

## Reflection

As in my reflection of Neon Shadow Drift, game that I have created for this module, I enjoy making it at the beginning of procession of the game itself. But slowly I run out of time or spend too much time on one specification task in hand like the levels for example., my main role for this project was almost “everything types of guy” such as designer, developer and even as a tester. For most of the endless runner games out there I wanted this game to become a quite fast, speedy and up of style as like up-to-date mobile game that new players might enjoy. Therefore, I build-up a new game called Neon Shadow Drift, it set up in a neon-lit city where the player drift as a mechanics that feels good and the procession skill system that makes the player want to keep on going as that is a dream for this game to the final game.

This wasn't all type of straightforward tasks as for example having the car to go left to right, making the road ahead goes toward the player (like the car is moving forward), the biggest problem was to make the game itself as a balance for the endless that might be fun to play. Another point to note is that how to make the player plays more then three or five minutes without getting themselves quite bored or even stop the game itself. As in developing my game I mostly realised on in my unity program, this was the wrong type of unity, when log-in to the program there is an error of missing files as well. Basically, to sort of this I have reinstalled the university recommending version and I have restarted the basics again like redefining the maps, get the assets from the unity website store as well. Plus, I reset the project as it is semi broken therefore, I start as refresh mode like get the game so I can be able to use it either at class or at home. One of the problems was at home was: I sorted it out on Friday by rewatching this YouTube as I didn't understand as the last time I watch it. <https://www.youtube.com/watch?v=Ldyw5IFkEUQ&t=74s>. which I able to sort it out by going step by step in that video. For this, I need to simplify, reconsider, and test repeatedly. I have many frustrate many days and sleepless nights to make this work out as tweaking physics, try to fix the right version or even get the car/road to work.

Also, I was sick for a long time for at least two weeks, where I was having some surgery done that, I can't do any work for this Neon Shadow Drift as a whole. As I miss out getting lots of different types of drone mesh fbx again with the script and tidy some levels to make it nice and polish up for game itself,

Things that I have learn during this module,

Having the road tile mapping where the road travelling toward the player,

```
public class MoveTowardsPlayer : MonoBehaviour
{
    public float speed = 10f; // speed of world movement
    public float resetZ = -20f; // when behind player
    public float forwardZ = 80f; // where to reposition ahead

    void Update()
    {
        // Move the segment toward the player (negative Z)
        transform.Translate(Vector3.back * speed * Time.deltaTime);

        // If the segment goes behind the player, reposition it
        if (transform.position.z < resetZ)
        {
            Vector3 newPos = transform.position;
```

```

        newPos.z += forwardZ * 2f; // depends on how long your segments are
        transform.position = newPos;
    }
}

```

Having the road to generating

```

public class GenrateLevel : MonoBehaviour
{
    public GameObject[] Road;
    public int zPos = 50;
    public bool creatingRoad = false;
    public int secNum;

    // Start is called before the first frame update
    void Start()
    {
        if (creatingRoad == false)
        {
            creatingRoad = true;
            StartCoroutine(GenerateRoad());
        }
    }
    IEnumerator GenerateRoad()
    {
        secNum = Random.Range (0,3);
        Instantiate(Road[secNum], new Vector3(0,0,zPos),Quaternion.identity );
        zPos += 50;
        yield return new WaitForSeconds(2);
        creatingRoad=false;
    }

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

```

Car (without the camera)

```

using UnityEngine;

public class PlayerController : MonoBehaviour
{
    [Header("Movement Settings")]
    public float moveSpeed = 10f; // How fast the player slides side to
side
    public float horizontalLimit = 3f; // Max distance from center (like road
width)
    public float smoothing = 5f; // How quickly the player's movement
smooths out

    private float targetX; // Desired x-position

```

```

void Update()
{
    HandleInput();
    MoveSmoothly();
}

void HandleInput()
{
    // Get input (-1 for A, +1 for D)
    float input = 0f;
    if (Input.GetKey(KeyCode.A)) input = -1f;
    if (Input.GetKey(KeyCode.D)) input = 1f;

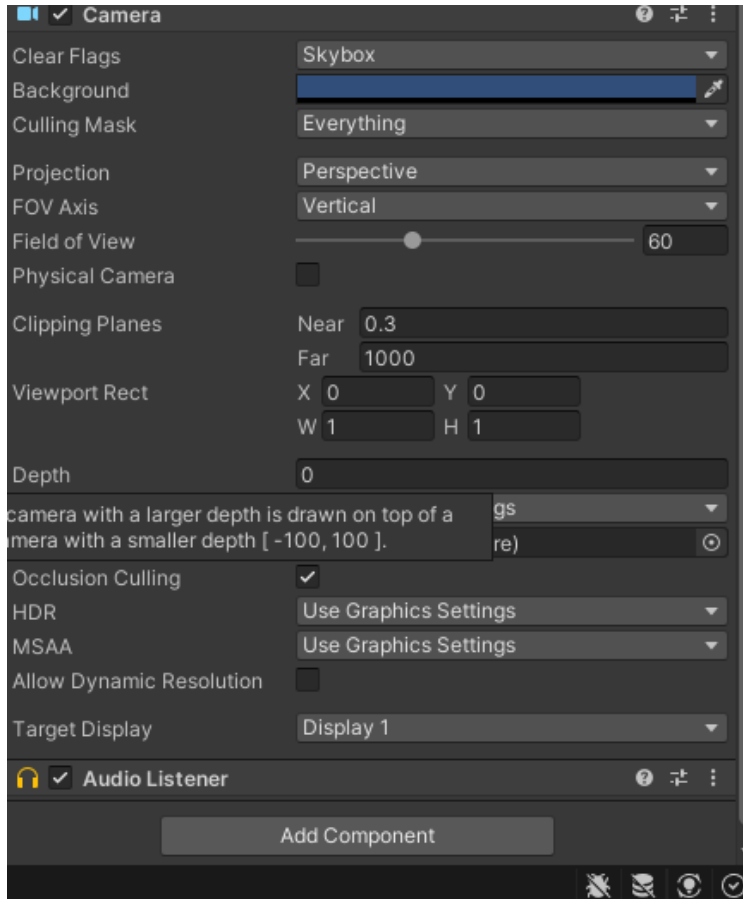
    // Update target position based on input
    targetX += input * moveSpeed * Time.deltaTime;

    // Clamp to road boundaries
    targetX = Mathf.Clamp(targetX, -horizontalLimit, horizontalLimit);
}

void MoveSmoothly()
{
    // Smoothly move toward the target X position like a car drifting
    Vector3 pos = transform.position;
    pos.x = Mathf.Lerp(pos.x, targetX, Time.deltaTime * smoothing);
    transform.position = pos;
}
}

```

Car (with the camera)



## Things that I wanted to be Added or Wishlist for Neon Shadow Drift

- Drone = having a drone that goes toward the player (the player must avoid the drone search light attached to it) also randomising sector in two levels such as warehouse and city that has a spawn timing, preventing all the drones from spawning at once or many different times for a more organic effect. it's adapted specifically to handle AI drone spawning for into my game. If the player got detected by the drone it chases the player like different types of colours (green(normal), orange (detect)and red (chasing the player))
- Add music tracks
- Main menu = new game, save, load, setting, sound
- Garage = update the player car with the power-up system from basic to expert, with stats, details of the updates, different types of colours for the car. Different cars to choice from beginner to expert.
- Achievements
- Sound = car going forward, drone searching, noise, traffic
- Hazard zone = having different types of hazard zone like police barriers, cones and fences, (more added later)
- Car = drifting visual like the tire's wheels are burning, smoking etc
  - Physic: tuning more able to hand well, responsive feel,
  - Vehicle: damages visual cracks, sparks or vibrating of the phone when hit on the siding etc
- Add more levels (that is regenerate randomly or been set up from the beginning to the ending of the game itself)
- Audio = radio chatting, drones chatting to each other, button clicking (by the player)
- Destruction System = links with the hazard zone, as well as breakable rooms that the car goes through in. cyber room, meeting room, server, more add later
- Score system – haptic feedback vibration for hits, boost, score multiplying for abilities timing
- Levels sectors =
  - city complex crossing, traffic and more drones
  - warehouse sector – indoor visuals, laser grids, forklifts
  - highway sector – above of the city more different layouts of roofs, areas of environment come and goes but reeling in and out
  - environment = fog, winds, fog, reflections, thunderstorms, rain
- Power-ups system = in this one the player uses of three different buttons
  - invisible cloaking (invisible of the visual from the drone)
  - boosting up energy (extra speed)
  - EMP (trigger that goes out of the car to emp of the drone where it been destroy)

## Reference

Physiopedia (2016). *Borton's Development Framework*. [online] Physiopedia. Available at: [https://www.physiopedia.com/Borton%27s\\_Development\\_Framework](https://www.physiopedia.com/Borton%27s_Development_Framework).

Rigor Mortis Tortoise, [www.youtube.com](https://www.youtube.com/watch?v=Ldyw5IFkEUQ). (n.d.). *Procedural Generation: Endless Runner Unity Tutorial (Updated 2023)*. [online] Available at:  
<https://www.youtube.com/watch?v=Ldyw5IFkEUQ> [Accessed 23 Apr. 2024]