2D PLAYER AND ENEMY MOVEMENT

Player and Enemy Movement components can be broken down into these properties:

Enemy Detect/Target - Enemy A.I. targets Player Asset with 'Player' Tag.
Enemy Patrol – Enemy Patrols left to right … stops at the edge of the platform and rotates 180 degrees and travels the other way.
Player Attack – Player is able to Instantiate bullets and fire at target/object.
Player Movement – Player moves left to right and double Jumps.
Player Gun Movement – Player Gun rotates 360 degrees, following the Player's arrow cursor.
Player 'Dash' – Brief interludes of Player speed.
Player Respawn – Player falls off platform, they Respawn at game object…

Setup Section …
Open Unity 2D and open generic asset shapes… (the scene is comprise out of various squares) Apply a Box Collider 2D to all of the objects in the scene and Apply a RigidBody 2D to the Player this will help it to move …

Patrolling state

If A.I. detects edge of platform using the 'RayCast2D'; which is generated by a game object in front of it … it will rotate 180 degrees and travel in the other direction.

Patrol Movement;

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

public class PatrolEnemy : MonoBehaviour
{
   public float Speed;
   public float Distance;

   public bool movingRight = true;

   public Transform checkGround;

   void Update()
   {
      transform.Translate(Vector2.right * Speed * Time.deltaTime);

      RaycastHit2D groundInfo = Physics2D.Raycast(checkGround.position, Vector2.down, 2f);
      if (groundInfo.collider == false)
      {
         if(movingRight == true)
         {
            transform.eulerAngles = new Vector3(0, -180, 0);
            movingRight = false;
         }
         else
```

```
        {
            transform.eulerAngles = new Vector3(0, 0, 0);
            movingRight = true;
        }
    }

  }
}
```

Enemy Detect State

Locate and Target Player Tag – Enemy movement.

Enemy Movement;

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

public class Enemy : MonoBehaviour
{


  public float Speed;

  private Transform target;

  void Start()
  {

    target = GameObject.FindGameObjectWithTag("Player").GetComponent<Transform>();
  }

  void Update()
  {
    if (Vector2.Distance(transform.position, target.position) > 2)
    {
      transform.position = Vector2.MoveTowards(transform.position, target.position, Speed *
Time.deltaTime);
    }
  }
}
```

Player Attack State

Player is able to aim range weapon in any direction … It follows cursor, Instantiates bullets and fires projectile in any given location…

Weapon Movement;

```
using System.Collections;
```

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

public class Weapon : MonoBehaviour
{
   public float offset;

   public GameObject projectile;
   public Transform shotPoint;

   private float time;
   public float start;


   private void Update()
   {
      Vector3 difference = Camera.main.ScreenToWorldPoint(Input.mousePosition) -
transform.position;
      float rotZ = Mathf.Atan2(difference.y, difference.x) * Mathf.Rad2Deg;
      transform.rotation = Quaternion.Euler(0f, 0f, rotZ + offset);

      if (time <= 0)
      {
         if (Input.GetMouseButtonDown(0))
         {
            Instantiate(projectile, shotPoint.position, transform.rotation);
            time = start;
         }
      }

      else
      {
         time -= Time.deltaTime;
      }

   }
}
```

Firing / Bullet Code Instantiation;

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

public class Bullet : MonoBehaviour
{
   public float speed;
   public float life;


   private void Start()
```

```
    {
        Invoke("DestroyProjectile", life);
    }

    private void Update()
    {
        transform.Translate(transform.up * speed * Time.deltaTime);
    }

}
```

Player Movement State

Player is able to move side to side … left to right and double jump to reach objective…

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

public class PlayerController : MonoBehaviour
{
    public float Speed;
    public float Jump;
    private float Move;

    private Rigidbody2D rb;

    private bool Ground;
    public Transform CheckGround;
    public float CheckRadius;
    public LayerMask WhatIsGround;

    private int JumpExtra;
    public int ExtraJumpValue;

    private void Start()
    {
        JumpExtra = ExtraJumpValue;
        rb = GetComponent<Rigidbody2D>();
    }
    private void FixedUpdate()
    {
        Ground = Physics2D.OverlapCircle(CheckGround.position, CheckRadius, WhatIsGround);
        Move = Input.GetAxis("Horizontal");
        rb.velocity = new Vector2(Move * Speed, rb.velocity.y);
    }
    private void Update()
    {
            if (Ground == true)
            {
                JumpExtra = ExtraJumpValue;
```

```
            }
            if (Input.GetKeyDown(KeyCode.UpArrow) && JumpExtra > 0)
            {
                rb.velocity = Vector2.up * Jump;
                JumpExtra--;
            }
            else if (Input.GetKeyDown(KeyCode.UpArrow) && JumpExtra == 0 && Ground == true)
            {
                rb.velocity = Vector2.up * Jump;
            }
        }
    }
```

Player Dash State

Player is able to move at an increased speed, only for a small increments at a time …

Player Dash Movement;

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

public class DashMove : MonoBehaviour
{
    private Rigidbody2D rb;
    public float dashSpeed;
    private float dashTime;
    public float startDashTime;
    private int direction;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        dashTime = startDashTime;
    }


    void Update()
    {
        if (direction == 0)
        {
            if (Input.GetKeyDown(KeyCode.Q))
            {
                direction = 1;
            }
            else if (Input.GetKeyDown(KeyCode.E))
            {
                direction = 2;
            }
        }
```

```
        else
        {
            if (dashTime <= 0)
            {
                direction = 0;
            }
            else
            {
                dashTime -= Time.deltaTime;

                if (direction == 1)
                {
                    rb.velocity = Vector2.left * dashSpeed;
                }

                else if (direction == 2)
                {
                    rb.velocity = Vector2.right * dashSpeed;
                }
            }

        }
    }
}
```

Player Respawn

Player falls off platform and is caught by a BoxCollider 2D, it is essentially teleported to an empty Game Object back onto the Platform…

Respawn;

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

public class Respawn : MonoBehaviour
{
    public Transform Player;
    public Transform respawnPoint;

    void OnTriggerEnter2D(Collider2D collision)
    {
        Player.transform.position = respawnPoint.transform.position;
    }


}
```