

LAMPIRAN



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Fikriadi sakrani
Nim : 1318078
Jurusan : Teknik Informatika S-1
Judul : Implementasi *Finite State Machine* Sebagai Kontrol Untuk *Non Player Karakter* Pada *Game Latri And The Last Tree*

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1) Pada

Hari : Senin
Tanggal : 20 Juli 2020
Nilai : 79

Panitia Ujian Skripsi
Dosen Pembimbing I

Suryo Adi Wibowo, S.T, M.T.
NIP .P.1031100438



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Fikriadi sakrani
Nim : 1318078
Jurusan : Teknik Informatika S-1
Judul : Implementasi *Finite State Machine* Sebagai Kontrol Untuk *Non Player Karakter* Pada Game *Lastri And The Last Tree*

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1) Pada

Hari : Senin
Tanggal : 20 Juli 2020
Nilai : 79

Panitia Ujian Skripsi
Dosen Pembimbing II

Febriana Santi Wahyuni, S.kom, M.Kom
NIP.P 1031000425



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang

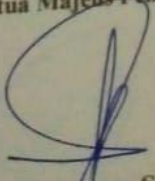
BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Fikriadi sakrani
Nim : 1318078
Jurusan : Teknik Informatika S-1
Judul : Implementasi *Finite State Machine* Sebagai Kontrol Untuk *Non Player Karakter* Pada *Game Latri And The Last Tree*

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1) Pada

Hari : Senin
Tanggal : 20 Juli 2020
Nilai : 79

Panitia Ujian Skripsi
Ketua Majelis Penguji


Survo Adi Wibowo, S.T.M.T.
NIP. P.1031100438



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata I Program Studi Teknik Informatika ,
maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Fikriadi Sakrani
NIM : 1318078
JURUSAN : Teknik Informatika S-1
JUDUL : IMPLEMENTASI FINITE STATE MACHINE SEBAGAI
KONTROL UNTUK NON PLAYER CHARACTER PADA
GAME LASTRI AND THE LAST TREE

No.	Penguji	Tanggal	Uraian	Paraf
1.	Penguji II	20 Juli 2020	<ol style="list-style-type: none">1. Membuat game intro lebih menarik. ✓2. Menuliskan misi dari tiap level. ✓3. Menjelaskan masing-masing character enemy. ✓4. Menjelaskan desain HP pada enemy dan bos pada tiap levelnya. ✓5. Memperbaiki penulisan abstrak dan latar belakang.6. Memperbaiki typo. ✓7. Memasukkan pengujian user pada laporan. ✓8. Memasukkan bukti valid pengujian user pada lampiran. ✓9. Demo program. ✓	

Dosen Penguji II

Nurlaily Vendyansyah, S.T., M.T
NIP.P 1031900557



PROGRAM STUDI TEKNIK INFORMATIKA S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
Jl. Raya Karanglo KM-02, Singosari, Kabupaten Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Fikriadi Sakrani
NIM : 1318078
JURUSAN : Teknik Informatika S-1
JUDUL : IMPLEMENTASI FINITE STATE MACHINE SEBAGAI KONTROL
UNTUK NON PLAYER CHARACTER PADA GAME LASTRI AND
THE LAST TREE

No.	Penguji	Tanggal	Uraian	Paraf
2.	Penguji II	20 Juli 2020	1. Menambahkan penjelasan AI game pada abstrak 2. Menambahkan animasi pada penjelasan game 3. Menambahkan sists pada bab 2	

Dosen Penguji II

Mira Orisa, ST, MT

NIP.P. 1031000435



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata I Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Fikriadi Sakrani
NIM : 1318078
JURUSAN : Teknik Informatika S-1
JUDUL : IMPLEMENTASI FINITE STATE MACHINE SEBAGAI KONTROL UNTUK NON PLAYER CHARACTER PADA GAME LASTRI AND THE LAST TREE

No.	Penguji	Tanggal	Uraian
1.	Penguji I	20 Juli 2020	<ol style="list-style-type: none">1. Membuat game intro lebih menarik.2. Menuliskan misi dari tiap level.3. Menjelaskan masing-masing character enemy.4. Menjelaskan desain HP pada enemy dan bos pada tiap levelnya.5. Memperbaiki penulisan abstrak dan latar belakang.6. Memperbaiki typo.7. Memasukkan pengujian user pada laporan.8. Memasukkan bukti valid pengujian user pada lampiran.9. Demo program.
2.	Penguji II	20 Juli 2020	<ol style="list-style-type: none">1. Menambahkan penjelasan AI game pada abstrak.2. Menambahkan animasi pada penjelasan game.3. Menambahkan sitasi pada bab 2.

Dosen Pembimbing I

Survo Adi Wibowo, S.T.M.T.
NIP. P.1031100438



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Fikriadi Sakrani
NIM : 1318078
JURUSAN : Teknik Informatika S-1
JUDUL : IMPLEMENTASI FINITE STATE MACHINE SEBAGAI KONTROL
UNTUK NON PLAYER CHARACTER PADA GAME LASTRI AND
THE LAST TREE

No.	Penguji	Tanggal	Uraian
1.	Penguji I	20 Juli 2020	<ol style="list-style-type: none">1. Membuat game intro lebih menarik.2. Menuliskan misi dari tiap level.3. Menjelaskan masing-masing character enemy.4. Menjelaskan desain HP pada enemy dan bos pada tiap levelnya.5. Memperbaiki penulisan abstrak dan latar belakang.6. Memperbaiki typo.7. Memasukan pengujian user pada laporan.8. Memasukkan bukti valid pengujian user pada lampiran.9. Demo program.
2.	Penguji II	20 Juli 2020	<ol style="list-style-type: none">1. Menambahkan penjelasan AI game pada abstrak.2. Menambahkan animasi pada penjelasan game.3. Menambahkan sists pada bab 2.

Dosen Pembimbing II

Febriana Santi Wahyuni, S.kom, M.Kom

NIP.P 1031000425



PT BSI (PERSERO) MALANG
BANK NAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UNIH DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus 1 : J. Bendungan Sgaru-garu No. 2 Telp. (0341) 531431 (Hunting), Fax. (0341) 532015 Malang 65145
Kampus 2 : J. Raya Kowonglo, Km 2 Telp. (0341) 417538 Fax. (0341) 417534 Malang

Malang, 24 April 2020

Nomor : ITN-04-128/III-T.INF/TA/2020
Lampiran : ---
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Suryo Adi Wibowo, ST, MT
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Fikriadi Sakrani
Nim : 1318078
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

20 Februari 2020 S/D 20 Agustus 2020

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
K a t u a,

Suryo Adi Wibowo, ST., MT.
NIP.P. 1051100438

Form S-4a



PT. BNI (PERSERO) MALANG
BANK MUDA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : J. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 552015 Malang 65145
Kampus II : J. Raya Kerejati, Km 2 Telp. (0341) 411838 Fax. (0341) 417034 Malang

Malang, 24 April 2020

Nomor : ITN-04-128/III-T.INF/TA/2020
Lampiran : ---
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Febriana Santi Wahyuni, S.Kom.M.Kom
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Fikriadi Sakrani
Nim : 1318078
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

20 Februari 2020 S/D 20 Agustus 2020

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
e t u a,



Suryo Adi Wilopo, ST., MT.
NIP.P. 1031100438



INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1

FORMULIR BIMBINGAN SKRIPSI

Nama : Fikriadi Sakrani
Nim : 13.18078
Masa Bimbingan : 20 Februari s/d 20 Agustus 2020
Judul Skripsi : Implementasi Finite State Machine Sebagai Kontrol Untuk Non Player Karakter Pada Game "Latri and the Last Tree"

No.	Tanggal	Uraian	Paraf Pembimbing
1.	28 April 2020	Desain karakter harus sesuai dengan kreator.	
2.	30-04-2020	sudah diselesaikan. Untuk seminar proggres gameplay minimal level 1	
3.	02-05-2020	laporan. Memasukkan karakter, background, dll kedalam	
4.	12 juni 2020	Menyelesaikan gameplay level 1 sampai 3	
5.	13 juni 2020	play. Menambahkan UI Menu Utama, About, dan how to	
6.	14 juni 2020	Laporan bab 1 s/d bab 5	
7.	15 Juni 2020	Jurnal untuk Seminar Hasil	
8.	6 juli 2020	Penambahan backsound	
9.	11 juli 2020	Penambahan bos pada tiap level	
10.	13 juli 2020	Revisi laporan	

Malang, 16 Juli 2020

Dosen Pembimbing

(Suryo Adi Wibowo, ST, MT)

NIP.P. 1031000438



INSTITUT TEKNOLOGI NASIONAL MALANG
Fakultas Teknologi Industri
Program Studi Teknik Informatika S1

FORMULIR BIMBINGAN SKRIPSI

Nama : Fikriadi Sakrani
Nim : 13.18078
Masa Bimbingan : 20 Februari s/d 20 Agustus 2020
Judul Skripsi : Implementasi Finite State Machine Sebagai Kontrol Untuk Non Player
Karakter Pada Game "Lastri and the Last Tree"

No.	Tanggal	Uraian	Paraf Pembimbing
1.	28 April 2020	Diskusi hasil pengerjaan program	<i>[Signature]</i>
2.	2 Mei 2020	Penulisan laporan bab 1 - 2	<i>[Signature]</i>
3.	6 Mei 2020	Diskusi hasil pengerjaan program	<i>[Signature]</i>
4.	15 Juni 2020	Makalah Seminar Hasil	<i>[Signature]</i>
5.	16 Juli 2020	Revisi Laporan	<i>[Signature]</i>
6.	17 Juli 2020	ACC Uji Kompre	<i>[Signature]</i>
7.			
8.			
9.			
10.			

Malang, 17 Juli 2020
Dosen Pembimbing

[Signature]

(Febriana Santi W, S.kom, M.Kom)
NIP.P. 1031000425

**IMPLEMENTASI FINITE STATE MACHINE SEBAGAI KONTROL
UNTUK NON PLAYER CHARACTER PADA GAME LASTRI AND
THE LAST TREE**

Pengujian User

Email Address	Apakah tampilan karakter pada game menarik?	Apakah game yang dimainkan menantang dan menarik?	Apakah controler pada game nyaman digunakan?	Apakah tombol controller telah sesuai dengan fungsinya?	Apakah pesan yang terkandung dalam game tersampaikan ?	Apakah AI pada musuh berjalan tanpa kendala?
dodiye.fahmye@gmail.com	Baik	cukup	Baik	Baik	cukup	Baik
Najad16081998@gmail.com	Baik	Baik	Baik	Baik	Baik	Baik
smakmun342@gmail.com	Baik	Baik	cukup	Baik	cukup	cukup
ck72knight@gmail.com	Baik	Baik	cukup	Baik	cukup	Baik
farisbinirhas1@gmail.com	cukup	Baik	Baik	Baik	Baik	Baik
muliambois@gmail.com	cukup	Baik	Baik	Baik	Baik	Baik
egaul4055@gmail.com	cukup	Baik	Kurang	cukup	Baik	Baik
luckykorainluckykorain@gmail.com	Baik	Baik	Baik	cukup	Baik	cukup
bird39777@gmail.com	Baik	Baik	Baik	Baik	Baik	Baik
maharaniriri12@gmail.com	Baik	Baik	Baik	Baik	Baik	Baik
jemmyfrans40@gmail.com	Baik	Baik	Baik	Baik	Baik	Baik
edwinmk666@gmail.com	Baik	cukup	cukup	Baik	cukup	Baik

Player script :

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

public delegate void DeadEventHandler();
public class Player : Character
{
    private static Player instance;

    public event DeadEventHandler Dead;

    public static Player Instance
    {
        get
        {
            if (instance == null)
            {
                instance =
                GameObject.FindObjectOfType<Player>();
            }
            return instance;
        }
    }
    [SerializeField]
    private Transform bulletPos;

    [SerializeField]
    private Transform[] groundPoints;

    [SerializeField]
    private float groundRadius;

    [SerializeField]
    private LayerMask whatIsground;

    [SerializeField]
    private bool airControl;

    [SerializeField]
    private float jumpForce;

    [SerializeField]
    private GameObject bulletPrefab;

    private bool immortal = false;

    private SpriteRenderer spriteRenderer;

    [SerializeField]
    private float immortalTime;

    public Rigidbody2D MyRigidbody { get; set; }

    public bool Slide { get; set; }
    public bool Jump { get; set; }
```

```

public bool OnGround { get; set; }

public override bool IsDead
{
    get
    {
        if (healthStat.CurrentValue <= 0)
        {
            OnDead();
        }

        return healthStat.CurrentValue <= 0;
    }
}

private Vector2 startPos;

// Use this for initialization
public override void Start ()
{
    base.Start();
    startPos = transform.position;
    spriteRenderer = GetComponent<SpriteRenderer>();
    MyRigidbody = GetComponent<Rigidbody2D> ();
}

void Update()
{
    if (!TakingDamage && !IsDead)
    {
        if (transform.position.y <= -14f)
        {
            Death();
            // MyRigidbody.velocity = Vector2.zero;
            // transform.position = startPos;
        }
        HandleInput ();
    }
}

// Update is called once per frame
void FixedUpdate ()
{
    if (!TakingDamage && !IsDead)
    {
        float horizontal = Input.GetAxis ("Horizontal");
        OnGround = IsGrounded();
        HandleMovement (horizontal);
        Flip (horizontal);
        HandleLayer();
    }
}

public void OnDead()
{

```

```

        if (Dead != null)
        {
            Dead();
        }
    }

    private void HandleMovement(float horizontal)
    {
        if (MyRigidbody.velocity.y < 0)
        {
            gameObject.layer = 10;
            MyAnimator.SetBool("land", true);
        }
        if (!Attack && !Slide && (OnGround || airControl))
        {
            MyRigidbody.velocity = new Vector2 (horizontal *
movementSpeed, MyRigidbody.velocity.y);
        }
        if (Jump && MyRigidbody.velocity.y == 0)
        {
            MyRigidbody.AddForce(new Vector2(0, jumpForce));
        }

        MyAnimator.SetFloat("speed", Mathf.Abs(horizontal));
    }

    private void HandleInput()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            MyAnimator.SetTrigger("jump");
        }
        if(Input.GetKeyDown(KeyCode.Q))
        {
            MyAnimator.SetTrigger("attack");
        }
        if (Input.GetKeyDown(KeyCode.S))
        {
            MyAnimator.SetTrigger("slide");
        }
        if (Input.GetKeyDown(KeyCode.R))
        {
            MyAnimator.SetTrigger("shot");
        }
    }

    private void Flip(float horizontal)
    {
        if (horizontal > 0 && !facingRight || horizontal < 0 &&
facingRight)
        {
            ChangeDirection();
        }
    }

    private bool IsGrounded()

```

```

    {
        if (MyRigidbody.velocity.y <= 0)
        {
            foreach (Transform point in groundPoints)
            {
                Collider2D[] colliders =
Physics2D.OverlapCircleAll(point.position,groundRadius, whatIsground);

                for (int i = 0; i < colliders.Length; i++)
                {
                    if (colliders[i].gameObject !=
gameObject)
                    {
                        return true;
                    }
                }
            }
        }
        return false;
    }

private void HandleLayer()
{
    if (!OnGround)
    {
        MyAnimator.SetLayerWeight(1,1);
    }
    else
    {
        MyAnimator.SetLayerWeight(1,0);
    }
}

public void Shoting(int value)
{
    if (OnGround && value == 0)
    {
        if (facingRight)
        {
            GameObject tmp = (GameObject)
Instantiate(bulletPrefab, bulletPos.position, Quaternion.identity);

            tmp.GetComponent<Bullet>().Initialize(Vector2.right);
        }
        else
        {
            GameObject tmp = (GameObject)
Instantiate(bulletPrefab, bulletPos.position, Quaternion.identity);
            tmp.GetComponent<Bullet>().Initialize(Vector2.left);
        }
    }
}

private IEnumerator IndicateImmortal()
{
    spriteRenderer.enabled = false;
}

```

```

        yield return new WaitForSeconds(.1f);

        spriteRenderer.enabled = true;

        yield return new WaitForSeconds(.1f);
    }

    public override IEnumerator TakeDamage()
    {
        if (!immortal)
        {
            healthStat.CurrentValue -= 10;
            if (!IsDead)
            {
                MyAnimator.SetTrigger("damage");
                immortal = true;

                StartCoroutine(IndicateImmortal());
                yield return new
WaitForSeconds(immortalTime);

                immortal = false;
            }
            else
            {
                MyAnimator.SetLayerWeight(1, 0);
                MyAnimator.SetTrigger("die");
            }
        }
    }

    public override void Death()
    {
        MyRigidbody.velocity = Vector2.zero;
        MyAnimator.SetTrigger("idle");
        healthStat.CurrentValue = healthStat.MaxValue;
        transform.position = startPos;
    }

    private void OnCollisionEnter2D(Collision2D other)
    {
        if (other.gameObject.tag == "Coin")
        {
            GameManager.Instance.Collecte++;
            Destroy(other.gameObject);
        }
    }
}

```

Enemy Script :

```

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

```

```

public class Enemy : Character
{
    private IEnemyState currentState;

    public GameObject Target { get; set; }

    [SerializeField]
    private float meleeRange;

    [SerializeField]
    private Transform leftEdge;

    [SerializeField]
    private Transform rightEdge;

    private Canvas heathCanvas;

    private bool dropItem = true;

    public bool InMeleeRange
    {
        get
        {
            if (Target != null)
            {
                return Vector2.Distance(transform.position,
Target.transform.position) <= meleeRange;
            }

            return false;
        }
    }

    public override bool IsDead
    {
        get
        {
            return healtStat.CurrentValue <= 0;
        }
    }

    // Use this for initialization
    public override void Start ()
    {
        base.Start();
        Player.Instance.Dead += new DeadEventHandler(RemoveTarget);
        ChangeState(new IdleState());

        heathCanvas = transform.GetComponentInChildren<Canvas>();
    }

    // Update is called once per frame
    void Update ()
    {
        if (!IsDead)
        {

```

```

        if (!TakingDamage)
        {
            currentState.Execute();
        }
        LookAtTarget();
    }
}

public void RemoveTarget()
{
    Target = null;
    ChangeState(new PatrolState());
}

private void LookAtTarget()
{
    if (Target != null)
    {
        float xDir = Target.transform.position.x -
transform.position.x;
        if (xDir < 0 && facingRight || xDir > 0 &&
!facingRight)
        {
            ChangeDirection();
        }
    }
}

public void ChangeState(IEnemyState newState)
{
    if (currentState != null)
    {
        currentState.Exit();
    }

    currentState = newState;
    currentState.Enter(this);
}

public void Move()
{
    if (!Attack)
    {
        if (GetDirection().x > 0 && transform.position.x <
rightEdge.position.x || GetDirection().x < 0 && transform.position.x >
leftEdge.position.x)
        {
            MyAnimator.SetFloat("speed", 1);

            transform.Translate(GetDirection() *
(movementSpeed * Time.deltaTime));
        }
        else if (currentState is PatrolState)
        {
            ChangeDirection();
        }
    }
}

```

```

        }
        else if (currentState is RangedState)
        {
            Target = null;
            ChangeState(new IdleState());
        }
    }

    public Vector2 GetDirection()
    {
        return facingRight ? Vector2.right : Vector2.left;
    }

    public override void OnTriggerEnter2D(Collider2D other)
    {
        base.OnTriggerEnter2D(other);
        currentState.OnTriggerEnter(other);
    }

    public override IEnumerator TakeDamage()
    {
        if (!heathCanvas.isActiveAndEnabled)
        {
            heathCanvas.enabled = true;
        }
        healtStat.CurrentValue -= 10;

        if (!IsDead)
        {
            MyAnimator.SetTrigger("damage");
        }
        else
        {
            if (dropItem)
            {
                GameObject coin =
                (GameObject)Instantiate(GameManager.Instance.CollectionPrefab, new
                Vector3(transform.position.x, transform.position.y,
                +2),Quaternion.identity);

                Physics2D.IgnoreCollision(coin.GetComponent<Collider2D>(),
                GetComponent<Collider2D>());
                dropItem = false;
            }

            MyAnimator.SetTrigger("die");
            yield return null;
        }
    }

    public override void Death()
    {
        dropItem = true;
        Destroy(gameObject);
        heathCanvas.enabled = false;
    }
}

```


}