PROGRESS REPORT

MICROGAME #3: Shmup

Andrew Adame

ID: 007516100

LEGEND: COMPLETED – UNFINISHED – WIP – FIX - FIXED

GITHUB: https://github.com/andrewadame/UnityProjectsCSE-4410/tree/master/ShmupProject

UNITY PLAY: https://play.unity.com/mg/other/shmuppub

1. Create new project Shmup
2. Create folders containing important assets (scripts, prefabs, animation, etc)
3. Create a basic Shmup Game
    a. Design Level
        i. Utilize tile maps
            1. Used provided sprites
        ii. ERROR
            1. Player not bound to play space
    b. Player
        i. Used provided plane spritePlayer-Controlled
            a. Used script “PlyrCtrlr”
        ii. Preferred Physics
            1. Responsive with some drift
        iii. Projectiles
            1. Fires projectiles “Blt” upon pressing SPACE
            2. Fires at a set rate
            3. Destroys enemy on hit
            4. Spawns from sides of player plane
        iv. Health
            1. Health bar that goes down every time Player is hit by enemy projectiles “EnBlt”
            2. Utilizes provided heart sprite for visualization
            3. Player take damage if bump is made to Enemy
            4. Short cooldown where player isn’t damaged after taking damage (Invincibility Frames)
    c. Enemies
        i. Movement
            1. Lightly follows Player movement

- ii. Enemy Hierarchy
    1. "En1" shoots slower than "En2"
- iii. Attack Player
    1. Fires projectiles at random
    2. Projectiles spawn at center of En
    3. Cooldown between each shot

d. Visuals
- i. Sprites provided by instructor
- ii. Animation
    1. Animation Script
        a. Called "BmCtrlr"
    2. Muzzle flashes when player is firing projectiles
        a. VISUAL ERROR
            i. Muzzle flash looks odd due to playing in middle of plane rather than at projectile spawns
    3. Enemies explode on death

e. Gameplay
- i. Game Script
    1. Called "GmeCtrlr
- ii. Start
    1. Player spawns, Enemies begin to spawn after 1 second
- iii. Live Tracking
    1. Default amount of lives is 10
- iv. Objective
    1. Survive as long as possible
- v. Point System
    1. Player earns points for every Enemy shot down
    2. Player looses points for every Enemy making it pass Player
- vi. Enemy Spawn System
    1. Enemy spawns consistently outside bounds of game
- vii. Game Over
    1. Game ends upon loosing all 10 lives

f. UI
- i. Player Healthbar
    1. Represented as red heart sprites
    2. Disappears one by one with every damage taken
- ii. Score Tracking
- iii. Game Over
    1. Stops game with screen that displays the message "Game Over! Press Any Key to Restart!"

g. **EXTRA**

i. Audio
ii. More Levels
iii. Original sprites
iv. Animation overhaul
v. Powerups
    1. Missiles?
    2. Bombs?
    3. More Projectiles?
vi. Larger Hierarchy of Enemies
vii. Introduce both air-to-ground enemies (Inspiration: Xevious)

SCRIPTS

PlyrCtrlr

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

public class PlyrCtrlr : MonoBehaviour
{

    [Header("Starting states")]
    public float speed;
    Vector2 input;
    Rigidbody2D plyrRgdBdy;

    [Header("Shooting")]
    public GameObject blt;
    public GameObject[] bltSpwnPos;
    private float cools;
    public float TmeBtwnShts;

    [Header("Health")]
    public int maxHlth = 10;
    [SerializeField]
    private int Hlth;

    public GameObject HlthImg;
    public GameObject hlthPrnt;
    public float tmeBtwnHrt = 0.3f;
    float iframe;

    public GameObject flsh;

    // Start is called before the first frame update
    void Start()
    {
        plyrRgdBdy = GetComponent<Rigidbody2D>();
        cools = TmeBtwnShts;
        Hlth = maxHlth;
        iframe = tmeBtwnHrt;
        for(int i = 0; i < Hlth - 1; i++)
        {
```

```
            AddHrt();
        }
    }

    void AddHrt()
    {
        GameObject hrt = Instantiate(HlthImg);
        hrt.transform.SetParent(hlthPrnt.transform);
    }

    void RmveHrt(int n)
    {
        if(hlthPrnt.transform.childCount > 0)
        {
            if(hlthPrnt.transform.childCount < n)
            {
                n = hlthPrnt.transform.childCount;
            }

            for(int i = 0; i < n; i ++)
            {
                Destroy(hlthPrnt.transform.GetChild(0).gameObject);
            }
        }
    }

    // Update is called once per frame
    void Update()
    {
        input = new Vector2(Input.GetAxis("Horizontal"), Input.GetAxis("Vertical"));
        plyrRgdBdy.AddForce(input * speed * Time.deltaTime);

        if(Input.GetKey(KeyCode.Space) && cools <= 0)
        {
            Shoot();
        }

        if (cools > 0)
        {
            cools -= Time.deltaTime;
        }

        if(iframe > 0)
        {
            iframe -= Time.deltaTime;
        }
    }

    void Shoot()
    {
        for(int i = 0; i < bltSpwnPos.Length; i++)
        {
            Instantiate(blt, bltSpwnPos[i].transform.position, Quaternion.identity);
        }

        Instantiate(flsh, transform.position, Quaternion.Euler(0, 0, 0));

        cools = TmeBtwnShts;
```

```
    }

    public void TkeDmg(int dmg)
    {
        if (iframe <= 0)
        {
            RmveHrt(dmg);
            Hlth = Hlth - dmg;
            if (Hlth <= 0)
            {
                GmeOvr();
            }
            iframe = tmeBtwnHrt;
        }
    }

    void GmeOvr()
    {
        FindObjectOfType<GmeCtrlr>().gmeOvr = true;
        FindObjectOfType<GmeCtrlr>().gameOverUI.SetActive(true);
        gameObject.SetActive(false);
        Time.timeScale = 0f;
    }

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if(collision.gameObject.CompareTag("Enemy"))
        {
            TkeDmg(1);
            Destroy(collision.gameObject);
        }
    }
}
```

GmeCtrlr

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

public class GmeCtrlr : MonoBehaviour
{

    public GameObject[] enemies;
    public float tmeBtwnSpwnLw = 0.5f;
    public float timeBtwnSpwnHi = 3f;

    float spwnCls;
    Vector2 bnds;
    Vector3 spwnPos;

    public Text screTxt;
    int scres = 0;
```

```
    public GameObject gameOverUI;
    public bool gmeOvr;

    // Start is called before the first frame update
    void Start()
    {
        bnds = Camera.main.ScreenToWorldPoint(new Vector3(Screen.width, Screen.height,
0f));
        spwnCls = Random.Range(tmeBtwnSpwnLw, timeBtwnSpwnHi);
        screTxt.text = "Scores: " + scres.ToString();
        gmeOvr = false;
    }

    // Update is called once per frame
    void Update()
    {
        if(spwnCls > 0)
        {
            spwnCls -= Time.deltaTime;
        }
        else
        {
            SpwnEn();
        }

        screTxt.text = "Scores: " + scres.ToString();

        if (gmeOvr && Input.anyKeyDown)
            Restart();
    }

    void SpwnEn()
    {
        spwnPos = new Vector3(Random.Range(-bnds.x + 1f, bnds.x - 1f), bnds.y +
Random.Range(0.25f, 3f), 0f);
        Instantiate(enemies[Random.Range(0, enemies.Length)], spwnPos,
Quaternion.Euler(0,0,180));
        spwnCls = Random.Range(tmeBtwnSpwnLw, timeBtwnSpwnHi);
    }

    public void AddScre(int amount)
    {
        scres += amount;
    }

    void Restart()
    {
        SceneManager.LoadScene("SampleScene");
        Time.timeScale = 1f;
    }
}
```

EnCtrlr

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

```
using UnityEngine;

public class EnCtrlr : MonoBehaviour
{

    Rigidbody2D enemyRgdBdy;
    PlyrCtrlr Plyr;
    public float xSpeed, ySpeed;

    public GameObject blt;
    public float tmeBtwnAtckLw = 0.5f;
    public float tmeBtwnAttckHi = 2f;

    float attckCls;

    public int maxEnHlth = 2;
    private int EnHlth;

    GmeCtrlr cont;
    public int amount;
    Vector2 bounds;

    public GameObject explsn;

    // Start is called before the first frame update
    void Start()
    {
        bounds = Camera.main.ScreenToWorldPoint(new Vector3(Screen.width, Screen.height,
0f));
        enemyRgdBdy = GetComponent<Rigidbody2D>();
        Plyr = FindObjectOfType<PlyrCtrlr>();

        attckCls = Random.Range(tmeBtwnAtckLw, tmeBtwnAttckHi);

        EnHlth = maxEnHlth;
        cont = FindObjectOfType<GmeCtrlr>();
    }

    // Update is called once per frame
    void Update()
    {
        float x = 0f;
        if (Plyr != null)
        {
            if (Plyr.transform.position.x > transform.position.x) //enemy go left
            {
                x = xSpeed;
            }
            else if (Plyr.transform.position.x < transform.position.x) //enemy go right
            {
                x = -xSpeed;
            }
        }

        enemyRgdBdy.AddForce(new Vector2(x, -ySpeed) * Time.deltaTime);

        if (attckCls > 0)
        {
```

```
            attckCls -= Time.deltaTime;
        }
        else
        {
            Attck();
        }

        if(transform.position.y < -bounds.y)
        {
            cont.AddScre(-amount);
            Destroy(gameObject);
        }
    }

    void Attck()
    {
        Instantiate(blt, transform.position, transform.rotation);
        attckCls = Random.Range(tmeBtwnAtckLw, tmeBtwnAttckHi);
    }

    public void TkeDmg(int dmg)
    {
        EnHlth -= dmg;
        if(EnHlth <= 0)
        {
            Die();
        }
    }

    public void Die()
    {
        cont.AddScre(amount);
        Instantiate(explsn, transform.position, Quaternion.Euler(0,0,0));
        Destroy(gameObject);
    }
}
```

bltCtrlr

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

public class bltCtrlr : MonoBehaviour
{

    Rigidbody2D bltRgdBdy;
    public float speed;

    public int dmg = 1;

    // Start is called before the first frame update
    void Awake()
    {
        bltRgdBdy = GetComponent<Rigidbody2D>();
    }
```

```
    private void OnEnable()
    {
        bltRgdBdy.AddForce(Vector2.up * speed);
        Invoke("Disable", 5f);
    }

    private void Disable()
    {
        Destroy(gameObject);
    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.CompareTag("Enemy"))
        {
            collision.gameObject.GetComponent<EnCtrlr>().TkeDmg(dmg);
            Invoke("Disable", 0.001f);
        }
    }
}
```

EnBltCtrl

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

public class EnBltCtrl : MonoBehaviour
{
    Rigidbody2D bltRgdBdy;
    public float speed;
    public int dmg = 1;

    // Start is called before the first frame update
    void Awake()
    {
        bltRgdBdy = GetComponent<Rigidbody2D>();
    }

    private void OnEnable()
    {
        bltRgdBdy.AddForce(-Vector2.up * speed);
        Invoke("Disable", 5f);
    }

    private void Disable()
    {
        Destroy(gameObject);
    }
```

```
    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.gameObject.CompareTag("Player"))
        {
            collision.gameObject.GetComponent<PlyrCtrlr>().TkeDmg(dmg);
            Invoke("Disable", 0.001f);
        }
    }
}
```

BmCtrlr

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

public class BmCtrlr : MonoBehaviour
{
    public AnimationClip clip;

    private void OnEnable()
    {
        Invoke("Disable", clip.length);
    }

    private void Disable()
    {
        Destroy(gameObject);
    }
}
```