Text

Description automatically generated

Games and Multimedia

Practical Assignment 1

Authors:

David Santos – 2202226

Inês Pombinho – 2202242

Professors:

Ricardo Antunes

Roberto Ribeiro

Luís Monteiro

Course:

Introduction to Programming

# Implemented Algorithms

## Enemy Detection (for all systems)

This feature implemented using colliders and Enter Trigger / On Trigger Stay event. Biters and Enemies are tagged differently and with different physics layers as well, in order to not trigger events unnecessarily.

## Alert System

The alert system was implemented using a reference to the main script and when the collision is triggered the parent script is warned and will activate the panic routine. If the enemy is already alert nothing will happen.

To implement this feature…

## Enemy Base Behaviour

All enemies including biters have this component. It is the base for movement and for wall detection. This component holds references to the AlertSystem, BiterBehaviour and EnemySocial.

## Enemy Social

This component is only active when the enemy state is set to normal. Whenever an enemy meets another non-biter enemy with the OntriggerEnter2d, it will check for

## Biter Behaviour

This component adds all aggressive behaviour to the enemy. When active the component will react to “OnTriggerEnter2d” events and will call the bite function.

## Biter Selection

The biters are selected at random during the start of the game, a timer is also set repeating the same selection every 30 seconds or so. This value is editable on the editor.

## Health and Armour System

The health and armour system are separated components. If the enemy contains an amour component the health system will pass the health calculations through it first.

# Reasoning behind the options taken

## Enemy Detection (for all systems)

The events “OnTriggerEnter2D” and “OnTriggerStay2D” were used as means to detect when to attack and when to be alert, this because we felt that it was the most efficient way to do it. An alternative would be to have a coroutine and check on both left and right sides at a given time but, in the end, we ended up using these events for simplicity sake. The fact they were in different layers made the need to have the “OnCollisionEnter2D” event and to have a check “if collision.compareTag(“Enemy”)” in order to ignore collisions but still allow the OnTriggerEnter2D to work.

## Alert System

The alert system is a different component that is activated when the enemy is on a Normal state. It holds a reference to the mean enemy base script and when the alert state is triggered it will call a function named “SetAlert”, a timer is set to call the function “CalmParent” returning the enemy to its normal behaviour.

The reason as to why the enemy does not run the alert routine if he’s already alert is to prevent situations in which he would just be flipping infinitely due to having two biters on each side or having a biter and a wall on another side.

## Enemy Base Behaviour

The base component for enemies involves the base checks for ground, walls and movement. This component also has the references to all the other components, having the needed functions to change the enemy mode when necessary.

## Enemy Social

The enemy social behaviour was implemented using the same trigger idea as previous components. When an enemy enters in contact with another it will check if each are already idle or not, if they are already Idle the enemy will ignore them and pass through (in order to only have 2 idle enemies in a group).

## Biter Behaviour

When activated all the other “normal behaviour” components will be disabled, the enemy will have a different colour and it will start biting other enemies. When an enemy reaches close to the biter, the attack animation will play and at the end of it the enemy will receive the specified damage.

## Biter Selection

This selection is done with the Game Manager. On the Awake function we search for all objects with tag “Enemy” since all enemies at the start are normal. We add these enemies to a list and then a random element will be selected. That element will be selected as the biter. This function of selection is tied in a repeating timer with the specified time on the editor. When a new biter is selected first all of the biters are cleared.

When an enemy dies their reference is removed from the enemies list, in order to prevent issues regarding choosing a non-existing enemy as a biter.

## Health and Armour System

When a Biter does the attack, they only get the reference for the component health system. This because the enemy does not need to get the Armour component as it may not exist but, we assume that all enemies have a health component at least. On the function “TakeDamage” of the Health system we check if we have the reference to the armour system, this reference is obtained at the Start method and if the result is null the damage will be calculated directly with the health of the enemy. If the reference is not null the damage will be passed through the Armour component first and then if the resulting calculation is bigger than 0 the health system will also calculate the rest of the damage and mark it as a bite.

# Implemented Features

All the features required in the practical assignment corresponding to this evaluation were fulfilled completely.

# Bibliography

During the development of this work, the following references were consulted:

* http://www.brechtos.com/tagselectorattribute/ (Using tags as a dropdown property in Unity’s inspector using PropertyDrawers)