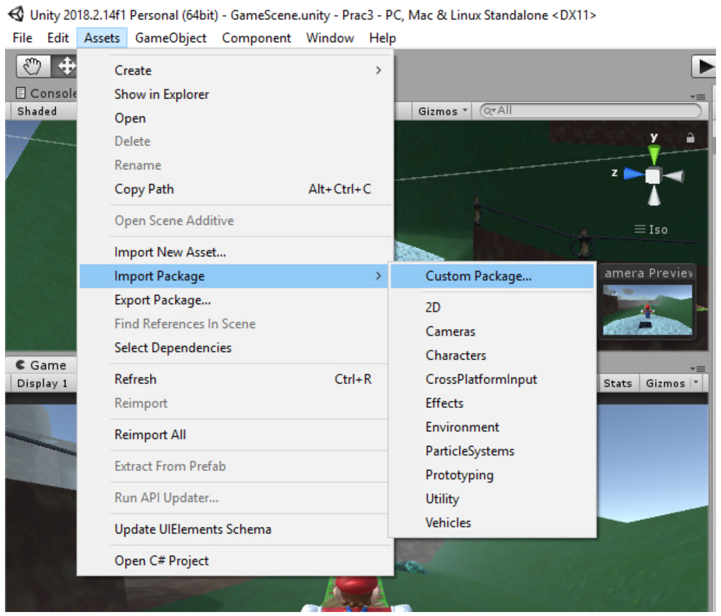
TecnoCampus Mataró-Maresme

DESENVOLUPAMENT DE JOCS 3D

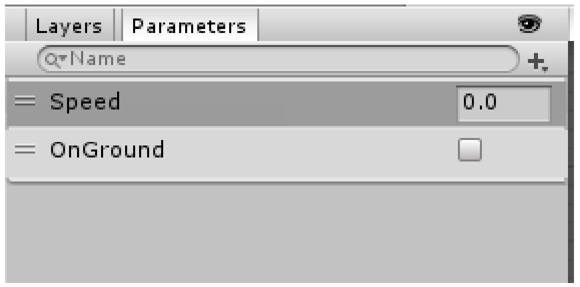
Práctica

Práctica 3 - https://youtu.be/D0PSJy2thnM

Importando assets



Player controller



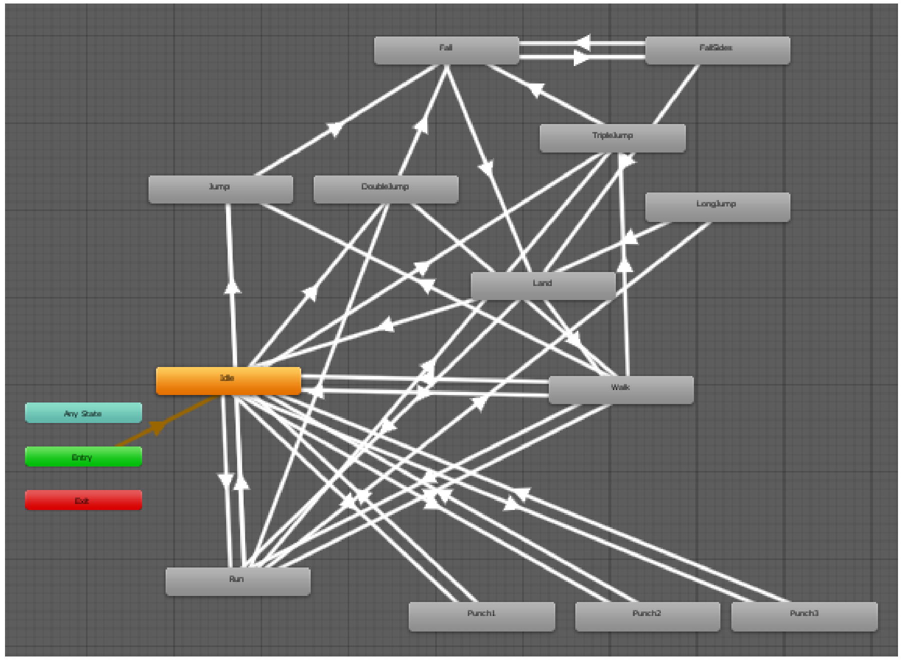
Layers | Parameters

Name

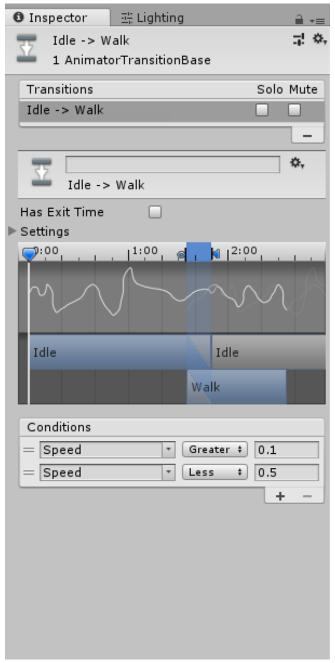
Speed 0.0

OnGround ☐

Player controller



Player controller


Inspector | Lighting

Idle -> Walk
1 AnimatorTransitionBase

Transitions	Solo	Mute
Idle -> Walk	☐	☐

Idle -> Walk

Has Exit Time ☐

Settings

Conditions

Speed	Greater	0.1
Speed	Less	0.5

Player Controller – Implementación

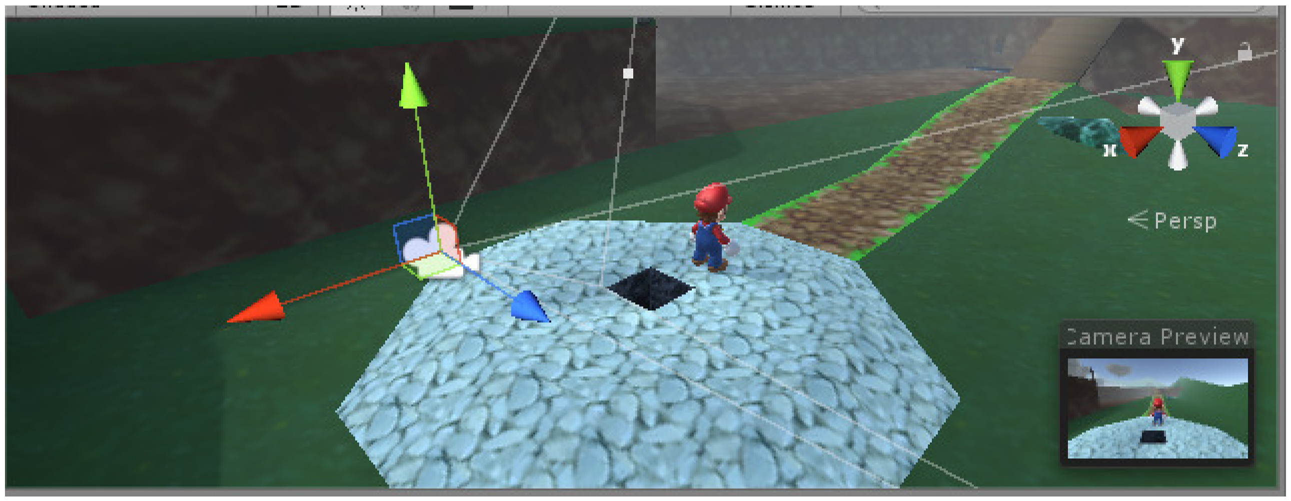
```
void Update()
{
	//..
	Vector3 l_Movement=Vector3.zero;
	Vector3 l_Forward=m_CameraController.transform.forward;
	Vector3 l_Right=m_CameraController.transform.right;
	l_Forward.y=0.0f;
	l_Forward.Normalize();
	l_Right.y=0.0f;
	l_Right.Normalize();
	if(Input.GetKey(m_UpKeyCode))
		l_Movement=l_Forward;
	else if(Input.GetKey(m_DownKeyCode))
		l_Movement=-l_Forward;
	//..
	m_Animator.SetFloat("Speed", l_HasMovement ? (l_Speed==m_RunSpeed ? 1.0f : 0.2f) : 0.0f);
}
```

Camera controller

50 common game camera mistakes -- and how to fix them

https://www.youtube.com/watch?v=C7307qRmlMI

Camera controller

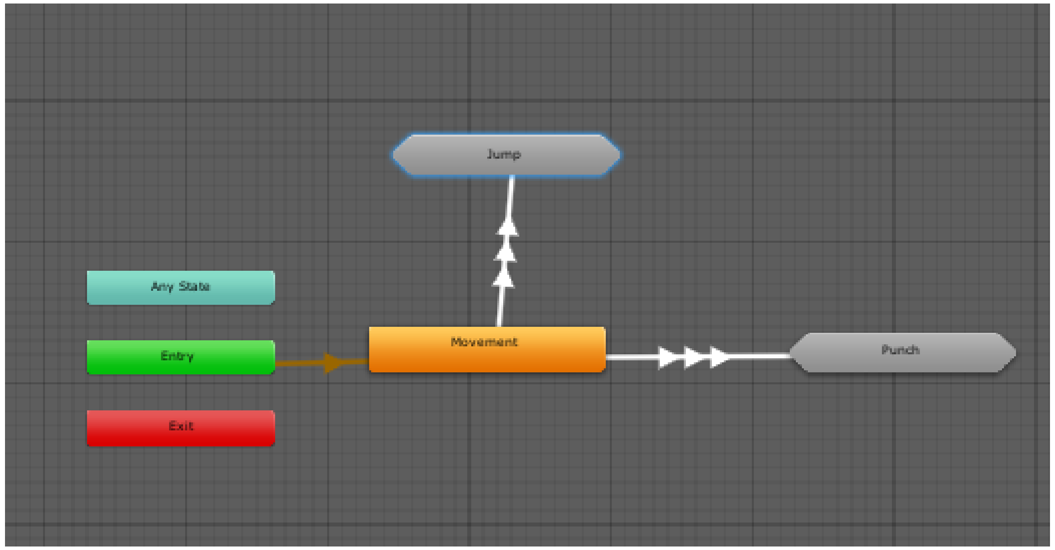


Camera controller

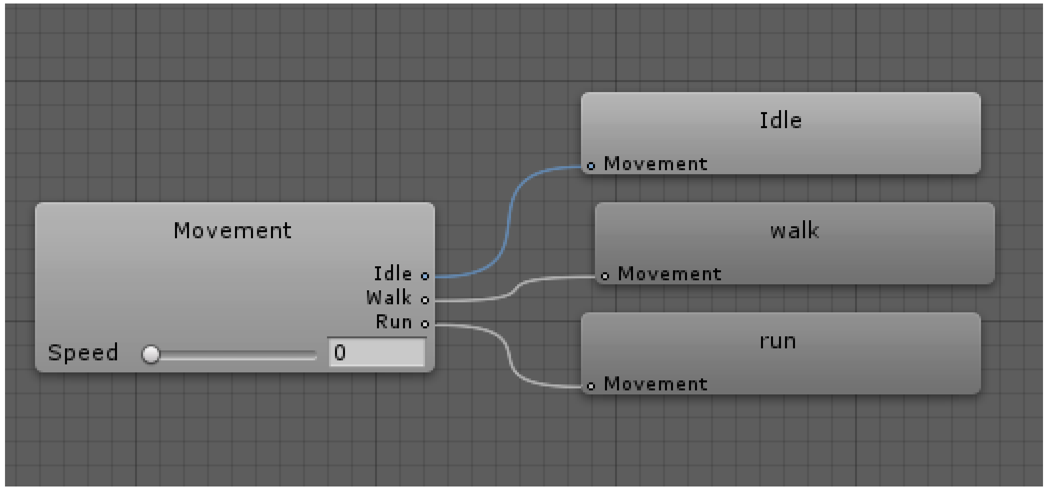
Camera Controller – Implementación

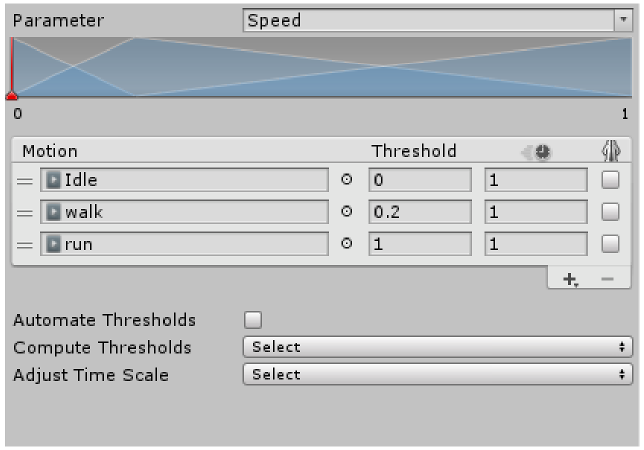
```
void LateUpdate()
{
        //..
        float l_MouseAxisX=Input.GetAxis("Mouse X");
        float l_MouseAxisY=Input.GetAxis("Mouse Y");
        //..
        Vector3 l_DesiredPosition=transform.position;

        if(!m_AngleLocked && (l_MouseAxisX>0.01f || l_MouseAxisX<-0.01f || l_MouseAxisY>0.01f || l_MouseAxisY<-0.01f))
        {
                Vector3 l_EulerAngles=transform.eulerAngles;
                float l_Yaw=(l_EulerAngles.y+180.0f);
                float l_Pitch=l_EulerAngles.x;

                l_Yaw+=m_YawRotationalSpeed*l_MouseAxisX*Time.deltaTime;
                l_Yaw*=Mathf.Deg2Rad;
                if(l_Pitch>180.0f)
                        l_Pitch-=360.0f;
                l_Pitch+=m_PitchRotationalSpeed*(-l_MouseAxisY)*Time.deltaTime;
                l_Pitch=Mathf.Clamp(l_Pitch, m_MinPitch, m_MaxPitch);
                l_Pitch*=Mathf.Deg2Rad;
                l_DesiredPosition=m_LookAt.position+new Vector3(Mathf.Sin(l_Yaw)*Mathf.Cos(l_Pitch)*l_Distance, Mathf.Sin(l_Pitch)*l_Distance, Mathf.Cos(l_Yaw)*Mathf.Cos(l_Pitch)*l_Distance);
                l_Direction=m_LookAt.position-l_DesiredPosition;
        }
        l_Direction/=l_Distance;

        if(l_Distance>m_DistanceToLookAt)
        {
                l_DesiredPosition=m_LookAt.position-l_Direction*m_DistanceToLookAt;
                l_Distance=m_DistanceToLookAt;
        }

        RaycastHit l_RaycastHit;
        Ray l_Ray=new Ray(m_LookAt.position, -l_Direction);
        if(Physics.Raycast(l_Ray, out l_RaycastHit, l_Distance, m_RaycastLayerMask.value))
                l_DesiredPosition=l_RaycastHit.point+l_Direction*m_OffsetOnCollision;

        transform.forward=l_Direction;
        transform.position=l_DesiredPosition;
}
```

Simplificando Player controller



BlendTree Player controller



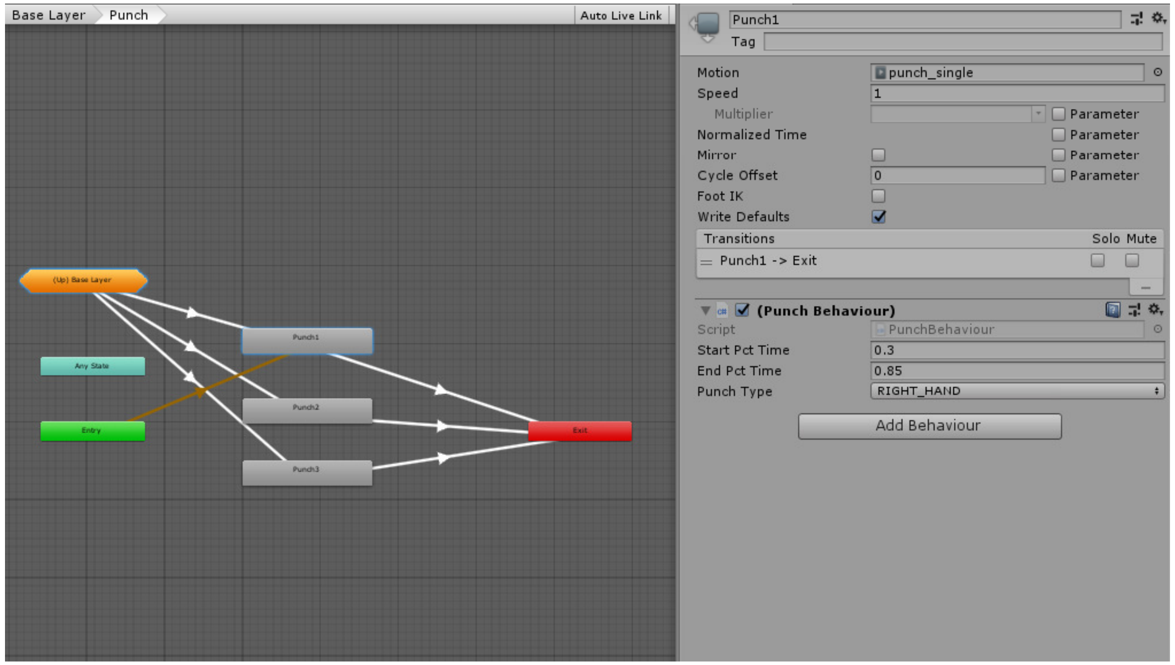


Parameter: Speed

Motion		Threshold		
Idle	⊙	0	1	☐
walk	⊙	0.2	1	☐
run	⊙	1	1	☐

Automate Thresholds ☐
Compute Thresholds: Select
Adjust Time Scale: Select

Punch Player controller



Punch1
Tag

Motion	punch_single	
Speed	1	
Multiplier		☐ Parameter
Normalized Time		☐ Parameter
Mirror	☐	☐ Parameter
Cycle Offset	0	☐ Parameter
Foot IK	☐	
Write Defaults	✓	

Transitions	Solo	Mute
Punch1 -> Exit	☐	☐

✓ **(Punch Behaviour)**

Script	PunchBehaviour
Start Pct Time	0.3
End Pct Time	0.85
Punch Type	RIGHT_HAND

PunchBehaviour – Implementación

```
public class PunchBehaviour : StateMachineBehaviour
{
	PlayerController m_PlayerController;
	public float m_StartPctTime;
	public float m_EndPctTime;
	public enum TPunchType
	{
		LEFT_HAND=0,
		RIGHT_HAND,
		FOOT
	}
	public TPunchType m_PunchType;

	override public void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
	{
		//...
	}
	override public void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
	{
		bool l_EnableHandPunch=stateInfo.normalizedTime>m_StartPctTime && stateInfo.normalizedTime<m_EndPctTime;
		if(m_PunchType==TPunchType.LEFT_HAND)
			m_PlayerController.EnableLeftHandPunch(l_EnableHandPunch);
		//...
	}
}
```

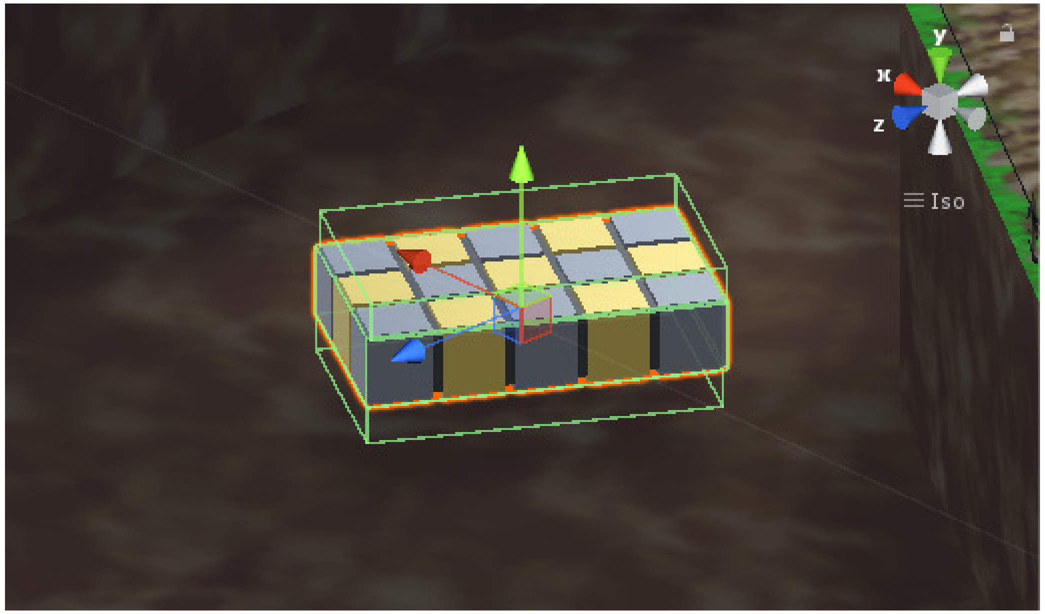
RestartGame – Implementación

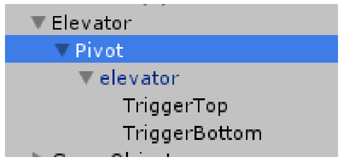
```
public interface IRestartGameElement
{
    void RestartGame();
}

public class GameController : MonoBehaviour
{
    List<IRestartGameElement> m_RestartGameElements=new List<IRestartGameElement>();

    void RestartGame()
    {
        foreach(IRestartGameElement l_RestartGameElement in m_RestartGameElements)
            l_RestartGameElement.RestartGame();
    }
    public void AddRestartGameElement(IRestartGameElement RestartGameElement)
    {
        m_RestartGameElements.Add(RestartGameElement);
    }
}
```

RestartGame PlayerController – Implementación

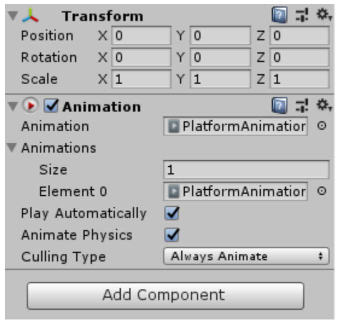
```
public class PlayerController : MonoBehaviour, IRestartGameElement
{
    //...
    public void RestartGame()
    {
        transform.position=m_RestartPosition;
        transform.rotation=m_RestartRotation;
    }
}
```

Plataforma ascensor





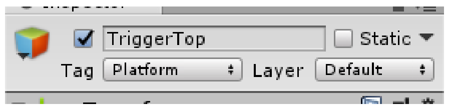
```
▼ Elevator
  ▼ Pivot
    ▼ elevator
        TriggerTop
        TriggerBottom
```



Transform			
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

✓ Animation	
Animation	PlatformAnimation
Animations	
Size	1
Element 0	PlatformAnimation
Play Automatically	✓
Animate Physics	✓
Culling Type	Always Animate

Add Component

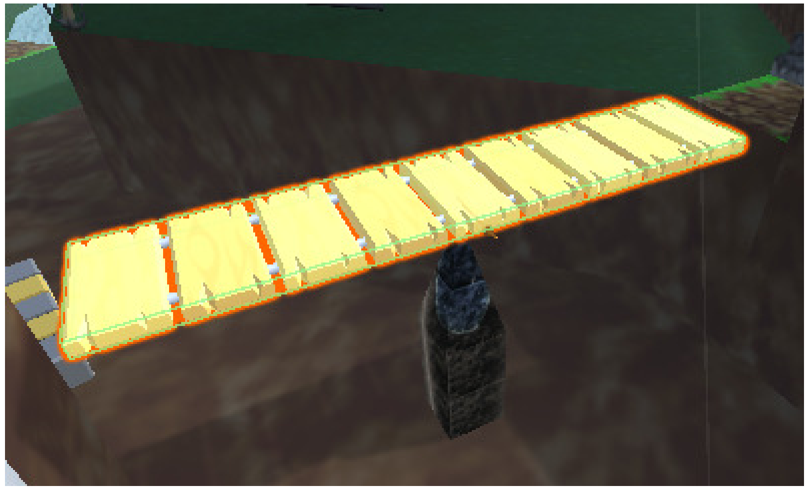


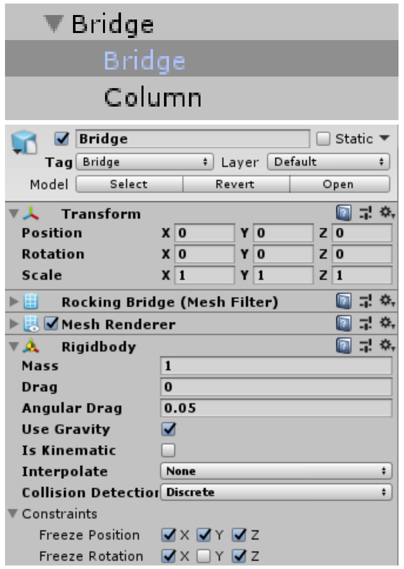
✓ TriggerTop		☐ Static	
Tag	Platform	Layer	Default

Plataforma ascensor – Implementación

```
void UpdatePlatform()
{
    if(m_CurrentPlatform!=null)
    {
        //Check with dot if current platform is looking up, if not detach platform
    }
}
public void OnTriggerEnter(Collider other)
{
    if(other.tag=="Platform" && m_CurrentPlatform==null)
        AttachPlatform(other.transform); //Set CurrentPlatform and set parent mario to platform
}
public void OnTriggerExit(Collider other)
{
    if(other.tag=="Platform" && m_CurrentPlatform!=null)
        DetachPlatform();
}
```

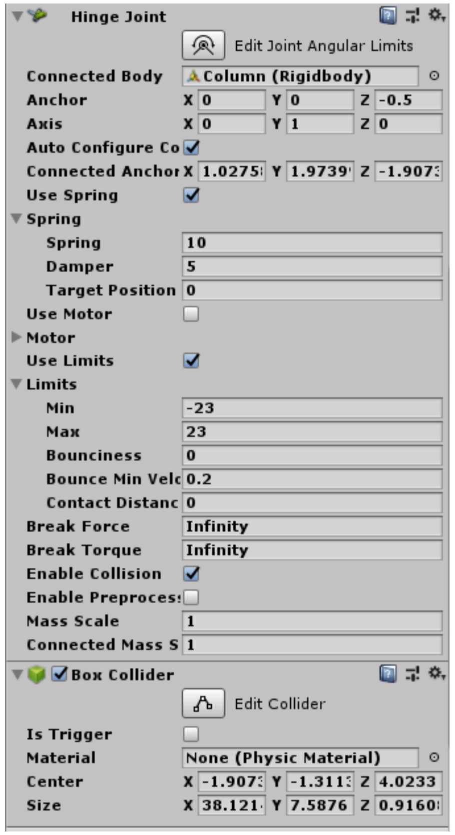
Plataforma puente





```
▼ Bridge
    Bridge
    Column
```

```
☑ Bridge                               ☐ Static ▼
Tag  Bridge            Layer  Default
Model   [Select]   [Revert]   [Open]

▼ Transform
Position        X 0      Y 0      Z 0
Rotation        X 0      Y 0      Z 0
Scale           X 1      Y 1      Z 1
▶ Rocking Bridge (Mesh Filter)
▶ ☑ Mesh Renderer
▼ Rigidbody
Mass                  1
Drag                  0
Angular Drag          0.05
Use Gravity           ☑
Is Kinematic          ☐
Interpolate           None
Collision Detectior   Discrete
▼ Constraints
   Freeze Position    ☑ X  ☑ Y  ☑ Z
   Freeze Rotation    ☑ X  ☐ Y  ☑ Z
```

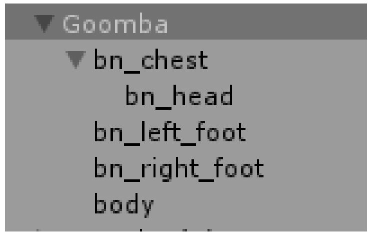


```
▼ Hinge Joint
                      Edit Joint Angular Limits
Connected Body        Column (Rigidbody)
Anchor                X 0        Y 0        Z -0.5
Axis                  X 0        Y 1        Z 0
Auto Configure Co     ☑
Connected Anchor      X 1.0275   Y 1.9739   Z -1.9073
Use Spring            ☑
▼ Spring
   Spring             10
   Damper             5
   Target Position    0
Use Motor             ☐
▶ Motor
Use Limits            ☑
▼ Limits
   Min                -23
   Max                23
   Bounciness         0
   Bounce Min Velc    0.2
   Contact Distanc    0
Break Force           Infinity
Break Torque          Infinity
Enable Collision      ☑
Enable Preproces      ☐
Mass Scale            1
Connected Mass S      1
▼ ☑ Box Collider
                      Edit Collider
Is Trigger            ☐
Material              None (Physic Material)
Center                X -1.9073  Y -1.3111  Z 4.0233
Size                  X 38.121   Y 7.5876   Z 0.9160
```

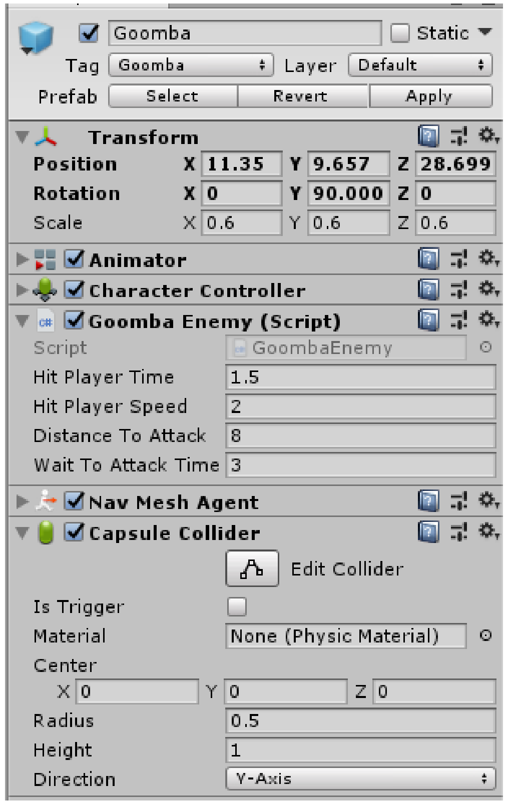
Plataforma puente – Implementación

```
public void OnControllerColliderHit(ControllerColliderHit hit)
{
    if(hit.collider.tag=="Bridge")
    {
        Rigidbody l_Bridge=hit.collider.attachedRigidbody;
        l_Bridge.AddForceAtPosition(-hit.normal*m_BridgeForce, hit.point);
    }
}
```

Kill Goomba



```
▼ Goomba
  ▼ bn_chest
      bn_head
    bn_left_foot
    bn_right_foot
    body
```



```
☑ Goomba                         ☐ Static
Tag  Goomba          Layer  Default
Prefab  Select  Revert  Apply
▼ Transform
Position   X 11.35   Y 9.657    Z 28.699
Rotation   X 0       Y 90.000   Z 0
Scale      X 0.6     Y 0.6      Z 0.6
▶ ☑ Animator
▶ ☑ Character Controller
▼ ☑ Goomba Enemy (Script)
Script                GoombaEnemy
Hit Player Time       1.5
Hit Player Speed      2
Distance To Attack    8
Wait To Attack Time   3
▶ ☑ Nav Mesh Agent
▼ ☑ Capsule Collider
                      Edit Collider
Is Trigger            ☐
Material              None (Physic Material)
Center
  X 0          Y 0          Z 0
Radius                0.5
Height                1
Direction             Y-Axis
```

Kill Goomba – Implementación Opción A (en player)

```
public void OnControllerColliderHit(ControllerColliderHit hit)
{
    //...
    if(hit.collider.tag=="Goomba")
    {
        if(CanKillWithFeet())
        {
            hit.collider.GetComponent<GoombaEnemy>().Kill();
            JumpOverEnemy();
        }
    }
}
public void JumpOverEnemy()
{
    m_VerticalSpeed=m_JumpOverEnemySpeed;
}
```

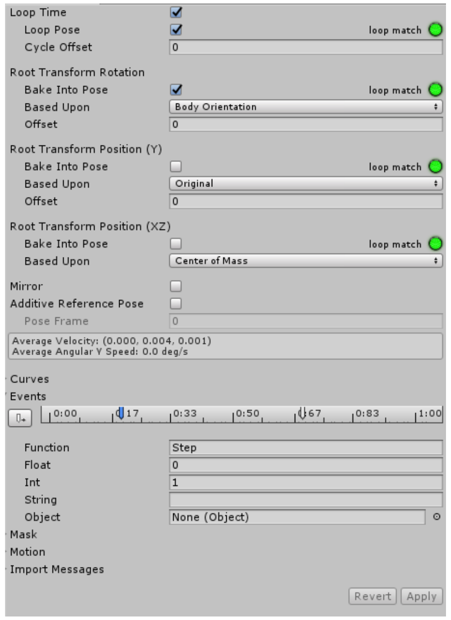
Kill Goomba – Implementación Opción B (en player)

```
public void OnTriggerEnter(Collider other)
{
    if(other.tag=="Goomba")
    {
        if(CanKillWithFeet())
        {
            other.GetComponent<GoombaEnemy>().Kill();
            JumpOverEnemy();
        }
    }
}
public void JumpOverEnemy()
{
    m_VerticalSpeed=m_JumpOverEnemySpeed;
}
```

Goomba (Implementación)

```
public class GoombaEnemy : MonoBehaviour, IRestartGameElement
{
     //...
     enum TState
     {
          IDLE,
          ATTACK,
          WAIT_TO_ATTACK,
          HIT_PLAYER
     }
     TState m_State=TState.IDLE;

     public void SetHitPlayer(Vector3 Direction)
     {
          m_State=TState.HIT_PLAYER;
          m_CurrentTime=0.0f;
          m_NavMeshAgent.SetDestination(transform.position);
          m_NavMeshAgent.ResetPath();
          m_NavMeshAgent.isStopped=true;
          m_NavMeshAgent.velocity=Vector3.zero;
          m_HitPlayerDirection=Direction;
     }
}
```

Eventos animation



```
Loop Time                      ✓
  Loop Pose                    ✓                 loop match 🟢
  Cycle Offset                 0
Root Transform Rotation
  Bake Into Pose               ✓                 loop match 🟢
  Based Upon                   Body Orientation
  Offset                       0
Root Transform Position (Y)
  Bake Into Pose               ☐                 loop match 🟢
  Based Upon                   Original
  Offset                       0
Root Transform Position (XZ)
  Bake Into Pose               ☐                 loop match 🟢
  Based Upon                   Center of Mass
Mirror                         ☐
Additive Reference Pose        ☐
  Pose Frame                   0
Average Velocity: (0.000, 0.004, 0.001)
Average Angular Y Speed: 0.0 deg/s
Curves
Events
  0:00   0:17   0:33   0:50   0:67   0:83   1:00
  Function                     Step
  Float                        0
  Int                          1
  String
  Object                       None (Object)
Mask
Motion
Import Messages
                                          Revert  Apply
```

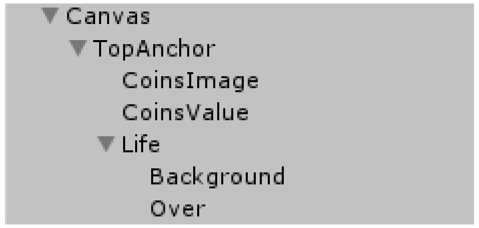
Eventos – Implementación Opción A

```
public void EventFunction(string stringParameter)
{
    //Code
}
```

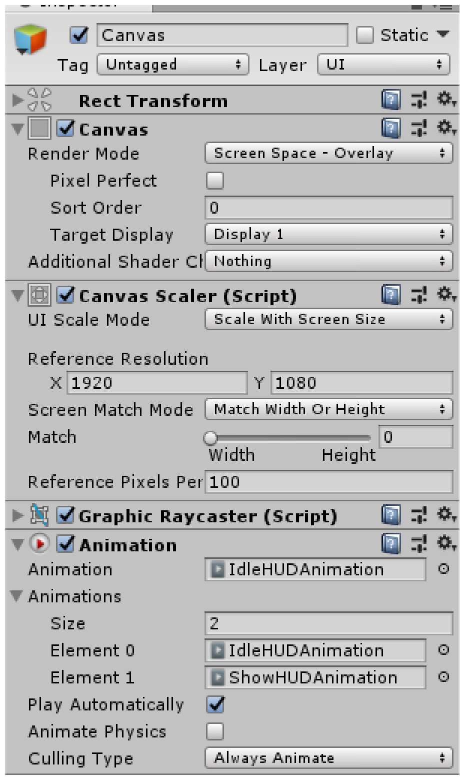
Eventos – Implementación Opción B

```
public void EventFunction(AnimationEvent animationEvent)
{
    string l_StringParmeter=animationEvent.stringParameter;
    float l_FloatParameter=animationEvent.floatParameter;
    int l_IntParameter=animationEvent.intParameter;
    //Code
}
```

Canvas



```
▼ Canvas
  ▼ TopAnchor
      CoinsImage
      CoinsValue
    ▼ Life
        Background
        Over
```



Canvas — Static
Tag: Untagged — Layer: UI

Rect Transform

Canvas

Render Mode	Screen Space - Overlay
Pixel Perfect	☐
Sort Order	0
Target Display	Display 1
Additional Shader Cl	Nothing

Canvas Scaler (Script)

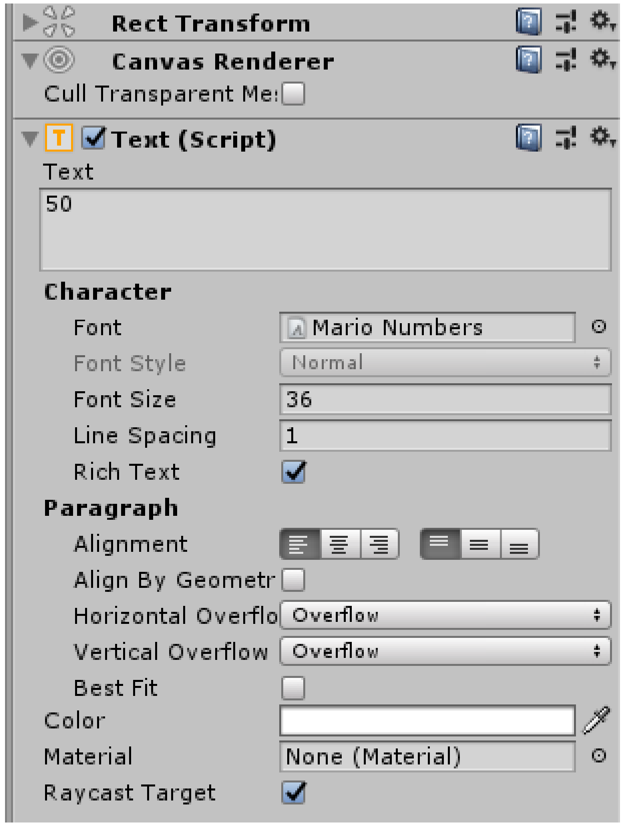
UI Scale Mode	Scale With Screen Size
Reference Resolution	X 1920 Y 1080
Screen Match Mode	Match Width Or Height
Match	0 (Width — Height)
Reference Pixels Per	100

Graphic Raycaster (Script)

Animation

Animation	IdleHUDAnimation
Animations	
Size	2
Element 0	IdleHUDAnimation
Element 1	ShowHUDAnimation
Play Automatically	✓
Animate Physics	☐
Culling Type	Always Animate

Canvas



Rect Transform

Canvas Renderer

Cull Transparent Me: ☐

☑ Text (Script)

Text

50

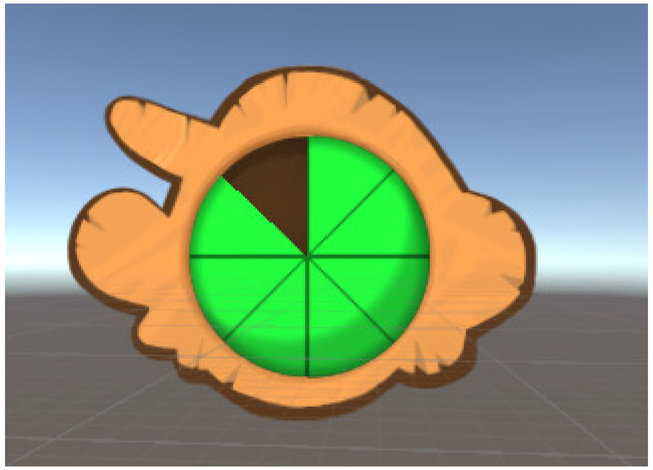
Character

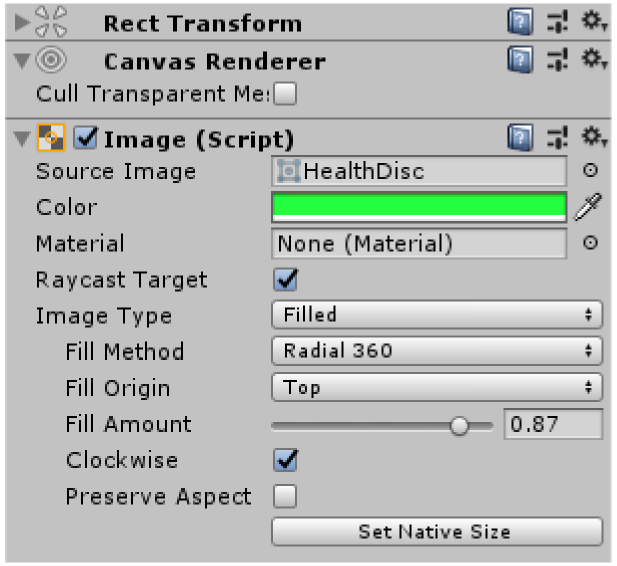
Font	Mario Numbers
Font Style	Normal
Font Size	36
Line Spacing	1
Rich Text	☑

Paragraph

Alignment	
Align By Geometr	☐
Horizontal Overflo	Overflow
Vertical Overflow	Overflow
Best Fit	☐
Color	
Material	None (Material)
Raycast Target	☑

Canvas





Rect Transform

Canvas Renderer

Cull Transparent Me: ☐

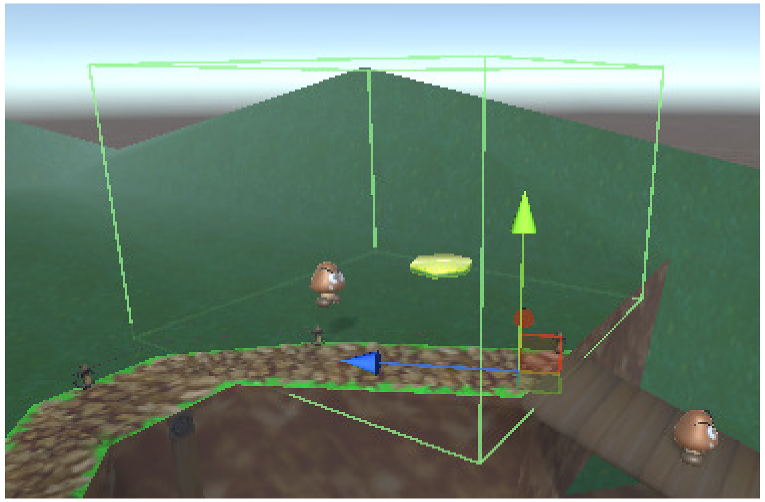
Image (Script) ✓

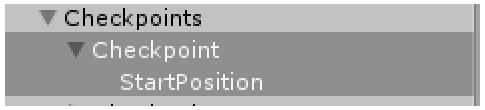
Source Image	HealthDisc
Color	🟩
Material	None (Material)
Raycast Target	✓
Image Type	Filled
Fill Method	Radial 360
Fill Origin	Top
Fill Amount	0.87
Clockwise	✓
Preserve Aspect	☐

GameController

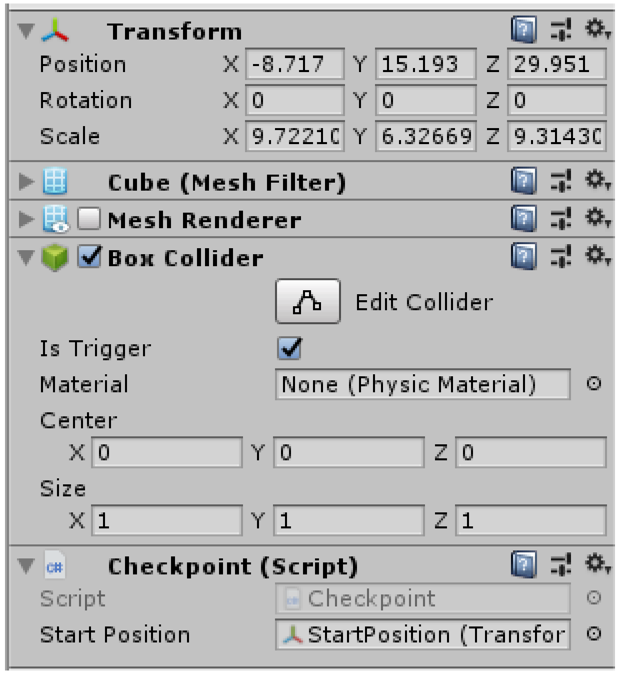
```
public class GameController : MonoBehaviour
{
    //...
    [Header("Animations")]
    public Animation m_Animation;
    public AnimationClip m_ShowHUDAnimationClip;
    public AnimationClip m_IdleHUDAnimationClip;

    public void UpdateCoins(bool SetAnimation=true)
    {
        m_CoinsText.text=m_PlayerController.GetCoins().ToString();
        if(SetAnimation)
            ShowHUD();
    }
    void ShowHUD()
    {
        AnimationState l_AnimationState=m_Animation[m_ShowHUDAnimationClip.name];
        if(!l_AnimationState.enabled || l_AnimationState.normalizedTime>=1.0f)
        {
            m_Animation.Stop();
            m_Animation.Play(m_ShowHUDAnimationClip.name);
        }
    }
}
```

Checkpoint





Checkpoints
Checkpoint
StartPosition



Transform
Position X -8.717 Y 15.193 Z 29.951
Rotation X 0 Y 0 Z 0
Scale X 9.72210 Y 6.32669 Z 9.31430

Cube (Mesh Filter)

Mesh Renderer

Box Collider
Edit Collider
Is Trigger ✓
Material None (Physic Material)
Center
X 0 Y 0 Z 0
Size
X 1 Y 1 Z 1

Checkpoint (Script)
Script Checkpoint
Start Position StartPosition (Transfor