|  |  |
| --- | --- |
|  |  |
|  |  |

**University of Arkansas – CSCE Department**

**Capstone II – Final Report – Spring 2021**

# **Immersive Video Game to Encourage Proper Breathing and Alleviate Anxiety in Young Patients during Preoperative Anesthesia Delivery**

**William Baker, Gavin Glenn, Jackson Gregory, Jared Harris, John Ostermueller, John Shelnutt**

## **Abstract**

Hospitals have expressed interest in an interactive breathing game for young children wearing a breathing mask that delivers anesthetics before surgery. The purpose of the game is to reduce anxiety in children, facilitate normal breathing while wearing the breathing mask, and aid doctors and pre-op nurses in the anesthesia process. The approach was to design and create an interactive game that detects breathing and encourages good breathing patterns through positive reinforcement in a relaxing and colorful game world. The game runs on the available hospital hardware and integrates seamlessly into the pre-operation process. This solution significantly benefits young patients in reducing their anxiety and feelings of discomfort around their environment and the breathing mask. It places a strong emphasis on the emotional wellbeing of children, allowing medical professionals to focus on their physical wellbeing going into surgery.

## **1.0** **Problem**

Young children about to go into surgery often feel anxious being away from their parents. They are around many strangers and in an unfamiliar setting. Additionally, the anesthetics they need before surgery are delivered using a breathing mask that can be intimidating or make the children uncomfortable, causing them to breathe in an irregular fashion. Hospitals would like a way to alleviate this anxiety in order to provide a better experience for young patients and a smoother anesthetic delivery process for the doctors and pre-op nurses. The issue is that this aspect of surgery for children has often been overlooked. There is a lack of solutions that offer children a calming yet interactive distraction to ease them into the act of breathing regularly in a breathing mask and allow doctors and pre-op nurses to focus on other aspects of preparation. Such a solution could be integrated with the existing pre-op workflow and could be utilized before any stressful operation or while getting imaging done. There has been a prototype of a breathing game developed by Stanford researchers and engineers on this topic, but the scope of the project is limited to the Stanford Children’s Health hospital. While many other hospitals would benefit greatly from a similar solution, there is a lack of widely available and easily integrated solutions available online.

## **2.0** **Objective**

The objective of this project is to create a game that children going into surgery will be able to play to simultaneously receive coaching on how to breathe in anesthetic and reduce operation anxiety. The game has a diver character swim through an underwater environment, collecting coins as they swim. The movement works as feedback to the patient by letting them know that they are breathing in the desired way. Movement is triggered by a nurse monitoring the patient’s breathing patterns using data gathered by onsite equipment used during the anesthetic process. This game will allow the nurse to be able to use the game to give feedback to the patient regarding their breathing and reduce preoperative anxiety that the patient may have during the anesthetic process.

## 

## 

## 

## 

## **3.0** **Background**

### **3.1** **Key Concepts**

The solution we aim to construct in this project is an ensemble of existing technologies. Everyone is familiar with video games; the purpose of this video game is not to level up or progress over time, but to remove anxiety through immersion into the game and encourage deep breathing with eye-catching visual feedback.

Due to the difficulty associated with directly gathering readings from the onsite equipment and transferring those readings to whatever device our game is currently running on for it to use, our group ultimately decided that this was not a feasible solution and instead opted for feedback triggers that are not tied to the anesthesia machine’s sensors. Utilizing a “man behind the curtain” style of feedback through the use of a clicker connected to the device the game is running on still allows us to achieve our goals of creating a game reduces preoperational anxiety while being able to be integrated into the preoperative process.

### **3.2** **Related Work**

The most notable existing implementation of a game to ease pre-op anxiety in children is “Sevo the Dragon” by Stanford Children’s Health’s Childhood Anxiety Reduction through Innovation and Technology (CHARIOT) team [1]. In CHARIOT’s game, the patient chooses a color for Sevo the Dragon, as well as his hat color. As the patient exhales through the anesthesia mask, Sevo breathes fire, cooking the patient’s choice of food. CHARIOT’s game is displayed on a projector screen that takes up a large portion of the patient’s field of view. Immersive environments and novel stimuli have been shown in previous research to enhance patient engagement with distractive entertainment [2].

In CHARIOT’s description of their game, we could not find mention of direct interfacing between the game and the anesthesia machine. In our introductory meeting, Dr. Allen and Ashlynn Young showed us that the patient’s breath mask has connected sensor equipment in the anesthesia machine. Using ports on the machine, we hoped to use this real-time, precise data in our game to give accurate visual feedback on inhalation and exhalation to both the patient and the caregivers.

As our work on this project commenced, we reached out to both Stanford’s CHARIOT team and the manufacturers of the anesthesia machines used at ACNW to explore the feasibility of using measurements directly from the equipment, without the need for installment of additional sensors or modified masks. Unfortunately, we determined that it was not feasible to interface directly with the onsite anesthesia machines to acquire real-time data for use in our project, forcing us to opt for a “man behind the curtain” style of movement triggers for our project.

## **4.0** **Design**

### **4.1** **Requirements and Design Goals**

Our application should at least abide by the following hard requirements:

● It does not encourage children to hyperventilate, either with too deep of breaths (slightly problematic) or too fast of breaths (majorly problematic)

● It does not encourage children to hypoventilate

● It does not gamify the experience in a competitive manner

● Data collected (if any) is anonymized

There are several other soft requirements that are less crucial to the core success of the project but are still quite important:

● The application should be intuitive and easy to use by children

● The application would ideally be hosted on a mobile device such that children can use it in a variety of scenarios (in a hospital bed, a wheelchair, sitting, standing, etc)

● The mobile device that hosts the application should be able to receive breathing data wirelessly

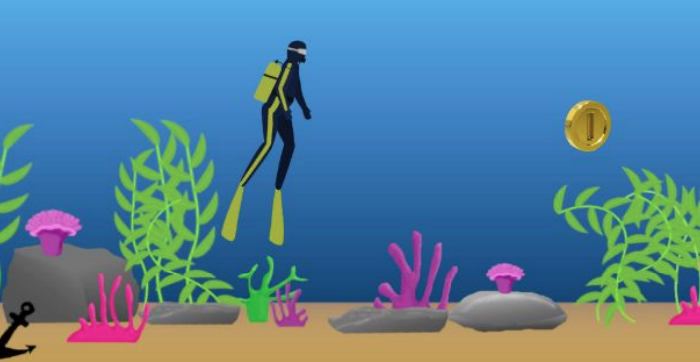
The application is intended for use with children ages 2-9 receiving preoperative anesthesia.

### **4.2** **Detailed Architecture**

**Architecture**

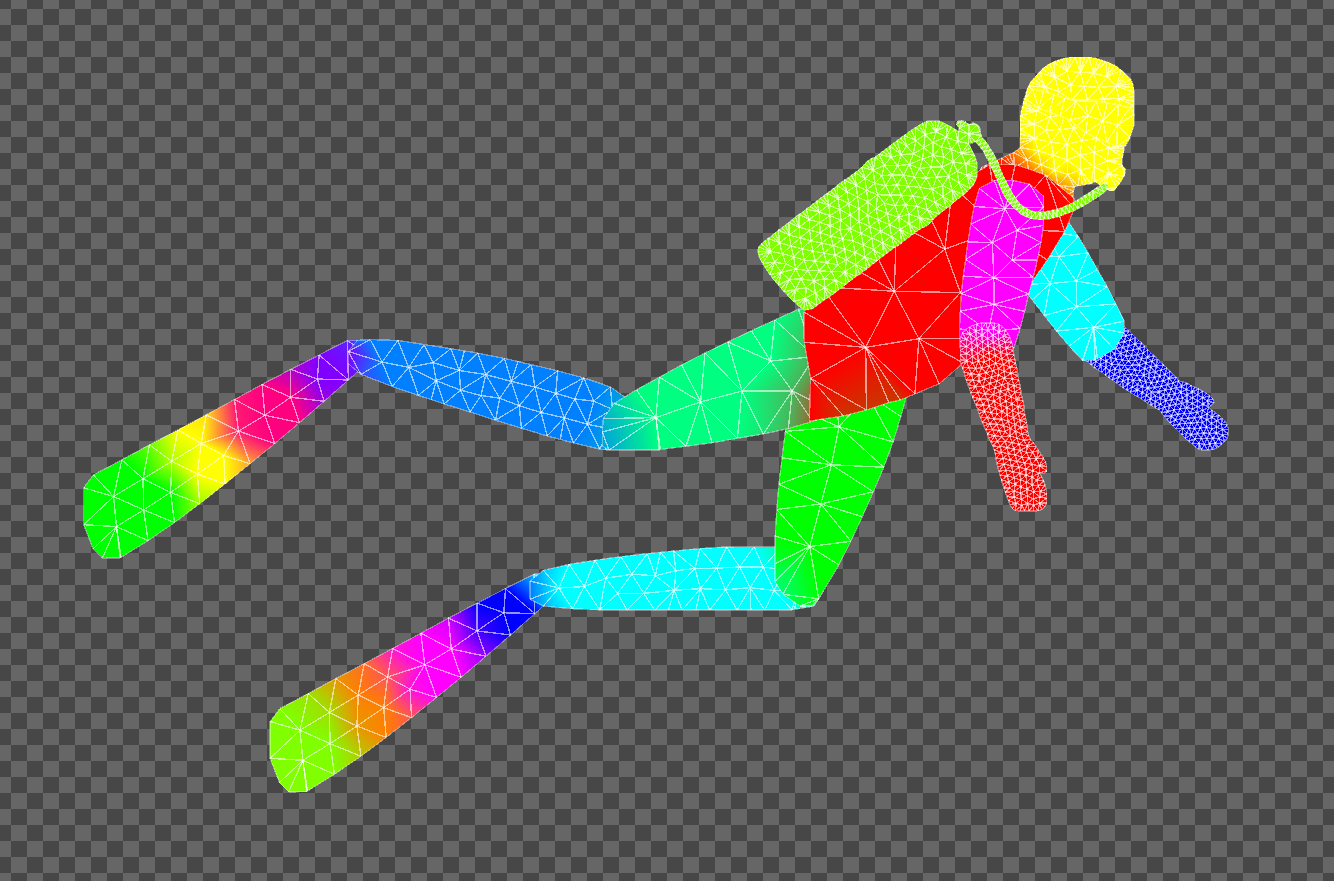
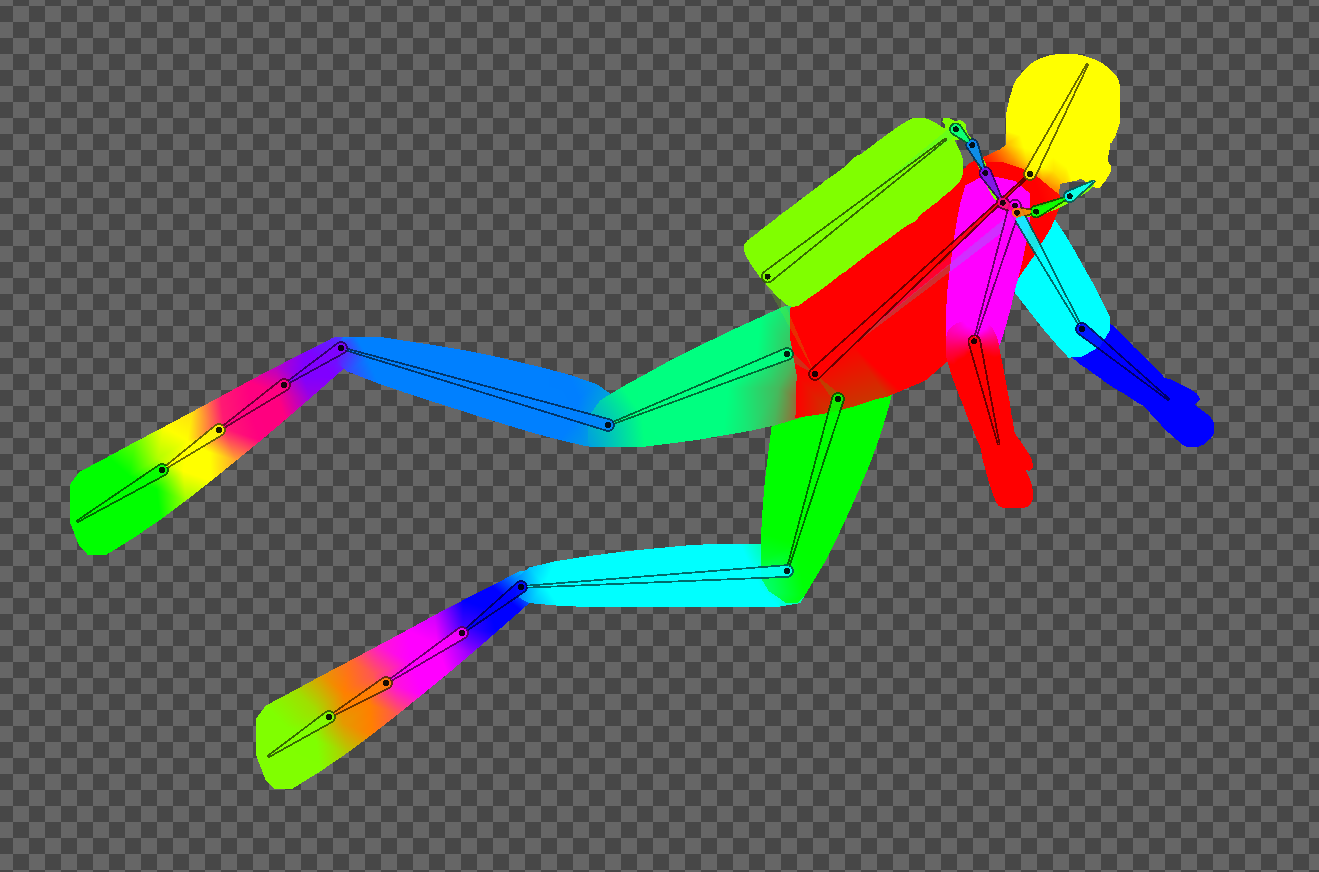
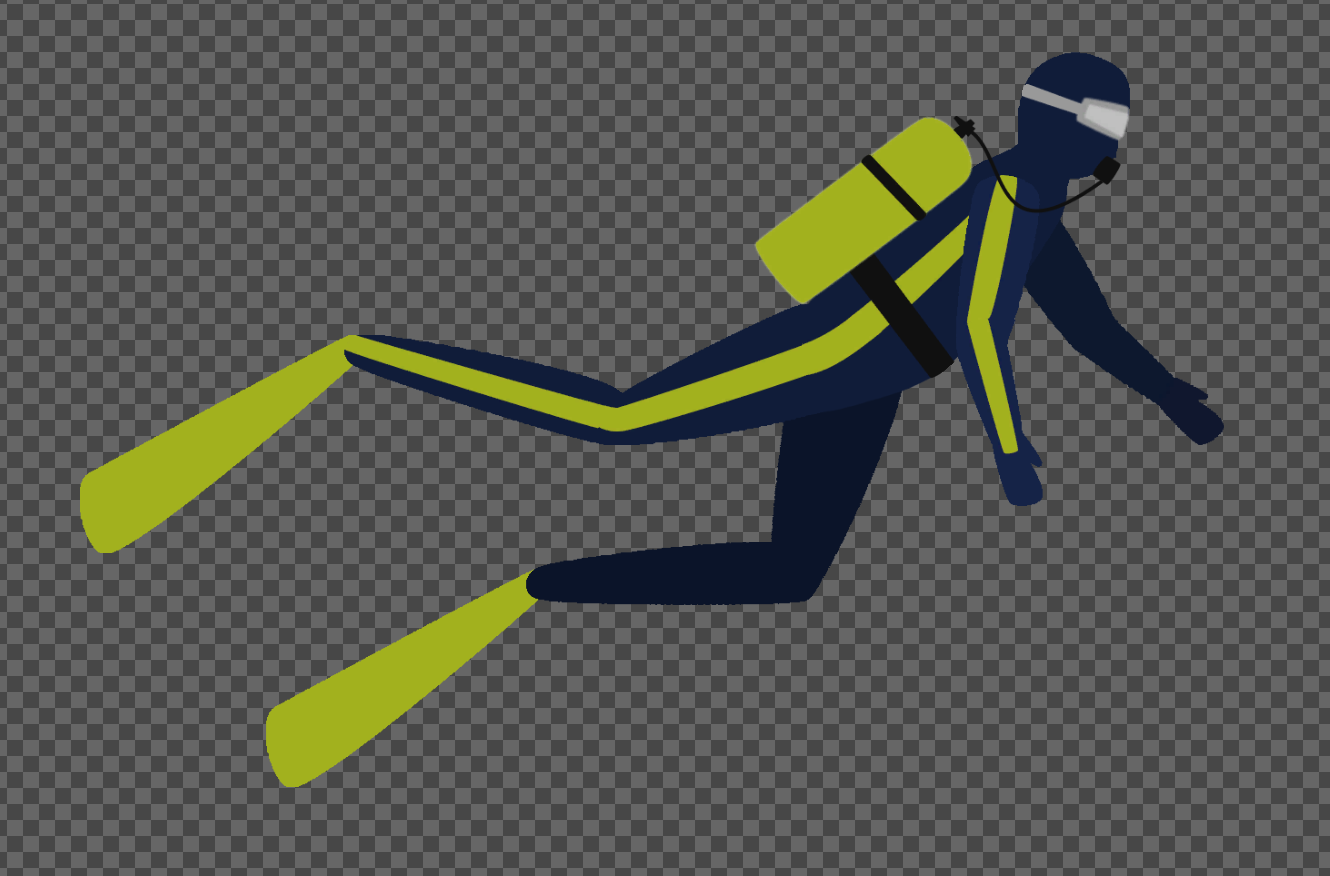
The game was designed using Unity as the primary game engine and the forefront of the application development process. Scripts for the camera movement and game object were written in C#.

The project was initially composed of a background image and a camera scrolling function that would transition across a replicated background to simulate movement in the positive x direction. The camera movement script was hooked up with our remote clicker, which we could use to trigger the camera to move along the scene. A diver object was added along with animations and sprites to simulate a diver beneath the water. Now we could move the player along with our camera for a seamless scrolling effect that indicated the diver’s movement across the screen.



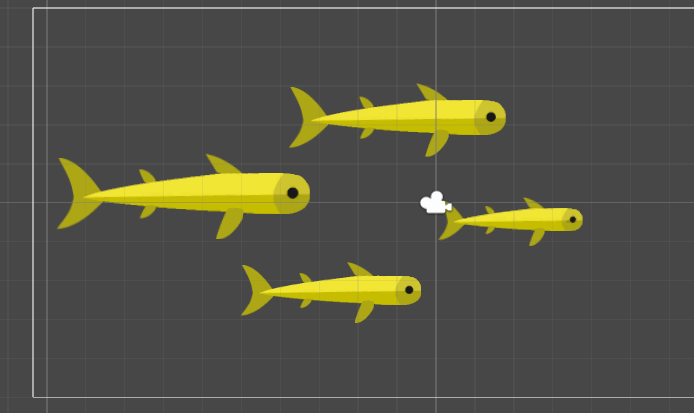
*The idle diver, with an asset-populated background and coin in sight.*

The diver’s idle, swimming, and transitioning animations were crafted for a realistic look when the player moves underwater. All animations for the game were designed in GIMP. After layers had been created in GIMP, they were exported into Unity, and then further edited with Unity’s built-in sprite editor.



*The diver model in three stages of skinning: the base sprite, added bones, and the bone-related geometry*

Several fish objects were also added to the game for another layer of detail in our background. The fish have their own custom swimming animations which move them across the screen. Several fish are grouped together under an empty game object in order to animate them as a collective school as well.



*Several of the fish in a test outside the game scene, controlled by a single game object*

Coins and a counter were added as an additional incentive for the children playing the game. Whenever the diver passes along the path of a coin, the coin disappears and the counter is incremented.

As another small engagement feature, an asset and script was added to make bubbles appear when the screen is tapped. This extra interaction hopefully adds another layer of engagement for the child-patient.

**Technologies Used**

Unity was our main platform for development and deployment of the game. We chose Unity for its built-in rendering engine as well as its potential for group collaboration when combined with GitHub.

We utilized GitHub for version control and easy collaboration among the team. GitHub proved to be crucial for allowing simultaneous contributions to a single game environment.

Gimp was our main tool for modeling assets before further skinning and animating them with Unity’s built-in sprite editor.

**Interface Design (Screenshots)**

Our project consists solely of our final deliverable: the game itself. Because of this, there is not an external interface to connect with our game, instead, the game is the interface itself. As such, the components of this interface are the features of our game, which we have described elsewhere in this document.

**Implementation**

Unity was our base platform of implementation, we started by creating a generic 2D game and adding code written in C# for each of the varying parts of the game.

For the initial movement of the screen the script CameraMovement.cs handles the movement by shifting the camera to view a new portion of the screen further to the right and the background gets re-tiled using the “tiled” option for the background, essentially creating an infinite tiling of the background so that the camera can continue to transition across it.

The diver was implemented by creating layered sprites in GIMP, then piecing these layers together using the built-in sprite editor in Unity. Bones were added in order to rig the sprites from animation. Then animations were added for swimming and idle states, as well as transitions between these two states.

For input control to allow the diver and the screen to progress the script InputController.cs was created. On the two inputs, page-up and page-down, the diver is put into two different states. The page-up trigger causes the diver to transition between one cycle of movement within the environment, while the page-down trigger causes the diver to move at a set interval.

The functionality of the spawning and the diver collecting the coins was implemented in the CoinBehavior.cs script. This script dictates the behavior between the diver and the coin, essentially when they collide with each other and the removal of the coins. The spawning of the coins is handled by the InputController.cs script which spawns coins after the diver has collected one. Whenever a coin is collected, a sound is played to notify the user that a coin has been collected, and the Inventory.cs script is called, which updates the visual counter that is associated with how many coins the user has collected.

One of the final things to be implemented was the addition of bubbles on haptic interaction with the screen. This is handled by the BubbleMovement.cs script, which is notified to spawn a bubble whenever a user provides a touch to the screen. These bubbles spawn and move upwards on the screen until out of view and then they are despawned by the script.

The final implementational feature has to deal with the background of the game. The background of the game was made specifically using Gimp. Each item was generated using a path selection tool and then bucket filled. Afterwards colors that go together were selected using Adobe’s color selection tool. Once the color of the object has been selected the object was then shaded using a dodge/burn tool which makes areas darker or lighter depending on the usage. This worked very effectively in trying to portray shadows on the object from the perceived light passing through the water.

**Lessons Learned**

This project was valuable for furthering each group member’s game development and collaboration skills. Due to the nature of our project, each individual’s contribution was highly dependent on other team members. Working on a single environment required extensive collaboration for all members. GitHub was an incredibly helpful resource for managing everyone’s contributions to the project, be it scripts or assets.

We also learned about setting realistic expectations and goals for our project--although we initially wanted to incorporate live breathing feedback from the ventilator, we soon realized that this was not a feasible goal. This realization was confirmed when we spoke with Joseph Lang, who worked with Stanford Children’s Hospital to help create Sevo the Dragon. As it turned out, even Stanford wasn’t capable of implementing such a feedback system with their ventilator. Designing a slightly more feasible feedback system could have saved us some time with our implementation, although we are glad to have explored and exhausted our possibilities.

**Potential Impact**

We hope this game will be able to greatly alleviate anxiety in children undergoing preoperative anesthesia. It is an incredibly stressful situation for child patients, and we think our app should be able to remove some of this stress.

We also believe this game has potential to ease the task of the nurse delivering anesthesia: if a child is distracted and unstressed during the anesthesia delivery process, it becomes a much easier job for the nurse to encourage the child to breath healthily.

**Future work**

One feature that could be incredibly impactful for patient engagement would be allowing the patient to choose the color of the diver. We are actually planning on adding multiple diver assets to the game repo so that children may have this choice in the near future. We think that even this small amount of choice would give the child an additional level of engagement.

Future work could include developing more scenes and/or giving the children a choice in their experience, for example a space environment, riding in a hot air balloon, etc. With our basic remote-controlled model, any kind of game could be easily laid on top provided new assets and animations were made.

Although it seems incredibly unlikely, it would be great to explore more direct communication between the ventilator and our application, like we had originally planned. Coupled with this, it would be great to explore more direct feedback within the game as far as breathing encouragement goes. The screen could have text prompts to encourage a child to breathe faster, slower, etc.

### **4.3** **Risks**

|  |  |
| --- | --- |
| **Risk** | **Risk Reduction** |
| Children are encouraged to hyperventilate | Nurses will use known demographic data for ideal breathing rates during the process to make appropriate movement triggers. Patients will not be encouraged to breathe faster/deeper if they have already passed the hyperventilation threshold |
| Children are encouraged to hypoventilate | See above. Patients will not be encouraged to breathe slower/shallower if they have already passed the hypoventilation threshold. |
| Children feel additionally stressed using the application | The game uses an incredibly minimal interface to reduce the possibility of creating a stressful/competitive experience. There is no end goal in the application, merely a general goal to maintain consistent breathing. |

### **4.4** **Tasks**

1) Investigate hospital equipment

a. Understand method of data transmission

b. Understand method of data manipulation and calculation

2) Decide which device/software we will use to implement game

a. Apple iPad vs Amazon Fire tablet vs other Android tablet

b. Swift vs Java vs C#

3) Design game

a. Method of communication (wired vs wireless)

b. General game design and functionality

4) Implement game on software

a. Create game assets

b. Create game animations

c. Code bubbles on tap

d. Code coin generation and pickup

e. Code manual and automatic modes

f. Code diver and background movement

5) Test game

a. Hospital trial runs

b. Personal testing

6) Document game

a. Create design document

b. Create final reports

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **4.5** **Schedule**

|  |  |
| --- | --- |
| **Tasks** | **Dates** |
| 1. Background investigation on hospital equipment | 11/14-11/28 |
| 2. Identify software which will be used | 1/15-1/31 |
| 3. Begin Designing game | 2/1-2/14 |
| 4. Begin Implementation of game | 2/15-2/28 |
| 5. Begin Testing of Game | 3/1-3/15 |
| 6. Document | 3/16-4/15 |

### **4.6** **Deliverables**

* Design Document: Contains a listing of each major hardware and software component
* Game source code: Directly from game development software
* Final Report

## 

## 

## 

## 

## **5.0** **Key Personnel**

**Will Baker** – Baker is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed several years of programming courses. He was formerly a Biomedical Engineering major, providing him with special insight into the scope of the project. He was responsible for implementing the different trigger modes that the game has as well as serving as the primary intermediary between the group and ACNW.

**Gavin Glenn** – Glenn is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed several years of programming courses. He was responsible for the creation of environmental assets used to create the underwater environment present within the game.

**Jackson Gregory** – Gregory is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed several years of programming courses, including Game Design and Mobile Programming, two particularly relevant courses. He was responsible for designing and coding the bubbles that serve as the primary way the patient interacts with the game, as well as the mechanism that allows coins to spawn at regular intervals.

**Jared Harris** – Harris is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed several years of programming courses, including Game Design, a particularly relevant course. He was responsible for programming the code for inserting environment elements and implementing camera movement into the game.

**John Ostermueller** – Ostermueller is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed several years of programming courses and has many years of frontend and design experience. He was responsible for the creation of the diver and fish assets and animations that populate the game’s environment.

**John Shelnutt** – Shelnutt is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed several years of programming courses, including Game Design and Mobile Programming, two particularly relevant courses. He was responsible for coding the main mechanics of the game, which includes the implementation of coins that the diver collects and the triggers that determine when the diver moves.

**Staci Allen** – Pediatric Anesthesiologist at Arkansas Children’s Northwest and Assistant Professor at UAMS. Dr. Allen worked with the team by providing requirements for the team as well as feedback on the project.

**Ashlynn Young** – Pre-op and Post-anesthesia Care Unit nurse at Arkansas Children’s Northwest. Young worked with the team in a similar manner to Dr. Allen, providing requirements for the team as well as feedback.

## **6.0** **Facilities and Equipment**

**Arkansas Children’s Northwest** – Contacts at ACNW were critical to project success, as they are the ones who define success by detailing patient needs. Team members visited ACNW under the supervision of the contacts to better understand the process that the game will fit into. This is the only location-specific portion of the project.

**Equipment:**

**GE Avance CS2** – This device is what monitors the patient’s breathing while anesthesia is being administered. The readings of the patient’s breath from this device will be what the nurse examines during the anesthetic process to determine when to trigger movement in the game.

**Amazon Fire Tablets** – The hospital has Amazon Fire Tablets which will be used to run the game for the patient.

**DinoFire Type C/USB Presentation Clicker** – The clicker the game uses to toggle between manual and automatic movement while also triggering movement when the game is set to manual movement.

## **7.0** **References**

[1] Rodriguez, S., Tsui, J.H., Jiang, S.Y. et al. Interactive video game built for mask induction in pediatric patients. Can J Anesth/J Can Anesth 64, 1073–1074 (2017). https://doi.org/10.1007/s12630-017-0922-0

[2] Rodriguez, S., Caruso, T. and Tsui, B. (2017), Bedside Entertainment and Relaxation Theater: size and novelty does matter when using video distraction for perioperative pediatric anxiety. Paediatr Anaesth, 27: 668-669.<https://doi.org/10.1111/pan.13133>