VIA UNIVERSITY COLLEGE
ICT ENGINEERING

Anatomy

Project Report

SGM1-S17

Bianca Sgondea (240312)

Supervisors

Michael Viuff

Kasper Knop Rasmussen

Concept

What?

“Anatomy” is designed with an educational purpose, having as target audience children between 8 to 12 years old. The game is programed to run on an interactive table using touchscreen and real objects with tags that the software can recognize. It can shortly be described as a puzzle game covering some parts of the anatomy field. Having in mind the target audience the graphic design of the game is done in a low poly manner which is more appropriate for children.

The idea behind “Anatomy” is that it is a very simple 2D game which allows children to interact with 3D printed organs with tags that has a correspondent organ in the game. By placing the 3D organs on the interactive table will generate the 2D organ in the game and if moving the 3D model the correspondent organ from the game will follow it. The interaction with the organ from the game will stop when it is locked to its correct position. The win condition in this game will be fulfilled when all organs are locked on the correct spot on the outline of the body. Another feature of the game is that it allows children to read some fun facts about a specific organ that they chose from a list which will maybe lead to the wish to learn more about anatomy. By playing this game children will start recognizing the shape the of the most important organs from the human body. They will also gain knowledge about the location of each organ and also some information about the functionality of it.

The fact that children can learn real life aspects just by playing the “Anatomy” game is the reason why it can be called a serious game. It educates children in the way that they are going to have a better understanding about their bodies. It is important for children to know what is happening inside their bodies so that they can prevent things from happening (maybe they will eat less sugar…). Learning by playing can increase children interest for that field.

Why so simple?

According to the scenario the interactive table on which the game will run will be placed in a science museum. That made us think about something easy to play with a high impact on learning. Which can be played by more players.

Implementation

The implementation of the “Anatomy” game is simple as the idea behind it. It consists of four main scrips: “Spawner”, “Organ Controller”, “Lock Organ” and the “Game Manager” and also other three scripts used more for the design and animation. The entire functionality of it is that the user can move around organs, place them on the correct position and navigate through an info panel. During the entire game the user has access to a restore button and mute or unmute. The access to the info panel is also made by a button which during the game will increase in size and at a certain point decrease. This happens because of the “Button Script” attached to it.

The game was designed to be used on an interactive table which gives the possibility of having 3D objects recognised by the software from the fiducial tags. Each object has o correspondent in the game. For example, an object with the tag “brain” is assigned to the object with the fiducial ID one. The fiducial tag once placed on the table will display that specific organ having same position and rotation as the tagged object placed on the table has. Based on the position of the tagged object the displayed organ also changes its position, for example if moving the fiducial, the correspondent organ will follow it. All this functionality is done by having two classed Transform Gesture and Transformer, attached to the game objects that needs to be dragged on the screen. Those two scripts where provided by Unity Asset Store.
During implementation, the only way of testing the functionality of the game was using a TUIO simulator.

Spawner

When the game starts all organ prefabs will be spawned at a specific location in the game outside of the view area. This is handled by the “Spawner” script which is attached to an empty game object named “Spawner”. This scrip contains three methods: `spawn()` consisting in a loop that takes each organ and set it as a child for “Spawner” game object and also sets the position to `Vector3.zero`. This is the only method from this script called in `Start()`. The other two methods `hideOrgans()` and `showOrgans()` are called in “Game manager” script and the only functionality that they have is that they disable and enable the sprite renderer. Each prefab that is spawned has the following four scripts attached. “Organ Controller”, “Lock Organ”, “Transform Gesture” and “Transformer”.

Organ Controller

Once a fiducial is placed on the interactive table, this script takes care of the display of the organ. When the fiducial touches the table the `Instance_TouchesBegan(object sender, TouchEventArgs e)` is called. It transforms the initial position where the organ was spawned to the position where the fiducial is placed. This is happening only if the organ is not locked on its position. This script also is taking care of what is happening when the fiducial is taken off the table by calling `Instance_TouchesEnded(object sender, TouchEventArgs e)` method. It checks if the organ is locked, if not it will deactivate the sprite renderer. All the methods from this script are registered as events in the `OnEnable()` method.

Lock Organ

During the game the organs need to be fixed on their correct position so that they can be counted for the win condition. This script checks if a specific organ is in a given range by always checking the position of the organ.
From my opinion, the most interesting aspect from this game is the actual locking of the organs. In this script are assigned some constrains that need to be full field to lock the organ. There are 6 private variables which are representing the range in which the organ needs to be for it to be locked, there is also a public reference to the organ outline from the game to which we assign this borders in `Start()` .A public boolean locked was needed - indicates if the organ is locked on it's correct position. It needs to be public because we also need to use it in "Organ Controller". The `check()` method has the functionality of checking if the organ is in the given range and if so, it places the organ on the exact position, sets the locked boolean to true which tells the "Organ Controller" that an organ is locked and disables the access of moving it. It also calls `incWinCount()` from "Game Controller" which increase a counter for the winning condition. The `check()` is called in the `Update()` if the the locked Boolean of a specific organ returns false.

Game Manager

The "Game manager" script holds the entire functionality that is not directly related to other concrete game object. This script is attached to an empty game object called "Game Manager" It consists of some main methods such as `resetGame()`, `toggleVolume()`, `openInfoPanel()`, `winCondition()` and other helper methods. Most of the methods are used for handling the buttons from the user interface. `winCondition()` method is called when all organs are locked.

The most interesting method from the “Game Manager” script is `openInfoPanel()` – it moves the main camera to the second point in the scene and enables another view containing two panels. In one of the panels are listed all the organs from the game as buttons. By pressing one of the buttons form the list will make the text change in the second panel. The panel that is holding all buttons has attached to it a script called “Info switcher” which contains methods that will display different information. So that when pressing a button, the text from the second panel will change according to the method that was called.