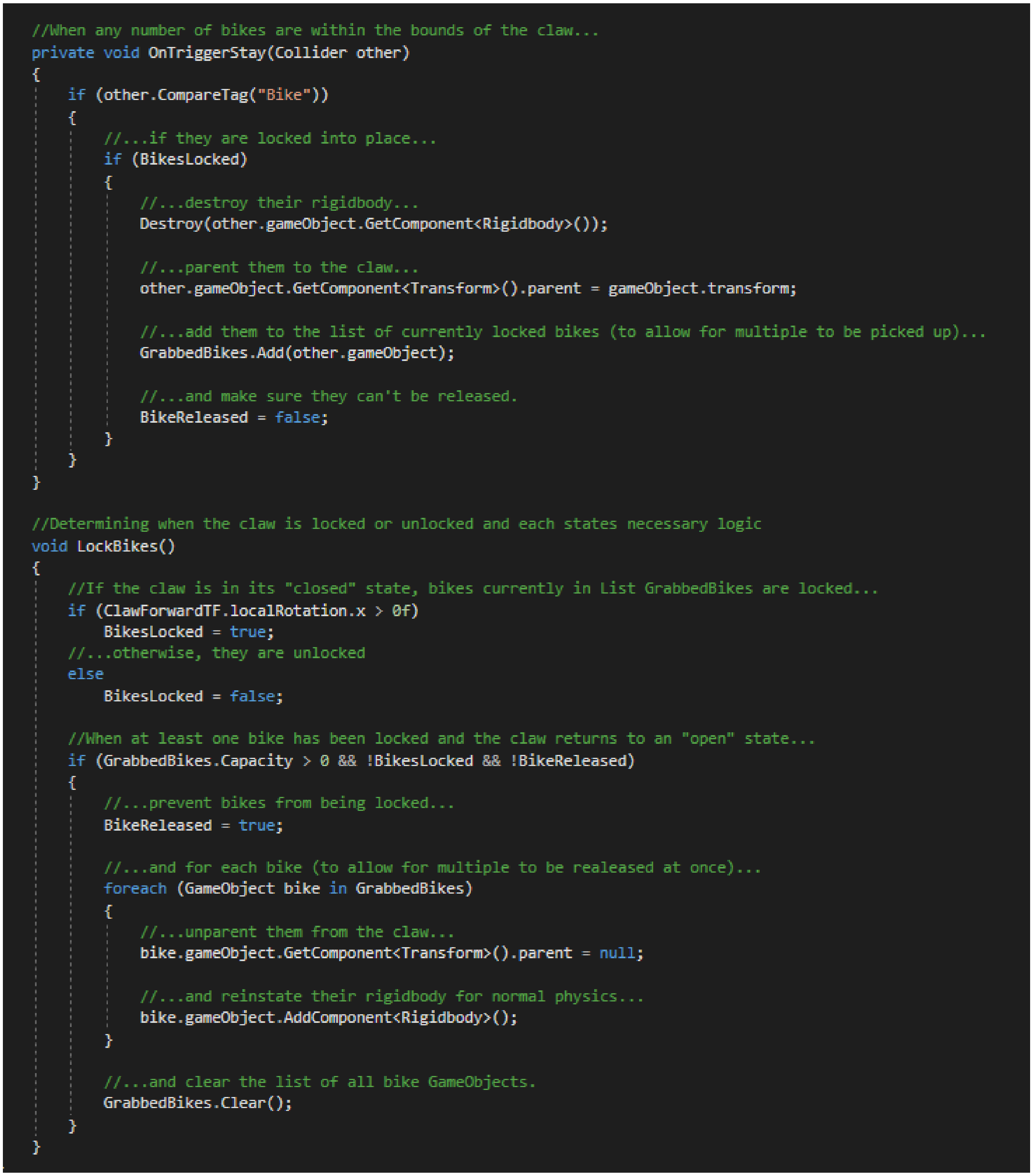
CODE SAMPLE

Below is a sample of code from *The Amsterdam Bicycle Fishery.* The core mechanic in this game is a giant claw that fishes bikes out of the water, not too different from an arcade claw machine. I talk about this in more detail in the Prototyping and Iteration video. There I discuss how the core functionality of the Claw ran into major issue throughout early prototyping. However, with these two functions below, I resolved all problems associated with the Claw interactions and more.



```
//When any number of bikes are within the bounds of the claw...
private void OnTriggerStay(Collider other)
{
    if (other.CompareTag("Bike"))
    {
        //...if they are locked into place...
        if (BikesLocked)
        {
            //...destroy their rigidbody...
            Destroy(other.gameObject.GetComponent<Rigidbody>());

            //...parent them to the claw...
            other.gameObject.GetComponent<Transform>().parent = gameObject.transform;

            //...add them to the list of currently locked bikes (to allow for multiple to be picked up)...
            GrabbedBikes.Add(other.gameObject);

            //...and make sure they can't be released.
            BikeReleased = false;
        }
    }
}

//Determining when the claw is locked or unlocked and each states necessary logic
void LockBikes()
{
    //If the claw is in its "closed" state, bikes currently in List GrabbedBikes are locked...
    if (ClawForwardTF.localRotation.x > 0f)
        BikesLocked = true;
    //...otherwise, they are unlocked
    else
        BikesLocked = false;

    //When at least one bike has been locked and the claw returns to an "open" state...
    if (GrabbedBikes.Capacity > 0 && !BikesLocked && !BikeReleased)
    {
        //...prevent bikes from being locked...
        BikeReleased = true;

        //...and for each bike (to allow for multiple to be realeased at once)...
        foreach (GameObject bike in GrabbedBikes)
        {
            //...unparent them from the claw...
            bike.gameObject.GetComponent<Transform>().parent = null;

            //...and reinstate their rigidbody for normal physics...
            bike.gameObject.AddComponent<Rigidbody>();
        }

        //...and clear the list of all bike GameObjects.
        GrabbedBikes.Clear();
    }
}
```