**Interactive Multimedia – Challenge 4 – Soccer Scripting – Document**

A screenshot of a computer

Description automatically generated

A screenshot of me with the Challenge 4 Game, which is where you control a checkered-designed ball, fighting soccer balls while using power-ups. Unfortunately, upon turning the game on, there is a huge number of problems in the game, which includes things such as: the enemy soccer balls are stationary and aren’t going after the player, the power-up gems are killing the player, multiple soccer balls are being added into the game for no reason, and whenever the player hits an enemy soccer ball, it the enemy flies right in front of the player instead of away. So, I would need to get started on fixing every single in this game and making it playable.

A screenshot of a computer program

Description automatically generated

A screenshot of the first problem getting fixed, which was the enemy soccer balls going onto the player ball instead of going back when they go into contact with the player, in which this problem was fixed by going into PlayerControllerX.cs and editing a line in private void OnCollisionEnter which was:  
Vector3 awayFromPlayer = other.gameObject.transform.position – transform-position;

A screenshot of a computer

Description automatically generated

A screenshot of the second problem getting fixed, which was whenever the player grabbed the gem power-up, a new wave of enemies appear, which doesn’t make a lick of sense, in which this was fixed by going into SpawnManagerX.cs and going into void Update() and editing it with:  
enemyCount = GameObject.FindGameObjectsWithTag(“Enemy”).Length;

A screenshot of a computer program

Description automatically generated

A screenshot of the third problem being fixed, which was the power being on the player ball all the time, which makes the game way too easy and therefore, the enemies do not stand a chance fighting back, so this problem was fixed by going into PlayerControllerX.cs and adding in a line of code into private void OnTriggerEnter(Collider other), being:  
StartCoroutine(PowerupCooldown());

A screenshot of a computer program

Description automatically generated

A screenshot of the fourth problem being fixed, which was where whenever a new wave of enemies came, it would always be the same number, being two, and this leads to not much challenge for the player if it’s the same amount every time, so this was fixed by going into SpawnManager.cs and editing:  
for (int i = 0; i < enemiesToSpawn; i++)

A screenshot of a computer program

Description automatically generated

A screenshot of the fifth problem being fixed, which was the most notable one where the enemy balls were remaining stationary and they weren’t offering much of a challenge for anyone going to play, so in order for this be fixed, was by going into EnemyX.cs and by going into void Start() and adding in:  
playerGoal = GameObject.Find(“Player Goal”);

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated

Multiple screenshots of the first bonus problem being solved, which was giving the player ball a turbo boost, in which this was solved by going into PlayerControllerX.cs and creating a Turbo Smoke object and then going back into the Unity file and adding a smoke particle effect into the Turbo Smoke icon of the player’s Player Controller script.

A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

Me fixing the second bonus problem, which was the lack of difficulty in the enemies for each passing wave, so in order to fix this, I went into SpawnManagerX.cs and EnemyX.cs and added in some code in order to fix this and make the game a little tougher as the player goes through each round and another enemy is added in.