Challenge 4

Soccer Scripting

Challenge Overview: Use the skills you learned in the Sumo Battle prototype in a completely different context: the soccer field. Just like in the prototype, you will control a ball by rotating the camera around it and applying a forward force, but instead of knocking them off the edge, your goal is to knock them into the opposing net while they try to get into your net. Just like in the Sumo Battle, after every round a new wave will spawn with more enemy balls, putting your defense to the test. However, almost nothing in this project is functioning! It's your job to get it working correctly.

Challenge Outcome:

- Enemies move towards your net, but you can hit them to deflect them away
- Powerups apply a temporary strength boost, then disappear after 5 seconds
- When there are no more enemy balls, a new wave spawns with 1 more enemy

Challenge Objectives: In this challenge, you will reinforce the following skills/concepts:

- Defining Vectors by subtracting one location in 3D space from another
- Track the number of objects of a certain type in a scene to trigger certain events
- Using Coroutines to perform actions based on a timed interval
- Using for-loops and dynamic variables to run code a particular number of times
- Resolving errors related to null references of unassigned variables

Challenge Instructions:

- Open your **Prototype 4** project
- **Download** the "Challenge 4 Starter Files" from the Tutorial Materials section, then double-click on it to **Import**
- In the *Project Window > Assets > Challenge 4 > **Instructions*** folder, use the resources as a guide to complete this challenge

	Challenge	Task	Hint
1	Hitting an enemy sends it back towards you	When you hit an enemy, it should send it *away* from the player	In PlayerControllerX.cs, to get a Vector *away* from the player, you should subtract the [enemy position] minus the [player's position] - not the reverse
2	A new wave spawns when the player gets a powerup	A new wave should spawn when all enemy balls have been removed	In SpawnManagerX.cs, check that the enemyCount variable is being set correctly
3	The powerup never goes away	The powerup should only last for a certain duration, then disappear	In PlayerControllerX.cs, the PowerupCoolDown Coroutine code looks good, but this coroutine is never actually called with the StartCoroutine() method
4	2 enemies are spawned in every wave	One enemy should be spawned in wave 1, two in wave 2, three in wave 3, etc	In SpawnManagerX.cs, the for-loop that spawns enemy should make use of the enemiesToSpawn parameter
5	The enemy balls are not moving anywhere	The enemy balls should go towards the "Player Goal" object	There is an error in EnemyX.cs: "NullReferenceException: Object reference not set to an instance of an object". It looks like the playerGoal object is never assigned.

	Bonus Challenge	Task	Hint
X	The player needs a turbo boost	The player should get a speed boost whenever the player presses spacebar - and a particle effect should appear when they use it	In PlayerController, add a simple if-statement that adds an "impulse" force if spacebar is pressed. To add a particle effect, first attach it as a child object of the Focal Point.
Y	The enemies never get more difficult	The enemies' speed should increase in speed by a small amount with every new wave	You'll need to track and increase the enemy speed in SpawnManagerX.cs. Then in EnemyX.cs, reference that speed variable and set it in Start().

Challenge Solution

1 In PlayerControllerX.cs, in OnCollisionEnter(), the ***awayFromPlayer*** Vector3 is in the opposite direction it should be.

```
Vector3 awayFromPlayer = transform.position -
other.gameObject.transform.position;
                                  = other.gameObject.transform.position -
transform.position;
```

2 In SpawnManagerX.cs, the ***enemyCount*** variable is counting the number of objects with a “Powerup” tag - it should be counting the number of objects with an “Enemy” tag

```
void Update() {
  enemyCount = GameObject.FindGameObjectsWithTag("~~Powerup~~ Enemy").Length;
  ...
}
```

3 In PlayerControllerX.cs, in the OnTriggerEnter() method, you need to initiate the ***PowerupCooldown*** Coroutine in order to begin the countdown process

```
private void OnTriggerEnter(Collider other) {
  if (other.gameObject.CompareTag("Powerup"))  {
    ...
    StartCoroutine(PowerupCooldown());
  }
}
```

4 In SpawnManagerX.cs, the for-loop that spawns enemy should make use of the ***enemiesToSpawn*** parameter

```
for (int i = 0; i < ~~2~~ enemiesToSpawn; i++) {
  Instantiate(enemyPrefab, GenerateSpawnPosition(), ...
}
```

5 In EnemyX.cs, the ***playerGoal*** variable is not initialized - initialize it in the Start() method

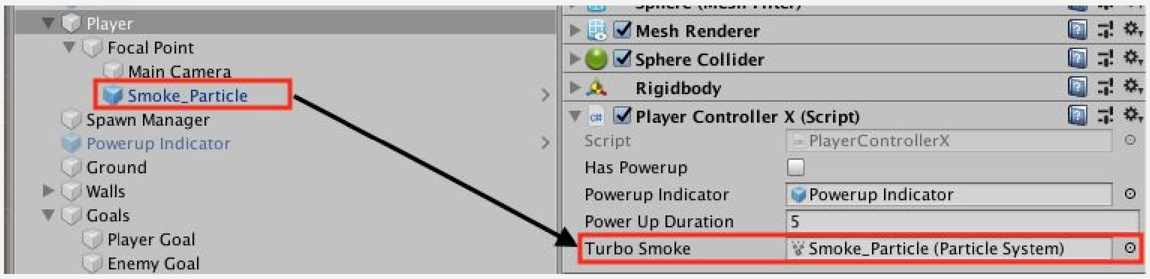
```
void Start() {
  enemyRb = GetComponent<Rigidbody>();
  playerGoal = GameObject.Find("Player Goal");
}
```

Bonus Challenge Solution

X1 To add a turbo boost, In PlayerControllerX.cs, declare a new ***turboBoost*** float variable, then in Update(), add a simple if-statement that adds an "impulse" force in the direction of the focal point if spacebar is pressed:

```
private float turboBoost = 10;

void Update() {
  ...
  if (Input.GetKeyDown(KeyCode.Space)) {
   playerRb.AddForce(focalPoint.transform.forward * turboBoost, ForceMode.Impulse);
  }
}
```

X2 Add the Smoke_Particle prefab as a child object of the focal point (next to the camera), then in PlayerControllerX.cs, declare a new turboSmoke particle variable and assign it in the inspector



Script	PlayerControllerX
Has Powerup	☐
Powerup Indicator	Powerup Indicator
Power Up Duration	5
Turbo Smoke	Smoke_Particle (Particle System)

X3 In PlayerControllerX.cs, in the if-statement checking if the player presses spacebar, play the particle

```
if (Input.GetKeyDown(KeyCode.Space)) {
  playerRb.AddForce(focalPoint.transform.forward * turboBoost, ForceMode.Impulse);
  turboSmoke.Play();
}
```

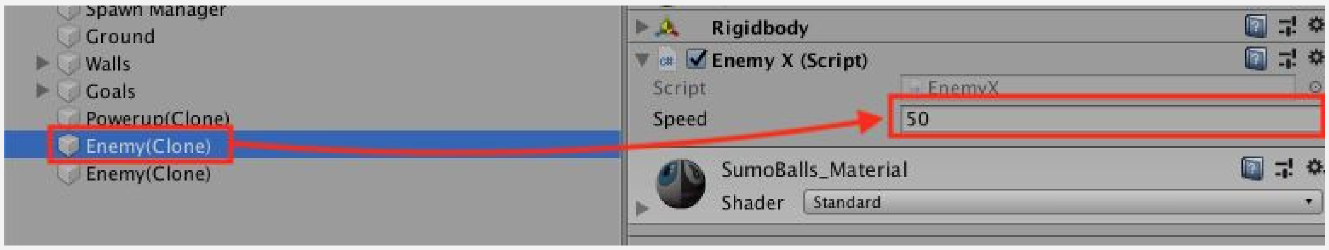
Y1 In SpawnManagerX.cs, declare and initialize a new public ***enemySpeed*** variable, then increase it by a certain amount every time a wave is spawned:

```
public int enemyCount;
public float enemySpeed = 50;

void SpawnEnemyWave(int enemiesToSpawn) {
  ...
  waveCount++;
  enemyCount += 25;
}
```

Y2 In EnemyX.cs, declare a new spawnManagerXScript variable, get a reference to it in Start(), then set the enemy's ***speed*** variable to your new ***enemySpeed*** variable

```
private GameObject playerGoal;
private SpawnManagerX spawnManagerXScript;

void Start() {
  enemyRb = GetComponent<Rigidbody>();
  playerGoal = GameObject.Find("Player Goal");
  spawnManagerXScript = GameObject.Find("Spawn Manager").GetComponent<SpawnManagerX>();
  speed = spawnManagerXScript.enemySpeed;
}
```

Y3 To test, make the ***speed*** variable in EnemyX.cs public and check the enemies' speed when they are spawned in different waves



Speed 50