

# Parts of the Heart - Design Document

## Game Overview

### Project Background

Medica Interactive is a medical education company that creates interactive experiences for young students to learn more about biology and medical science.

The company wants to create various AR learning experiences that will be included in educational materials for elementary students.

### High level concept

**App name:** Interactive Heart

**Platform:** Web

**Target audience:** Children aged 8-12

**Target device:** Mid- to high-end Android and iOS mobile phones and tablets

**Release date:** Late 2023

**Publisher:** Medica Interactive

The Interactive Heart is a simple app where players can tap to explore the main parts of a human heart to unlock information about its function. Once the player has explored this virtual heart, they will have the knowledge they need to make a short presentation.

### Game objective

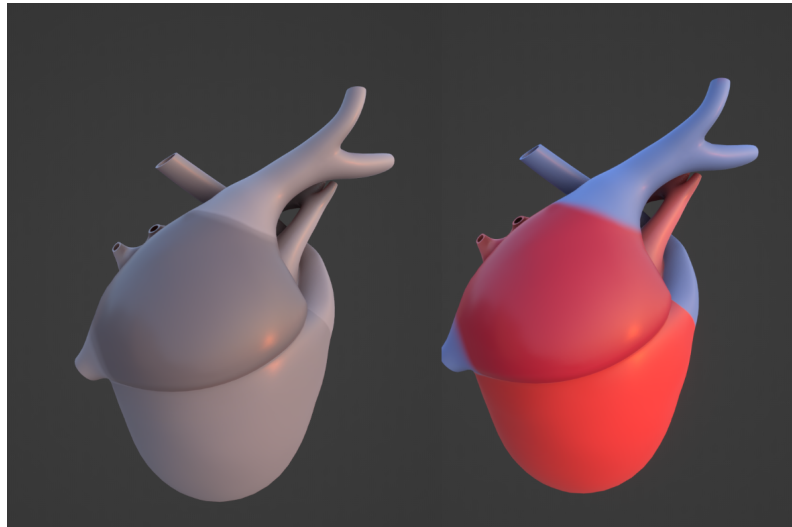
Use augmented reality to view a human heart and scan its major parts to learn more about how they function. This knowledge can be used to create a short presentation.

## Game rules

- Using the School ebook, scan the Medica Interactive QR code and open the experience.
- Tap the heart to bring it to life.
- Tap each part of the heart to unlock more information.
- If the player is not actively exploring the heart, it will eventually stop beating and fall to the ground.

## Heart gameplay

The heart will initially be gray and lifeless. Once the user taps the heart, it will start beating and assume a healthy color.



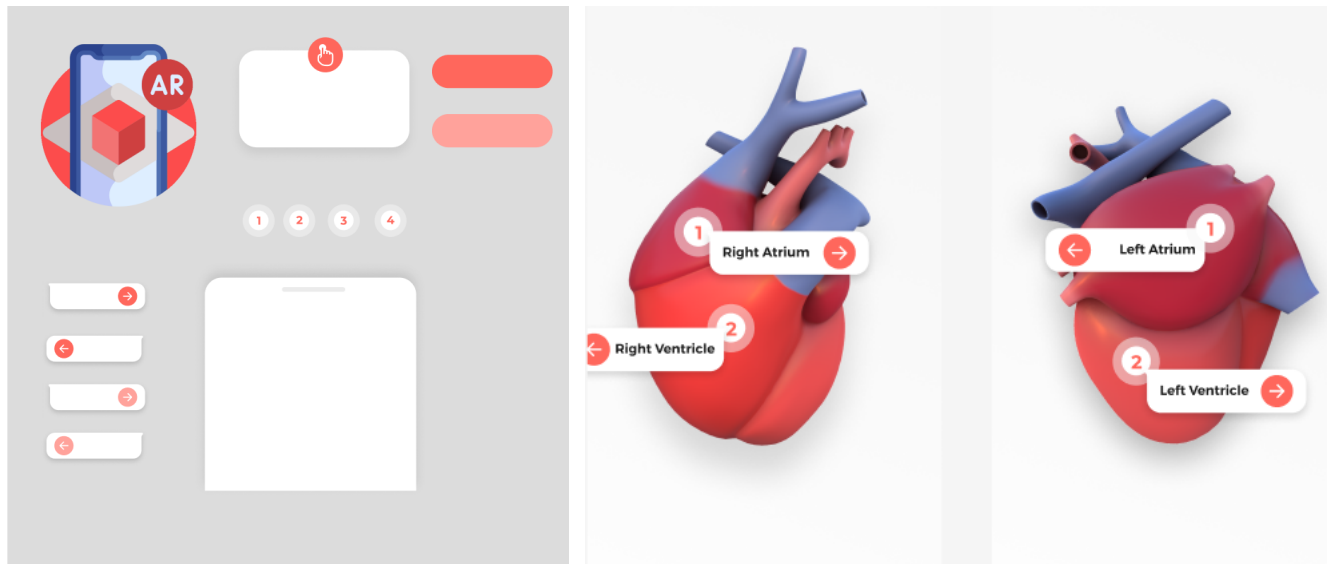
The heart will be divided into its four principal parts. Everytime the user taps one of these parts, informational will appear on screen. The four heart sections are:

- Right Atrium
- Right Ventricle
- Left Atrium
- Left Ventricle

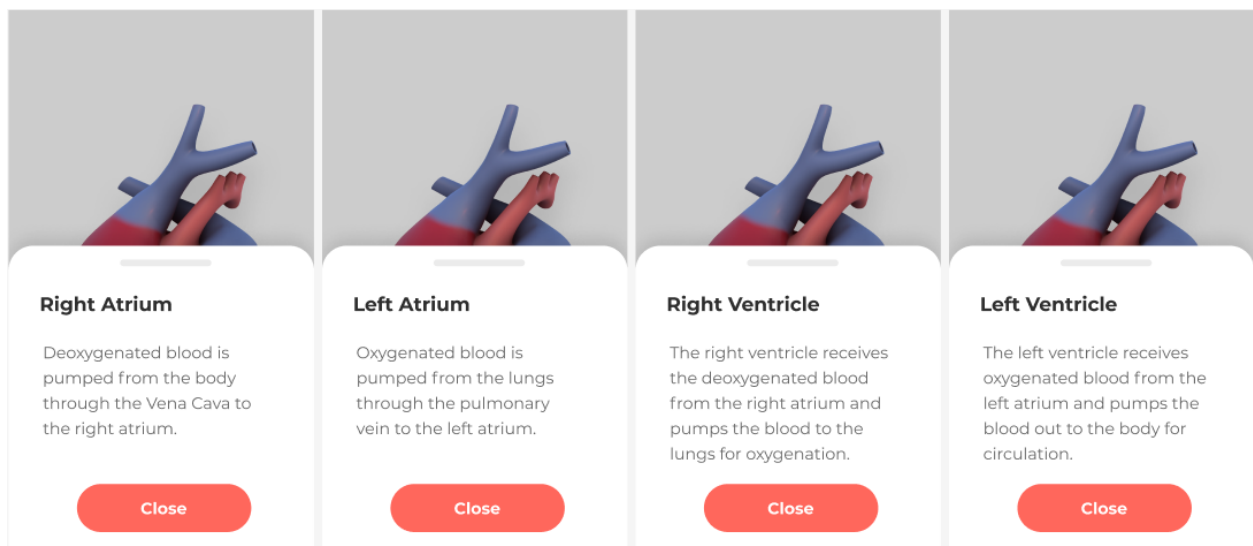
If the heart is left unattended and stops beating, it must be tapped to be brought back to life again.

## User Interface (UI)

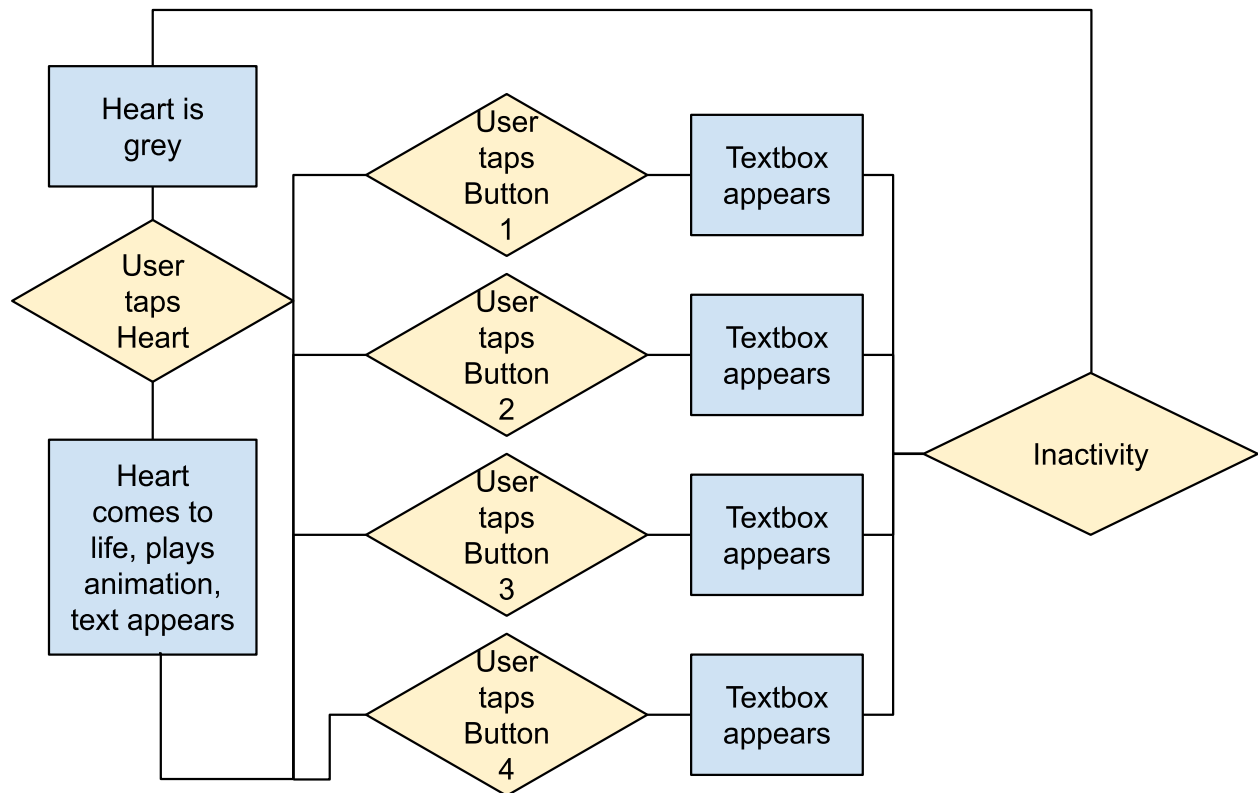
The UI will consist of four buttons, one pointing to each section of the Heart. These buttons will be provided as a whole texture atlas, that will be overlaid on top of the Heart's model in 3D space.



Each of these buttons will open an information panel when clicked and display the content as shown below.



## User flow



## Feedback

The following feedback will play when the user interacts with the heart.

Event	Action	Feedback
Heart is Gray	none	Text: Tap the heart
Tap Heart	Heart comes to life	Visual: Textures appear on the heart Visual: Heartbeat animation Visual: Heart starts floating Sound: Heartbeat sound effect
Section 1 - Right Atrium	Text appears	Visual: TextBox UI Text: The right atrium receives non-oxygenated blood from the body's largest veins.

Section 2 - Right Ventricle	Text appears	Visual: TextBox UI Text: The right ventricle pumps blood to the lungs where it receives oxygen.
Section 3 - Left Atrium	Text appears	Visual: TextBox UI Text: The left atrium receives oxygenated blood from the lungs and pumps it through the left ventricle.
Section 4 - Left Ventricle	Text appears	Visual: TextBox UI Text: The left ventricle pumps oxygen rich blood to the rest of the body.
Inactivity	Heart back to idle	Visual: Gray textures on the heart Visual: Heartbeat animation stops Visual: Heart falls to the ground Sound: Heartbeat sound effect stops

## Asset list

### Visual assets

- Font:
  - Montserrat-Bold.ttf
  - Montserrat-Regular.ttf
- Heart:
  - Heart.fbx
  - Heart\_life\_diffuse.png
  - Heart\_lifeless\_diffuse.png
  - Heart\_Occlusion.png
  - Heart\_Normalmap.png
- UI (texture atlas):
  - AR\_Heart\_UI\_Assets.png

### Audio assets

- Beat.mp3

### Javascript assets

- GroupOpacity.js
- UIController.js
- OpenInfoWindow.js
- CloseInfoWindow.js

# Technical details

## UI

2D screen for information window

3D screen for heart's buttons

## Recommended functionality

Create scripts for:

- Animation controller
- Group opacity
- Material swap
- Opening information window
- Physics handler
- Touch input
- Close information window
- Lifeless timer
- Play heart sound
- UI Controller

## Notes

1. Opacity instead of enable/disable on 3D UI
2. 2D UI could have disable/enable mode
3. Timer using delta time
4. State event (enable/disable)
5. Use the sprite editor on the UI asset to subdivide the assets (performance)

## Proposed hierarchy

1. Scenario
  - a. Camera
  - b. Lights
2. 2D screen
3. Pattern marker
  - a. Heart
  - b. Floor
  - c. 3D screen
4. Scripting

# Folder structure

Fonts

Models

UserInterface

Scripts

Sounds