**Project Description**

The objective of the prototype is to test the functionality of the breathing system. The application takes a “float” type value and trigger events according to the number between -1 and 1. The -1 value represents the Breath In moment and the 1 is the Breath Out. The events were developed in different iterations.

**Iteration 1:**

The Camera moves around a circuit when the user presses the space bar. The breath value doesn’t affect the movement.

**Iteration 2:**

The player moves around the circuit. The movement speed responses to the breath value automatically. The user can’t control it.

The player has burble particles to give feedback with respect to the breath moment.

**Iteration 3:**

The user can control the breath moment using the keyboard. Pressing “Q” the player breathes in and pressing “E” the player breathes out.

\* Particle system optimized.

\* Sound effects implemented.

\* Player size responses to the breath value.

**Iteration 4:**

[Breath Library Implementation: Planned]

**Technical Description**

Each iteration is saved in a Unity Scene. The project has the following files in the “Assets” folder:

**Animations:** Animator – Controls the blue borders in the screen. The border appears when the breath value is keeping a lot of time. Other files in the folders are the different Animation Clips.

**Materials:** Materials and Sprite Sheets – “Player” material is used in the player representation (The Cube), “Plane” material is used in the environment.

Burbles material are the burbles particle animation.

**Sprites:** 2D Textures – Blue border and burbles.

**Sounds:** Sound Files – Only some sound files are used in the project.

**Scripting Summary**

**BreathInput**

The main script in the project. Its function are take the breath value and send it to the another scripts. The final objective is to use this script to receive the breath value of the breathing library.

**BreathSim:**

Temporal script is used to simulate a breath cycle, sending values between -1 and 1 to the BreathInput.

**BreathWarning:**

Controls the blue borders animation. The border appears when one breath moment is so large.

**BurblesGenerator:**

It changes the particle values according to the BreathInput values. This script is executed always, when the breath value changes the particle response it.

**ManualBreathing:**

Temporal script is used to send breath values using the keyboard Q and E.

**ReSize:**

Modify the player size according to the breath value.

**SoundController:**

Use the keyboard to trigger sound effects. This script will be rewritten to use the breath value to replace the keyboard input.

**Waypoints & WaypointsMover:**

Control the movement of the player according to the breath value.

**UML Diagram**

Diagram

Description automatically generated