

{

    float x;

    float y;
```

```
    public float distance = 10;

    public GameObject player;


    // Start is called before the
first frame update

    void Start()

    {

    }

    // Update is called once per
frame

    void LateUpdate()

    {

        CameraRotate();

    }

    void CameraRotate()

    {

        //x += Input.GetAxis("Mouse
X");

        //y -= Input.GetAxis("Mouse
Y");

        //transform.rotation =
Quaternion.Euler(y, x, 0);

        //y = Mathf.Clamp(y, -10,
30);

        Vector3 reDistance = new
Vector3(0f, -1.8f, distance);

        transform.position =
player.transform.position -
transform.rotation * reDistance;

    }

}
```

LHS_CountdownController.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;


public class LHS_CountdownController
: MonoBehaviour

{

    public int countdownTime = 4;

    public Text countdownDisplay;

    public GameObject anim;


    public GameObject Num_A;   //1

    public GameObject Num_B;   //2

    public GameObject Num_C;   //3

    public GameObject Num_GO;

    //public GameObject Bar;


    Animator animator;


    public AudioSource mysfx;

    public AudioClip startsfx;

    public AudioClip gosfx;


    private void Awake()

    {


        animator =
anim.GetComponent<Animator>();

StartCoroutine(CountdownToStart());


        Num_A.SetActive(false); //1

        Num_B.SetActive(false); //2

        Num_C.SetActive(false); //3
```

```
        Num_GO.SetActive(false);

        Time.timeScale = 0;
    }

    IEnumerator CountdownToStart()
    {
        //Bar.SetActive(true);

        while (countdownTime > 0)
        {
            //Bar.SetActive(true);

            ChangeImage();

            countdownDisplay.text =
countdownTime.ToString();

            yield return new
WaitForSecondsRealtime(1f);

            countdownTime--;
        }

        countdownDisplay.text =
"GO!";
        //Num_GO.SetActive(false);

        Time.timeScale = 1;

        yield return new
WaitForSecondsRealtime(1f);

countdownDisplay.gameObject.SetActive
(false);
    }

    void ChangeImage()
    {
        int i = countdownTime;

        if (i == 4)
        {
            Num_C.SetActive(true);
            animator.SetBool("Num3",
true);

mysfx.PlayOneShot(startsfx);
        }
        if (i == 3)
        {
            //Num_C.SetActive(false);
            Num_B.SetActive(true);

//animator.SetBool("Num3", true);

mysfx.PlayOneShot(startsfx);
        }
        if (i == 2)
        {
            //Num_B.SetActive(false);
            Num_A.SetActive(true);

//animator.SetBool("Num3", true);

mysfx.PlayOneShot(startsfx);
        }
```

```
        if (i == 1)
        {
            //Num_A.SetActive(false);
            Num_GO.SetActive(true);
//animator.SetBool("Num3", true);
            mysfx.PlayOneShot(gosfx);
        }

    }
}
```

LHS_MainPlayer.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using
UnityStandardAssets.CrossPlatformInpu
t;

public class LHS_MainPlayer :
MonoBehaviour
{

    public float speed = 10;
    public float rotateSpeed = 5;
    public float jumpPower = 5;

    private Camera currentCamera;
    public bool UseCameraRotation =
true;

    public ParticleSystem dust;
    public GameObject bar;

    public string playerTag;
    public float bounceForce;
    public ParticleSystem bounce;


    public AudioSource mysfx;
    public AudioClip jumpfx;
    public AudioClip bouncefx;


    Animator anim;
    Rigidbody rigid;

    bool isJump;
    //bool isGround;
    bool jDown;

    bool isDie;

    float hAxis;
    float vAxis;

    Vector3 moveVec;

    // Start is called before the
first frame update
    void Awake()
    {
        rigid =
GetComponent<Rigidbody>();
```

```
        anim =
GetComponentInChildren<Animator>();


        bar.SetActive(true);

        bar.transform.position =
Camera.main.WorldToScreenPoint(transf
orm.position + new Vector3(0, 3.35f,
0));

    }


    private void Start()

    {

        currentCamera =
FindObjectOfType<Camera>();

    }


    private void FixedUpdate()

    {

        FreezeRotation();

        GetInput();

        Move();

        Turn();

        Jump();

        Die();

        Expression();

    }


    void FreezeRotation()

    {

        rigid.angularVelocity =
Vector3.zero;

    }


    void GetInput()

    {

        hAxis =
CrossPlatformInputManager.GetAxis("Ho
rizontal");

        vAxis =
CrossPlatformInputManager.GetAxis("Ve
rtical");

        jDown =
Input.GetButton("Jump");

    }


    void Move()

    {

        moveVec = new Vector3(hAxis,
0, vAxis).normalized;


        if (UseCameraRotation)

        {

            Quaternion v3Rotation =
Quaternion.Euler(0f,
currentCamera.transform.eulerAngles.y
, 0f);

            moveVec = v3Rotation *
moveVec;

        }


        transform.position += moveVec
* speed * Time.deltaTime;


        anim.SetBool("isMove",
moveVec != Vector3.zero);

    }


    void Turn()

    {

        if (hAxis == 0 && vAxis == 0)

            return;

        Quaternion newRotation =
Quaternion.LookRotation(moveVec);
```

```
        rigid.rotation =
Quaternion.Slerp(rigid.rotation,
newRotation, rotateSpeed *
Time.deltaTime);
    }

    void Jump()
    {
        if (jDown && !isJump)
        {
            rigid.AddForce(Vector3.up
* jumpPower, ForceMode.Impulse);
            isJump = true;

            //anim.SetBool("isJump",
true);
anim.SetTrigger("doJump");
mysfx.PlayOneShot(jumpfx);
            dust.Play();
        }
    }

    public void PlayerJump()
    {
        rigid.AddForce(Vector3.up *
jumpPower, ForceMode.Impulse);
        isJump = true;

        //anim.SetBool("isJump",
true);
        anim.SetTrigger("doJump");
        mysfx.PlayOneShot(jumpfx);
        dust.Play();
    }

    void Die()
    {
        if (isDie)
        {
            anim.SetTrigger("doDie");
            isDie = true;
        }
    }

    private void
OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag
== "Floor")
        {
            anim.SetBool("isJump",
false);
            isJump = false;
        }

        else if
(collision.gameObject.tag ==
"Platform")
        {
            anim.SetBool("isJump",
false);
            isJump = false;
        }

        else if
(collision.collider.tag == "Wall")
        {
```

```
            anim.SetTrigger("doDie");

            isDie = false;


            rigid.velocity = new
Vector3(0, 0, 0);


rigid.AddForce(Vector3.back *
bounceForce, ForceMode.Impulse);



mysfx.PlayOneShot(bouncefx);
            bounce.Play();


            bounce.transform.position
= transform.position;
        }
    }


    void Expression()
    {
        if
(Input.GetKeyDown(KeyCode.Alpha1))
        {

anim.SetTrigger("doDance01");
        }


        else if
(Input.GetKeyDown(KeyCode.Alpha2))
        {

anim.SetTrigger("doDance02");
        }


        else if
(Input.GetKeyDown(KeyCode.Alpha3))
        {

anim.SetTrigger("doVictory");
        }
    }


    public void Animation1()
    {
        anim.SetTrigger("doDance01");
    }


    public void Animation2()
    {
        anim.SetTrigger("doDance02");
    }


    public void Animation3()
    {
        anim.SetTrigger("doVictory");
    }
}
```

LHS_OnRotatePlatform.cs

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


public class LHS_OnRotatePlatform :
MonoBehaviour
{
    private void
OnCollisionEnter(Collision collision)
    {
```

```
        if
(collision.gameObject.CompareTag("Pla
tform"))

        {


            this.transform.parent =
collision.transform;



        }

    }

    private void
OnCollisionExit(Collision collision)

    {

        if
(collision.gameObject.CompareTag("Pla
tform"))

            this.transform.parent =
null;

    }

}
```

LHS_Respawn2.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;



public class LHS_Respawn2 :
MonoBehaviour

{

    [SerializeField] float
spawnValue;

    [SerializeField] GameObject
player;



    //[SerializeField] private
Transform player;

    [SerializeField] Transform
respawnPoint;

    Animator anim;



    private RaycastHit hit;

    private int layerMask;

    public float distance = 10;

    AudioSource resp;



    void Awake()

    {

        anim =
player.GetComponentInChildren<Animato
r>();

        layerMask = 1 << 7;

        resp =
GetComponent<AudioSource>();

    }



    private void FixedUpdate()

    {

        /*

        if
(player.transform.position.y < -
spawnValue)

        {

            DownPlayer();

        }

        */



        // RespawnTriggerDistance

        // DownPlayer

        // DownPlayer

        if
(Physics.Raycast(player.transform.pos
```

```
ition, -player.transform.up, out hit,
distance, layerMask))
        {
            //RespSound();
            DownPlayer();
            resp.Play();
        }

    }

    void DownPlayer()
    {
        anim.SetBool("isFalling",
true);
        //resp.Play();
    }

        private void
OnTriggerEnter(Collider other)
    {

if(other.CompareTag("Player"))
        {


            anim.SetBool("isFalling",
false);


            player.transform.position
= respawnPoint.transform.position;

//player.transform.GetChild(0).transf
orm.position = new Vector3(0, 0.09f,
0);

//Physics.SyncTransforms();
        }
    }
    /*public void RespSound()
    {
        AudioSource resp =
GetComponent<AudioSource>();
        resp.Play();
    }
    */
}
```

LoadingCount.cs

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


public class LoadingCount :
MonoBehaviour
{
    public int fullmember;
    public Text countText;
    public float increasingNum;
    public GameObject pressBtn;


    // Start is called before the
first frame update
    void Start()
    {
        //
pressText.SetActive(false);
    }
```

```
    void Update()

    {

        /*if(increasingNum <=
fullmember)

        {



            countText.text =
increasingNum + " / " + fullmember;

        }*/

        if (increasingNum <=
fullmember)

        {

            increasingNum +=
Time.deltaTime;

            countText.text =
Mathf.Round(increasingNum) + " / " +
fullmember;

            if (increasingNum >=
fullmember)

            {


pressBtn.SetActive(true);



StartCoroutine("ShowReady");

            }

        }

    }

    IEnumerator ShowReady()

    {

        yield return new
WaitForSeconds(11f);

        pressBtn.SetActive(true);

    }

}
```

PressAnyKey.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;



public class PressAnyKey :
MonoBehaviour

{

    public GameObject pressText;

    void Awake()

    {

        pressText.SetActive(false);

        if (pressText.activeSelf ==
true)

        {


StartCoroutine("ShowReady");

        }



    }



    IEnumerator ShowReady()

    {

        yield return new
WaitForSeconds(11f);

        pressText.SetActive(true);

    }

}
```

RotateCenter.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;



public class RotateCenter :
MonoBehaviour

{

    public float rotatingSpeed = 130;

    // Start is called before the
first frame update

    void Start()

    {

    }

    // Update is called once per
frame

    void Update()

    {


transform.Rotate(Vector3.forward,
rotatingSpeed * Time.deltaTime);



    }

}
```

RotPlatform.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;



public class RotPlatform_Down :
MonoBehaviour

{

    public float rotatingSpeed = 130;

    // Start is called before the
first frame update

    void Start()

    {

    }



    // Update is called once per
frame

    void Update()

    {


transform.Rotate(Vector3.down,
rotatingSpeed * Time.deltaTime);

    }

}
```

RotPlatform_Up.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;



public class RotPlatform_Up :
MonoBehaviour

{

    public float rotatingSpeed = 130;



    // Start is called before the
first frame update

    void Start()

    {



    }
```

```
    // Update is called once per
frame

    void Update()

    {

        transform.Rotate(Vector3.up,
rotatingSpeed * Time.deltaTime);

    }

}
```

SceneChange.cs

```
Using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;


public class SceneChange :
MonoBehaviour

{


    public void MainMenu()

    {


SceneManager.LoadScene("MainMenu");

    }


    public void InGameSceneChange()

    {


SceneManager.LoadScene("InGame");

    }


    public void EndingGame()

    {


SceneManager.LoadScene("Ending");

    }


    public void RestartGames()

    {


SceneManager.LoadScene(SceneManager.G
etActiveScene().name);

    }


    public void QuitGames()

    {

        Application.Quit();

    }

}
```

SceneChanngeLobbyToWaiting.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;


public class
SceneChanngeLobbytoWaiting :
MonoBehaviour

{

    public void SceneChange()

    {


SceneManager.LoadScene("WaitingUser")
;

    }
```

```
}
```

Score.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using TMPro;


public class Score : MonoBehaviour

{


    // Start is called before the
first frame update

    void Start()

    {

        PlayerPrefs.SetInt("Score",
0);

    }


    // Update is called once per
frame

    void Update()

    {

        GetComponent<TMP_Text>().text
=
PlayerPrefs.GetInt("Score").ToString(
);

    }

}
```

ShowHideInfo.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;


public class ShowHideInfo :
MonoBehaviour {


    public GameObject InfoText;

    private bool ShowInfo = false;


    public void showhideInfo()

    {

        if (!ShowInfo)

        {

            InfoText.SetActive(true);

            ShowInfo = true;

        }

        else

        {

InfoText.SetActive(false);

            ShowInfo = false;

        }

    }

}
```

Soal.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using DG.Tweening;


public class Soal : MonoBehaviour

{
```

```
    public Animator IconBenar,
IconSalah;

    public List<GameObject>
UIAnimation = new List<GameObject>();

    public GameObject PanelKuis;

    public float fadeTime = 1f;

    public int ScoreValue;

    public AudioSource BenarSfx,
SalahSfx, GameoverSfx;



    // Start is called before the
first frame update

    void Start()

    {


StartCoroutine(ListAnimation());

    }



    IEnumerator ListAnimation()

    {

        foreach (var item in
UIAnimation)

        {

            item.transform.localScale
= Vector3.zero;

        }

        foreach (var item in
UIAnimation)

        {


item.transform.DOScale(1f,
fadeTime).SetEase(Ease.OutBounce);

            yield return new
WaitForSeconds(0.25f);

        }

    }

    public void Jawaban(bool jawab)

    {

        if (jawab)

        {

            BenarSfx.Play();


IconBenar.Play("Jawaban");

            int Score =
PlayerPrefs.GetInt("Score") +
ScoreValue;


PlayerPrefs.SetInt("Score", Score);


StartCoroutine(HidePanel());



        }

        else

        {

            SalahSfx.Play();


IconSalah.Play("Jawaban");


StartCoroutine(HidePanel());

        }

    }



    IEnumerator HidePanel()

    {

        yield return new
WaitForSeconds(1.1f);

        PanelKuis.SetActive(false);

    }



    public void NextKuis()

    {
```

```
        gameObject.SetActive(false);

transform.parent.GetChild(gameObject.
transform.GetSiblingIndex() +
1).gameObject.SetActive(true);

    }

}
```

Sound.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;


[System.Serializable]

public class Sound

{

    public string name;

    public AudioClip clip;

}
```

TriggerFinish.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;


public class TriggerFinish :
MonoBehaviour

{

    public UIManager UiManager;


    void OnTriggerEnter(Collider
TriggerCol)

    {

        if (TriggerCol.gameObject.tag
== "Player")

        {

UiManager.PlayerSuccess();

        }

    }

}
```

TriggerSceneChange.cs

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;


public class TriggerSceneChange :
MonoBehaviour

{

    private void
OnCollisionEnter(Collision collision)

    {

        if(collision.gameObject.name
== "CM vcam8")

        {

SceneManager.LoadScene("InGame");

        }

    }

}
```

UIManager.cs

```
using System.Collections;

using System.Collections.Generic;
```

```
using UnityEngine;

using UnityEngine.UI;


public class UIManager :
MonoBehaviour

{

    public float limitTime;

    public Text textTimer;

    int min;

    float sec;

    public GameObject roundOver;

    public Animator success;

    public Animator failure;

    public GameObject player;

    public GameObject destPos;

    public GameObject
boxTriggerPoint;

    public GameObject Joystick;



    public static UIManager Instance;

    private void Awake()

    {

        Instance = this;

    }



    // Start is called before the
first frame update

    void Start()

    {

        GameObject player =
GameObject.Find("Player");

        roundOver.SetActive(false);

    }

    float waitTime = 2f;

    float curretTime = 0;

    // Update is called once per
frame

    void Update()

    {

        Timer();

    }



    void Timer()

    {

        limitTime -= Time.deltaTime;



        if (limitTime >= 60f)

        {

            min = (int)limitTime /
60;

            sec = limitTime % 60;

            textTimer.text = min + "
: " + (int)sec;

        }

        if (limitTime < 60f)

            textTimer.text =
"<color=white>" + (int)limitTime +
"</color>";

        if (limitTime < 30f)

            textTimer.text =
"<color=red>" + (int)limitTime +
"</color>";

        if (limitTime <= 0)

        {

            textTimer.text =
"<color=red>" + "Time Over" +
"</color>";
```

```
roundOver.SetActive(true);


            curretTime +=
Time.deltaTime;


            if (roundOver.activeSelf
== true)
            {
                if (curretTime >
waitTime)
                {
                    if (curretTime >
3f)
                    {

roundOver.SetActive(false);

failure.Play("RoundOver");

Joystick.SetActive(false);
                    }
                }
            }
        }
    }


    public void PlayerSuccess()
    {
        roundOver.SetActive(false);
        success.Play("RoundOver");
        Joystick.SetActive(false);
        enabled = false;

LHS_Particle.Instance.Success();

LHS_Particle.Instance.transform.posit
ion = player.transform.position + new
Vector3(0, 4f, 0);
    }
}
```

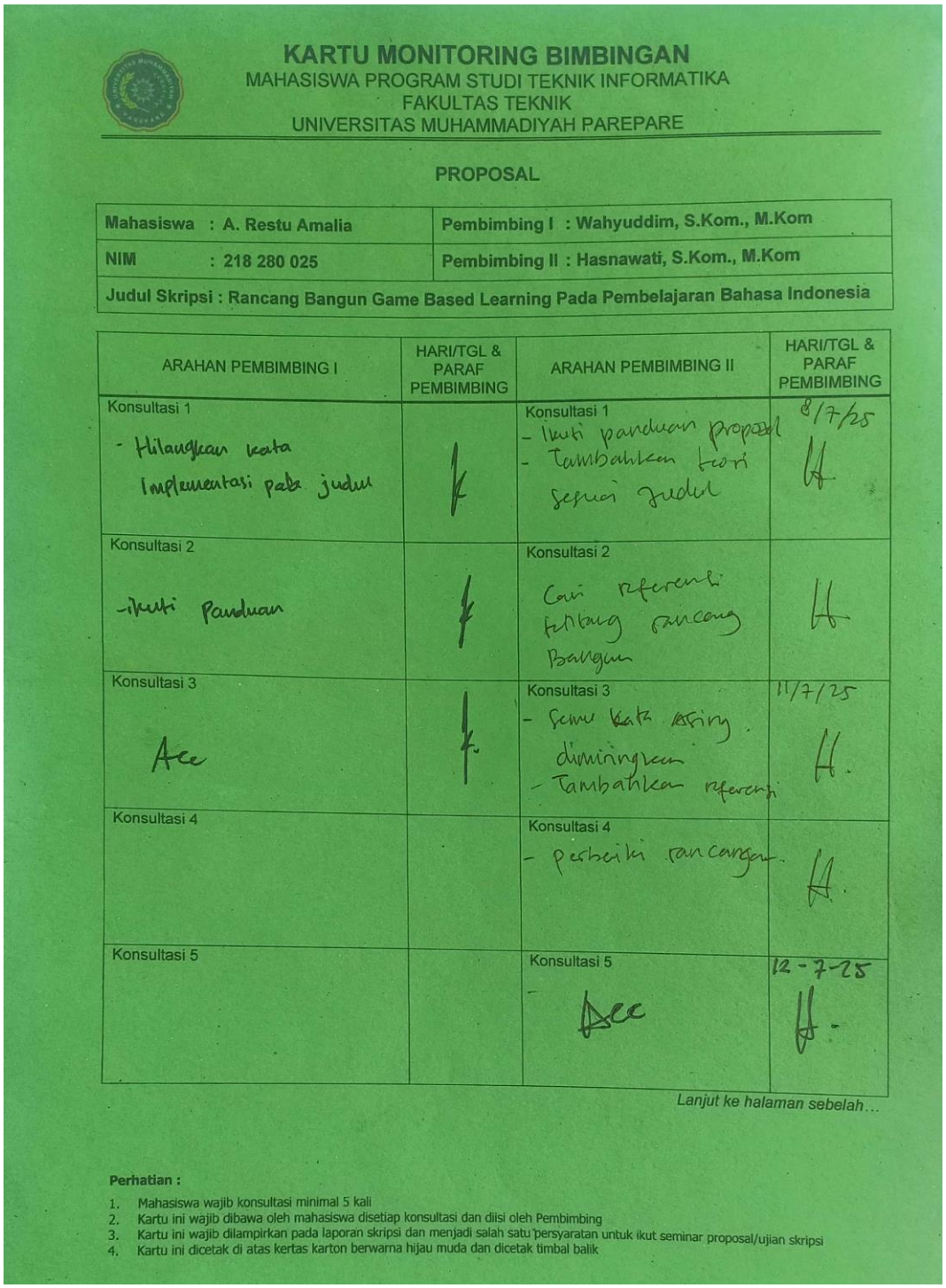
WaitingCharacterMove.cs

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


public class WaitingCharacterMove :
MonoBehaviour
{
    public GameObject destPos;
    float speed = 1.8f;
    Vector3 dir;


    // Start is called before the
first frame update
    void Start()
    {
    }
    // Update is called once per
frame
    void Update()
    {
        dir =
destPos.transform.position -
transform.position;
        transform.position += dir *
speed * Time.deltaTime;
    }
}
```

***Lampiran 2* – Kartu Monitoring Bimbingan Proposal**



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : A. Restu Amalia	**Pembimbing I : Wahyuddim, S.Kom., M.Kom**
NIM : 218 280 025	**Pembimbing II : Hasnawati, S.Kom., M.Kom**
Judul Skripsi : Rancang Bangun Game Based Learning Pada Pembelajaran Bahasa Indonesia	

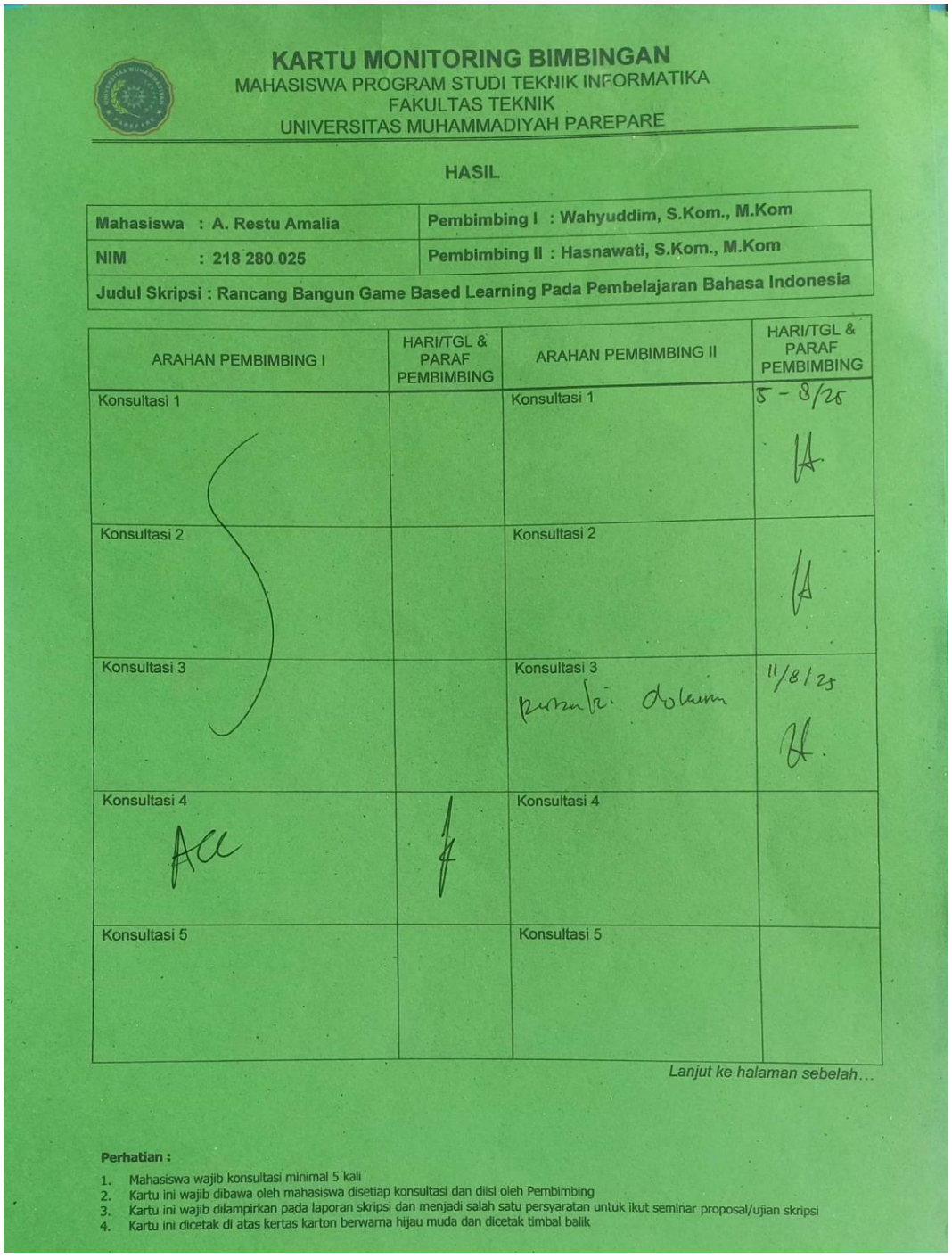
ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 - Hilangkan kata Implementasi pada judul		Konsultasi 1 - Ikuti panduan proposal - Tambahkan teori sesuai judul	8/7/25
Konsultasi 2 -ikuti Panduan		Konsultasi 2 Cari referensi tentang rancang Bangun	
Konsultasi 3 Acc		Konsultasi 3 - Semua kata asing diminingkan - Tambahkan referensi	11/7/25
Konsultasi 4		Konsultasi 4 - perbaiki rancangan	
Konsultasi 5		Konsultasi 5 Acc	12-7-25

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

Lampiran 3 – Kartu Monitoring Bimbingan Skripsi



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : A. Restu Amalia	**Pembimbing I : Wahyuddim, S.Kom., M.Kom**
NIM : 218 280 025	**Pembimbing II : Hasnawati, S.Kom., M.Kom**
Judul Skripsi : Rancang Bangun Game Based Learning Pada Pembelajaran Bahasa Indonesia	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1		Konsultasi 1	5 - 8/25
Konsultasi 2		Konsultasi 2	
Konsultasi 3		Konsultasi 3 perbaiki dokum	11/8/25
Konsultasi 4 ACC		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik