Welcome to issue 12.

Welcome back to another game creating adventure. When you download the assets of this issue you will get the full open resource project that you can open in Unity. This way you can start right away and change/add things to your own likings. If you're new to Unity and C-Script I advise you to read and experiment with all previous issues.

In this Issue we will create a basic game of 1942. You can see this all in action right here : https://youtu.be/TWyCPI46hZk

Back in the days I played this arcade game for many hours. We will setup all the basics so you can make as many levels as you need. Since I am creating this magazines for one year now I will only explain parts of the scripts that are not basic. You should know what variables are right now. If not than read all the previous issues. I am still learning and this is where I am now, so if you read all previous issues we should be on the same level. Ready ? Lets check this out.

Camera Position script.

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

public class CamPositions : MonoBehaviour
{

/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///

private GameManager GM;
```

This script is given to the main camera. Every time the level starts it checks the RestartPoint number and sets its position according to it.

```
void Start()
{
GM = FindObjectOfType<GameManager>();
if (GM.RestartPoint == 1)
{
transform.position = new Vector3(0, 0, -10);
}
if (GM.RestartPoint == 2)
{
transform.position = new Vector3(0, 55.5f, -10);
}
if (GM.RestartPoint == 3)
{
transform.position = new Vector3(0, 109.3f, -10);
}
}
}
```

Don’t destroy Script.

When the level restarts we need some objects to stay so it can be used again. So a level should reset but keeps the important objects from being destroyed.

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

public class DD : MonoBehaviour
{

/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///
```

We use the awake void (function) as soon as the scripts awakes it checks if the instance (Script name) is set to null. We did set it like this with the DD instance = null line. When this is not the case we will set it to instance and otherwise we destroy it. Finally when the level scene reloads we will not destroy this game object. You can add this Awake function into other scripts or use it as a single script to give to gameObjects that should not be destroyed on reload scene.

```
public static DD instance = null;
void Awake()
{
if (instance == null)
{
instance = this;
}
else if (instance != this)
{
Destroy(gameObject);
}
DontDestroyOnLoad(gameObject);
}
}
```

Enemy Script.

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

public class Enemies : MonoBehaviour
{

/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///

public int EnemyType;
public float Speed;
private Transform Cammy;
private GameManager GM;
public AudioClip Explode;
```

The enemy game objects have a box 2d collider and a rigidbody2D with the Z restraint selected and the gravity set to 0. The box collider is set as a trigger. An enemyType number is used so you can add other types of enemies later with their own type numbers. When the trigger is a game object with the name tag bullet it will add one point to the hit variable and plays a sound near the camera position. The enemy and trigger object get destroyed.

```
private void OnTriggerEnter2D(Collider2D collision)
{
if (collision.gameObject.tag == ("Bullet") && EnemyType == 1)
{
AudioSource.PlayClipAtPoint(Explode, new Vector3(Cammy.position.x,
Cammy.position.y, -5));
GameManager.Hits += 1;
Destroy(collision.gameObject);
Destroy(this.gameObject);
}
}
```

To make sure the sounds are good to hear we use the camera position. In start we will give this a name and make it look for the camera using the tag name MainCamera.

```
void Start()
{
Cammy = GameObject.FindGameObjectWithTag("MainCamera").GetComponent<Transform>();
GM = FindObjectOfType<GameManager>();
}
```

When the enemy comes close in range of the camera (Closer than 6) it will speed up and moves faster down until it reach the point (Smaller than -12) where it gets destroyed.

```
void Update()
{
if (Vector2.Distance(transform.position, Cammy.position) < 6)
{
Speed = 2.5f;
}
if (GM.IsGameInPlay == true)
{
transform.Translate(Vector2.down * Speed * Time.deltaTime);
if (transform.position.y < -12)
{
Destroy(this.gameObject);
}
}
}
}
```

GameManager Script.

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


public class GameManager : MonoBehaviour
{

/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///

public bool IsGameInPlay = false;
private AudioSource Asource;
public static float Hits;
public static float HScoreValue;
public Text HScoreValueT;
public static float ScoreValue;
public Text ScoreValueT;
public static float LivesValue;
public static float LoopValue;
public GameObject Loop3, Loop2, Loop1, Lives3, Lives2, Lives1, GameOver;
public AudioClip GameO;
public bool GameOverNow;
public int RestartPoint;
public static GameManager instance = null;
private Transform Cammycam;
```

The game manager script is needed also when the level is reloaded. So we insert the awake function with the don’t destroy script.

```
void Awake()
{
if (instance == null)
{
instance = this;
}
else if (instance != this)
{
Destroy(gameObject);
}
DontDestroyOnLoad(gameObject);
}
```

We start up the usual elements and make it find the camera. Also it will load a Highscore value named Hscore by using PlayerPrefs.GetFloat (Float variable).

```
void Start()
{
GameOverNow = false;
Asource = GetComponent<AudioSource>();
Asource.Play();
LivesValue = 3;
LoopValue = 3;
GameOver.SetActive(false);
RestartPoint = 1;
Cammycam =
GameObject.FindGameObjectWithTag("MainCamera").GetComponent<Transform>();
PlayerPrefs.GetFloat("Hscore", HScoreValue);
}
```

In order to use the scene manager you need to add this line into the first library lines. Using UnityEngine.SceneManagement. The script uses also UI elements so another important line must be added : using UnityEngine.UI.

```
public void CompleteResetGame()
{
LivesValue = 3;
LoopValue = 3;
ScoreValue = 0;
GameOver.SetActive(false);
GameOverNow = false;
RestartPoint = 1;
Asource.Play();
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
}
```

A coroutine is made that when activated will reset the scene or activate all the game over stuff. Depending on the number of lives set in the livesValue variable. When needed the high score is saved. (SetFloat)

```
IEnumerator ResetNow()
{
yield return new WaitForSeconds(1f);
if (LivesValue == 0)
{
stopMusic();
GameOverNow = true;
GameOver.SetActive(true);
IsGameInPlay = false;
Cammycam =
GameObject.FindGameObjectWithTag("MainCamera").GetComponent<Transform>();
AudioSource.PlayClipAtPoint(GameO, new Vector3(Cammycam.position.x,
Cammycam.position.y, -10));
if (ScoreValue > HScoreValue)
{
HScoreValue = ScoreValue;
PlayerPrefs.SetFloat("Hscore", HScoreValue);

}
}
if (LivesValue != 0)
{
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
}
}


public void ResetLevel()
{
StartCoroutine(ResetNow());
}

public void stopMusic()
{
Asource.Stop();
}

public void ContinueLevel()
{
RestartPoint += 1;
Asource.Play();
IsGameInPlay = true;
}
```

The loop and live variables correspond with the images set on the UI canvas. Depending on the numbers the images are shown (SetActivate(true)) or not show (SetActivate(false))

```
void Update()
{
HScoreValueT.text = HScoreValue.ToString("HIGH-SCORE" + "\n" + "00000000");
if (GameOverNow == true && Input.GetKeyDown(KeyCode.R))
{
CompleteResetGame();
GameOverNow = false;
}
if (LoopValue == 3)
{
Loop3.SetActive(true);
Loop2.SetActive(true);
Loop1.SetActive(true);
}
if (LoopValue == 2)
{
Loop3.SetActive(false);
Loop2.SetActive(true);
Loop1.SetActive(true);
}
if (LoopValue == 1)
{
Loop3.SetActive(false);
Loop2.SetActive(false);
Loop1.SetActive(true);
}
if (LoopValue == 0)
{
Loop3.SetActive(false);
Loop2.SetActive(false);
Loop1.SetActive(false);
}
if (LivesValue == 3)
{
Lives3.SetActive(true);
Lives2.SetActive(true);
Lives1.SetActive(true);
}
if (LivesValue == 2)
{
Lives3.SetActive(false);
Lives2.SetActive(true);
Lives1.SetActive(true);
}
if (LivesValue == 1)
{
Lives3.SetActive(false);
Lives2.SetActive(false);
Lives1.SetActive(true);
}
if (LivesValue == 0)
```

```
{
Lives3.SetActive(false);
Lives2.SetActive(false);
Lives1.SetActive(false);
}
ScoreValueT.text = ScoreValue.ToString("SCORE" + "\n" + "0000000");
if (Input.GetKeyDown(KeyCode.Space) && IsGameInPlay == false)
{
IsGameInPlay = true;
}
}
}
```

Instead of the plane flying forward we make other objects slide down. When the IsGameInPlay bool variable in the game manager script is set to true the object slides down with the given speed until it reaches the point where it gets destroyed. (-12)

Move Down Script.

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

public class MoveDown : MonoBehaviour
{

/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///

public float Speed;
private GameManager GM;


void Start()
{
GM = FindObjectOfType<GameManager>();
}

void Update()
{
if (GM.IsGameInPlay == true)
{
transform.Translate(Vector2.down * Speed * Time.deltaTime);
if (transform.position.y < -12)
{
Destroy(this.gameObject);
}
}
}
}
```

Player Script.

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

public class Player : MonoBehaviour
{

/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///

public float ScoreHits;
public float MovingSpeed, MovingSpeed2;
private float Xmove;
private float Ymove;
public GameObject Bullets;
public Transform Barrel1, Barrel2, Barrel3;
public int MoveMode;
public Sprite Left, Straight, Right, Loop2, Loop3, Loop4, Loop5, Loop6;
public bool MakeLooping;
public bool TrippleBullets;
public Vector2 LandingZone;
public float Landingtime = 2f;
private GameManager GM;
public AudioClip Levelclear, Shoot, Loop, Explode;
private Transform Cammycam;

private void Start()
{
ScoreHits = 0;
Cammycam =
GameObject.FindGameObjectWithTag("MainCamera").GetComponent<Transform>();
GM = FindObjectOfType<GameManager>();
TrippleBullets = false;
MoveMode = 0;
MakeLooping = false;
LandingZone = new Vector2(Cammycam.position.x, Cammycam.position.y);
transform.position = new Vector3(Cammycam.position.x, Cammycam.position.y - 1, 0);
}
```

Depending on the trigger with each their own tag name actions are activated. So when hitting an enemy the objects gets destroyed and when landing is hit the move mode is set to 4.

```
private void OnTriggerEnter2D(Collider2D collision)
{
if (collision.gameObject.tag == ("Takeoff"))
{
MoveMode = 1;
}
if (collision.gameObject.tag == ("Pow"))
{
TrippleBullets = true;
Destroy(collision.gameObject);
}
if (collision.gameObject.tag == ("Landing"))
{
MoveMode = 4;
}
if (collision.gameObject.tag == ("Enemy"))
{
GameManager.LivesValue -= 1;
GM.ResetLevel();
AudioSource.PlayClipAtPoint(Explode, new Vector3(Cammycam.position.x,
Cammycam.position.y, -10)); ;
Destroy(this.gameObject);
}
}
```

Instead of using a real animation we fake one by using different images and make them play right after each other. While playing the collider is disabled until all images have played. All this is created in a coroutine that will start when the makeLooping variable is set to false and the Z key is pressed.

```
IEnumerator Looping()
{
GetComponent<BoxCollider2D>().enabled = false;
MakeLooping = true;
this.GetComponent<SpriteRenderer>().sprite = Straight;
yield return new WaitForSeconds(0.2f);
this.GetComponent<SpriteRenderer>().sprite = Loop2;
yield return new WaitForSeconds(0.2f);
this.GetComponent<SpriteRenderer>().sprite = Loop3;
yield return new WaitForSeconds(0.2f);
this.GetComponent<SpriteRenderer>().sprite = Loop4;
yield return new WaitForSeconds(0.2f);
this.GetComponent<SpriteRenderer>().sprite = Loop5;
yield return new WaitForSeconds(0.2f);
this.GetComponent<SpriteRenderer>().sprite = Loop6;
yield return new WaitForSeconds(0.2f);
this.GetComponent<SpriteRenderer>().sprite = Straight;
yield return new WaitForSeconds(0.2f);
GetComponent<BoxCollider2D>().enabled = true;
MakeLooping = false;
}
```

To make sure the plane stays in the given boundaries we use Mathf.Clamp and us positions form the camera so the plain is always in site. When movement on x-axis is smaller than 0 it uses the fly left image and when its bigger than 0 it uses the fly right image. When no movement on the x-axis takes place the straight image is used.

```
void Update()
{
Xmove = Input.GetAxisRaw("Horizontal");
Ymove = Input.GetAxisRaw("Vertical");
if (MoveMode == 4 && Landingtime > 0)
{
float step = MovingSpeed * Time.deltaTime;

transform.position = Vector2.MoveTowards(transform.position, LandingZone,
MovingSpeed2 * Time.deltaTime);
Landingtime -= Time.deltaTime;
if (transform.position.x < 0)
{
this.GetComponent<SpriteRenderer>().sprite = Right;
}
if (transform.position.x > 0)
{
this.GetComponent<SpriteRenderer>().sprite = Left;
}
if (transform.position.x == 0)
{
this.GetComponent<SpriteRenderer>().sprite = Straight;
}
if (Landingtime <= 0)
{
GM.stopMusic();
AudioSource.PlayClipAtPoint(Levelclear, new Vector3(Cammycam.position.x,
Cammycam.position.y, -10));
MoveMode = 0;
GameManager.ScoreValue += 250;
ScoreHits = GameManager.Hits * 100;
GameManager.ScoreValue = GameManager.ScoreValue + ScoreHits;
GameManager.LoopValue = 3;
GM.IsGameInPlay = false;
Landingtime = 2;
TrippleBullets = false;
GM.Invoke("ContinueLevel", 15);
}
}
```

```
if (MoveMode == 1)
{
if (Xmove < 0 && MakeLooping == false)
{
this.GetComponent<SpriteRenderer>().sprite = Left;
}
if (Xmove > 0 && MakeLooping == false)
{
this.GetComponent<SpriteRenderer>().sprite = Right;
}
if (Xmove == 0 && MakeLooping == false)
{
this.GetComponent<SpriteRenderer>().sprite = Straight;
}
if (Input.GetKey(KeyCode.Z) && MakeLooping == false && GameManager.LoopValue > 0)
{
StartCoroutine(Looping());
AudioSource.PlayClipAtPoint(Loop, new Vector3(Cammycam.position.x,
Cammycam.position.y, -10));
GameManager.LoopValue -= 1;
}
if (Input.GetKeyDown(KeyCode.Space) && MakeLooping == false)
{
AudioSource.PlayClipAtPoint(Shoot, new Vector3(Cammycam.position.x,
Cammycam.position.y, -10));
Instantiate(Bullets, Barrel1.position, Quaternion.identity);
Instantiate(Bullets, Barrel2.position, Quaternion.identity);
if (TrippleBullets == true)
{
Instantiate(Bullets, Barrel3.position, Quaternion.identity);
}
}
Vector3 moveDir = new Vector3(Xmove, Ymove).normalized;
transform.position += moveDir * MovingSpeed * Time.deltaTime;
transform.position = new Vector3(Mathf.Clamp(transform.position.x, -4, 4),
Mathf.Clamp(transform.position.y, Cammycam.position.y - 3, Cammycam.position.y), 0);
}
}
}
```

The bullets go up until they reach the point outside the camera view. Outside it gets destroyed. The bullet object will look for the camera at start.

Bullet Script.

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

public class PlayerBullet : MonoBehaviour
{
/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///

public float Speed;
private Transform Cammycam;

private void Start()
{
Cammycam =
GameObject.FindGameObjectWithTag("MainCamera").GetComponent<Transform>();
}

void Update()
{
transform.Translate(Vector2.up * Speed * Time.deltaTime);
if (transform.position.y > Cammycam.position.y + 5f)
{
Destroy(this.gameObject);
}
}
}
```

The power object is a trigger for the player. Once picked up it will give the player to fire 3 bullets at once instead of 2 by setting the tripplebullets bool variable to true.

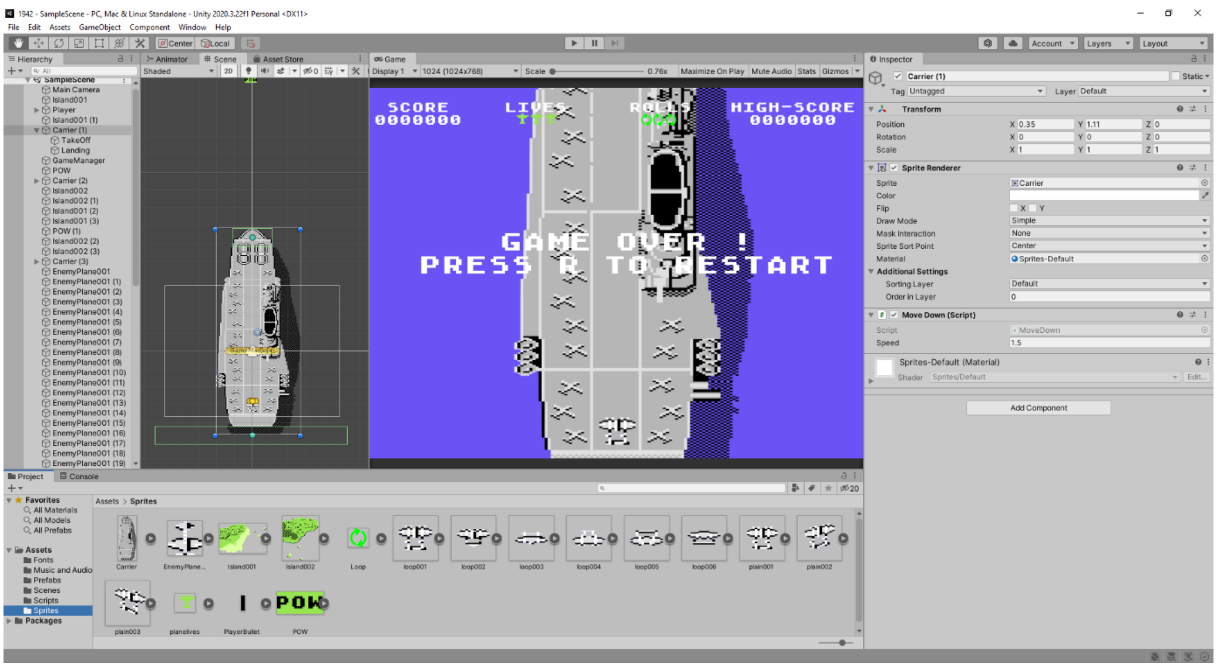
Powerup Script.

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

public class Pow : MonoBehaviour
{
/// 1942 retro game by René Pol Issue 12 March 2022 Create Games///

public float Speed;
private GameManager GM;

void Start()
{
GM = FindObjectOfType<GameManager>();
}
```

```
void Update()
{
if (GM.IsGameInPlay == true)
{
transform.Translate(Vector2.down * Speed * Time.deltaTime);
if (transform.position.y < -12)
{
Destroy(this.gameObject);
}
}
}
}
```

The carrier gameObject have 2 empty game objects with collision boxes set to trigger. One for landing and the other for takeoff. Depending on which one the player hits it will set the players MoveMode according to it. When landed the next level will start.

There you have it. All the basic stuff needed to create your own levels. These scripts are giving you a good head start so happy game making ! As this is a vertical scroller game you should be able to turn it into a Horizontal one.



1942 - SampleScene - PC, Mac & Linux Standalone - Unity 2020.3.22f1 Personal <DX11>

SampleScene, Main Camera, Island001, Player, Island001 (1), Carrier (1), TakeOff, Landing, GameManager, POW, Carrier (2), Island002, Island002 (1), Island001 (2), Island001 (3), POW (1), Island002 (2), Island002 (3), Carrier (3), EnemyPlane001, EnemyPlane001 (1), EnemyPlane001 (2), EnemyPlane001 (3), EnemyPlane001 (4), EnemyPlane001 (5), EnemyPlane001 (6), EnemyPlane001 (7), EnemyPlane001 (8), EnemyPlane001 (9), EnemyPlane001 (10), EnemyPlane001 (11), EnemyPlane001 (12), EnemyPlane001 (13), EnemyPlane001 (14), EnemyPlane001 (15), EnemyPlane001 (16), EnemyPlane001 (17), EnemyPlane001 (18), EnemyPlane001 (19)

Carrier (1)
Tag Untagged
Layer Default

Transform
Position X 0.35 Y 1.11 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Sprite Renderer
Sprite: Carrier
Flip: X Y
Draw Mode: Simple
Mask Interaction: None
Sprite Sort Point: Center
Material: Sprites-Default
Additional Settings
Sorting Layer: Default
Order in Layer: 0

Move Down (Script)
Script: MoveDown
Speed: 1.5

Sprites-Default (Material)
Shader: Sprites/Default

HOW TO

What if I need a wandering NPC ? So no path points but just a game object wandering around in any direction ?

You can see this all working at : https://youtu.be/qFmENf9fOBE

The game object has box collider and a rigid body. The name tag is Crawler. When you give it this script it will move forward for a certain time, then stops and rotates until it starts moving again. When the objects hits an other crawler or an object with the name tag Wall it will rotate 180 degrees in the opposite direction.

Wandering NPC script.

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

public class Wandering : MonoBehaviour
{

/// Wandering NPC by René Pol Issue 12 March 2022 Create Games///


public float speed;
public float RotateSpeed;

public int MoveMode;
public float Crawlingtime;
public float Rotatingtime;

private Animator Anim;
```

The game object uses animations that came with it so we get the Animator Component. The rotation is set in a random direction at start. Move mode is set to 1 and the crawling + Rotating time are set to a random number.

```
void Start()
{
Anim = GetComponent<Animator>();
MoveMode = 1;
Crawlingtime = 3 + Random.Range(1, 4);
Rotatingtime = 5 + Random.Range(1, 4);
transform.Rotate(0, Random.Range(0, 360), 0);
}
```

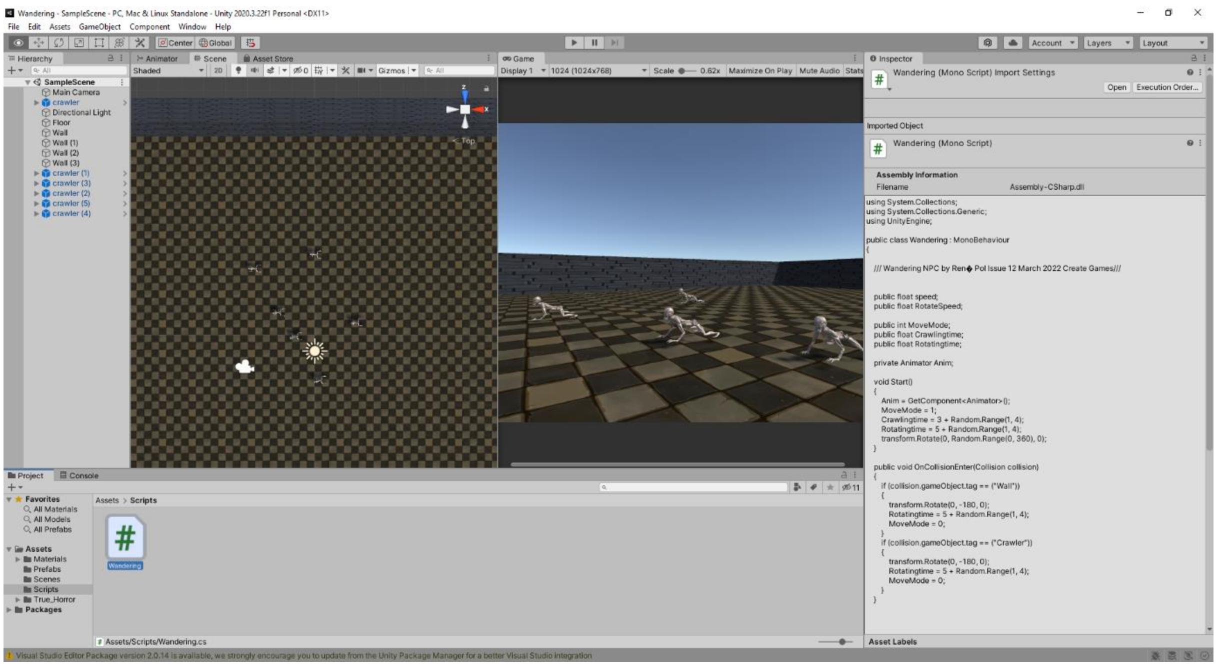
When an object with the tag Wall is hit the gameobject will rotate 180 degrees and start the rotating time while move mode is set to 0. The same happens when an object with the tag Crawler is hit.

```
public void OnCollisionEnter(Collision collision)
{
if (collision.gameObject.tag == ("Wall"))
{
transform.Rotate(0, -180, 0);
Rotatingtime = 5 + Random.Range(1, 4);
MoveMode = 0;
}
if (collision.gameObject.tag == ("Crawler"))
{
transform.Rotate(0, -180, 0);
Rotatingtime = 5 + Random.Range(1, 4);
MoveMode = 0;
}
}
```

If Crawling time hits 0 the move mode will be set to 0 and the Crawling time is reset. Movement mode 0 makes the game object rotate until the rotation time hits 0. Then the Movement mode is set back to 1 and the rotation time is reset.

```
void Update()
{
if (MoveMode == 0)
{
Rotatingtime -= Time.deltaTime;
transform.Rotate(0, 25 * RotateSpeed * Time.deltaTime, 0);
Anim.Play("pounce");
if (Rotatingtime <= 0)
{
MoveMode = 1;
Rotatingtime = 5 + Random.Range(1, 4);
}
}
if (MoveMode == 1)
{
Crawlingtime -= Time.deltaTime;
transform.Translate(Vector3.left * speed * Time.deltaTime);
Anim.Play("crawl");
if (Crawlingtime <= 0)
{
MoveMode = 0;
Crawlingtime = 3 + Random.Range(1, 4);
}
}
}
}
```

You can change the duration of crawling and rotating by changing the random numbers. Speed of movement can be set in the inspection view. This is ideal to make animals or humans just walk around without the need of path points. Happy game creating and I'll see you next issue.



Assembly Information

Filename Assembly-CSharp.dll

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

public class Wandering : MonoBehaviour
{

    /// Wandering NPC by Ren� Pol Issue 12 March 2022 Create Games///


    public float speed;
    public float RotateSpeed;

    public int MoveMode;
    public float Crawlingtime;
    public float Rotatingtime;

    private Animator Anim;

    void Start()
    {
        Anim = GetComponent<Animator>();
        MoveMode = 1;
        Crawlingtime = 3 + Random.Range(1, 4);
        Rotatingtime = 5 + Random.Range(1, 4);
        transform.Rotate(0, Random.Range(0, 360), 0);
    }

    public void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.tag == ("Wall"))
        {
            transform.Rotate(0, -180, 0);
            Rotatingtime = 5 + Random.Range(1, 4);
            MoveMode = 0;
        }
        if (collision.gameObject.tag == ("Crawler"))
        {
            transform.Rotate(0, -180, 0);
            Rotatingtime = 5 + Random.Range(1, 4);
            MoveMode = 0;
        }
    }
```

Visual Studio Editor Package version 2.0.14 is available, we strongly encourage you to update from the Unity Package Manager for a better Visual Studio integration