



SciKids

Software Design

Version 2.0

2/6/19

SciKids

Sponsor: Elizabeth Glass

Mentor: Austin Sanders

Samantha Earl, Claudia Coronel, Gwen Morris

Table of Contents

Introduction.....	2
Implementation Overview.....	3
Architectural Overview.....	4
Module and Interface Description.....	6
Implementation Plan.....	11
Conclusion.....	12

1. Introduction

According to the U.S. Department of Commerce, STEM occupations are growing at 17% annually while other occupations are growing at 9.8% annually. These statistics show that there is a growing need for workers with a STEM-related education. Science, technology, engineering, and mathematics play an important role in bettering the future for the U.S. A study from Pew has reported that most Americans believe the k-12 STEM education in the United States is mediocre at best. Some problems associated with this view on STEM education include disinterested students, outdated curriculum and outdated materials. Our client wishes to have a platform that will present resources related to STEM education and career development to a wide range of people. She wants to show the possibilities and benefits that a career in STEM can provide. Overall, the main purpose of this project is to encourage individuals to consider a career that involves STEM. With a student development team of Samantha Earl, Claudia Coronel, and Gwen Morris along with our mentor Austin Sanders, we plan on making a gesture-based program with STEM related content. Our client, Elizabeth Glass, is the director of career development for Northern Arizona University's CEFNS college and was responsible for creating the project proposal. She wishes for this project to be both entertaining and informative, and that is why she, as stated prior, has ultimately decided on a gesture based engagement station for STEM recruitment and career readiness.

Gestures help users to engage in and retain information. Because of this effect, Elizabeth wishes to have gestures incorporated into the project's recruitment technologies. She decided on this when initially seeing the lack of gesture-based technologies that aim to increase interest in STEM. The goal for the team is to create an interactive system where students are introduced to STEM subjects and careers. If an individual chooses a STEM field, they should have resources guiding them through their chosen field. Since there is a wide age range, there will be different modules for different age sections. Younger groups will focus more on interactive and abstract games to pique interest. This can include content like matching games to get kids moving and expose them to STEM related material. Older children (around middle school) will usually have some idea of what type of career they would like to pursue. By showing these children the benefits of STEM, the possibility for a future career in STEM is more likely. Finally, older individuals (those in high school or above) are able to make an informed decision if a STEM career is right for them. They will be in more need of resources that are geared towards career development. This content will include tips for interviews and resumes. In

addition, the content will provide information that older individuals will need as they search for employment and explore the interests they developed as children into a career path to follow. Overall, our solution consists of an interactive system that will be able to educate users on the possibilities of a STEM-based education as well as provide resources to users looking to start a future career. The technologies we will be using to implement our solution will be discussed in the following section.

2. Implementation Overview

The solution that the team will implement is a gesture based learning system that allows individuals to engage in a virtual environment using motions and movements for STEM recruitment. The recruitment will primarily focus on kindergarten through college and community aged individuals. In order to do this, we need a camera that can track user depth and positioning as well as software that can translate user motions into certain gestures that we can later assign actions to.

Our gesture-based system involves 3 main technologies - the Intel D435 Camera, NuiTrack, and Unity. A more detailed description of each technology can be seen below.

- *Intel D435 Camera*: This camera includes a pair of depth sensors, a RGB sensor, and an infrared projector.
- *NuiTrack*: NuiTrack is a middleware that can track a 3D skeletal body and recognizes certain gestures a user makes. By utilizing some of its available modules we will be able to bring value to hand gestures within our system.
- *Unity*: Unity is a game development engine that is compatible with a wide range of technology. We will use this to create the GUI and backend processes, which is made simpler through Unity. NuiTrack also offers Unity support that allows developers to directly map gestures to GUIs made through Unity.

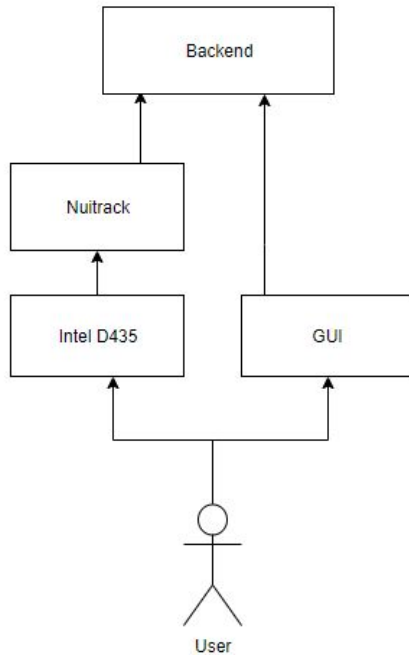


Figure 1: System Diagram

Figure 1: System Diagram above shows how each component interacts with each other. The user will connect the camera to the computer and the software will concur the camera is connected. The user will then interact with the system by the way of our IntelRealsense camera and GUI. The gestures that the camera picks up will then be interpreted through the Nutitrack SDK. The gestures recognized by the SDK will be mapped to specific actions in the backend that will allow the user to interact with the system.

The Intel RealSense camera and the Nutitrack SDK will allow us to track and map user gestures while Unity will allow us to create both the GUI and backend processes. Together, we will be able to create gesture-controlled software that many people can use, as is part of our client's requirements. We will discuss our system overall architecture in the following section before going into detail of each needed component.

3. Architectural Overview

The architecture of this product is to have a user access the system through a hardware set up containing a computer and a depth sensing camera. The user will then run the system using a

standalone build made through Unity, where the GUI will appear and allow the user to select different options for their experience. The core of this project is the navigational system and the local SQLite database. There are also age-based modules that can be edited by future teams.

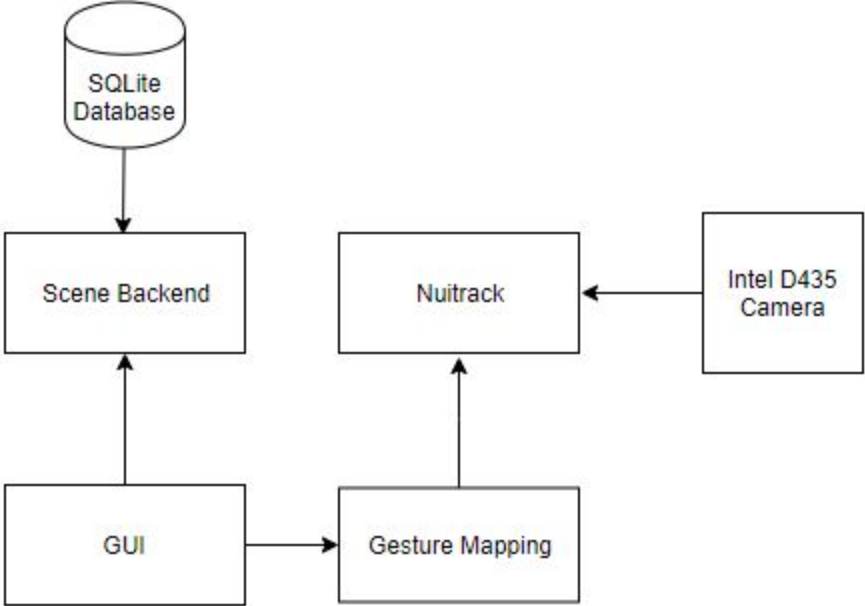


Figure 2: Architecture Diagram

A general overview of the system’s flow is depicted by *Figure 2*. The GUI and backend are created together using Unity and C# in Visual Studio. At first, the user will be able to select options to either login, register, access modules, or learn more about the team. The login and register portion of the software will be completed by using SQLite, an embedded SQL database engine. Since Unity uses scenes to host properties of a specific game, our modules will be configured as different scenes. Another great feature that Unity provides is the ability to import external scenes into a current project. This will allow future programmers to load new content into the modules.

The local SQLite database will be used to store usernames, passwords, and any scores or progress made through modules. This will be done using three connected tables that will interact with each other to keep track of the high scores, what game it corresponds to, and the individual who achieved the high score.

The Intel D435 camera and NuiTrack will be used to read user positioning and gestures. These gestures will then be mapped to specific actions. For example, a closed fist indicates a mouse click and can be used to click buttons that allows the user to navigate through the menu. These actions will be specified through backend scripts.

4. Module and Interface Descriptions

For each of the modules in our system, we will provide a description of its purpose and a diagram outlining the system. The modules we will be discussing in our section include SQLite database, scene backend, gesture mapping, and our GUI.

4.1 SQLite Database

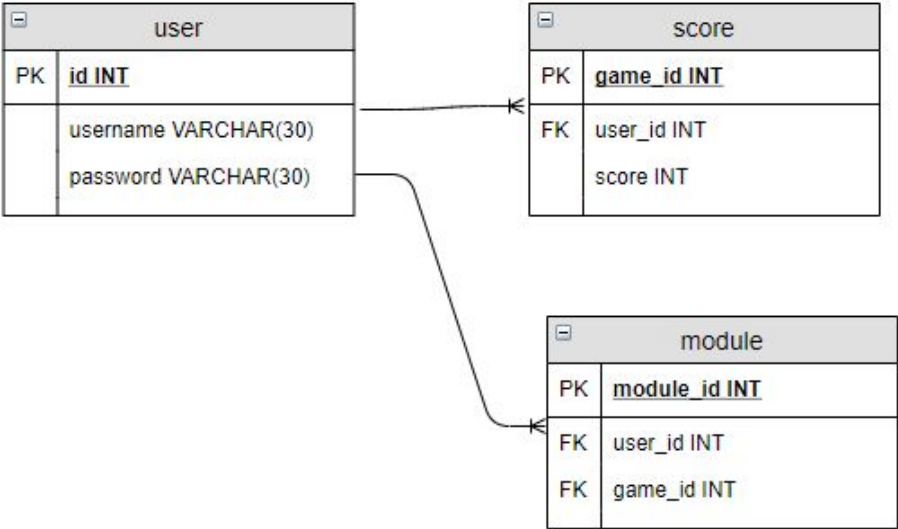


Figure 3: SQLite Database Structure

The database we will be using for our system will contain the tables as shown in *Figure 3*. The system uses a local SQLite database that has 3 tables - “user,” “score,” and “module.” The users table will be accessed through the login and register scenes. The primary key is the user’s id, which automatically increments every time a new user is added to the table. This primary key acts as a foreign key to the other tables. The scores table keeps track of which activities the

user has played and their score in the activity. The module table keeps track of how many users are using a certain module and what activities are inside the module.

4.2 Scene Backend

When creating a new Unity project, we are given an empty scene. Scenes in Unity contain the environments and menus of the project. Each scene file is thought of as a unique layer of the overall structure of the project. In each scene, we place our environments and decorations. We are essentially designing and building our project in pieces with containing different scenes. Different elements of a scene include C# scripts, sprites, prefabs, etc. Our team will be developing different scenes with their own purposes. The following are the various scenes in our system.

- *Login*: The login scene simply takes in a username and password and checks the database for the corresponding entries. The following UML diagrams in *Figure 4* and *Figure 5* shows the needed fields and methods for logging into the system.

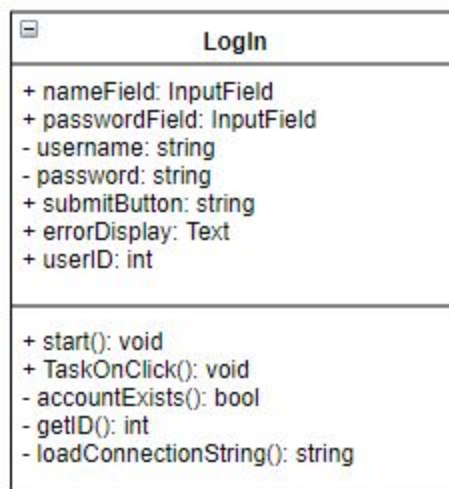


Figure 4: Login UML

Once the “submit” button is clicked, TaskOnClick() checks both InputFields to make sure neither is null. It then calls on loadConnectionString() to get the file path the database is in. The file path changes when the program is being accessed through the Unity editor or a standalone build. The file path is then used to open the database and the text from

both the nameField and passwordField are passed into accountExists(). This method ensures that both the given username and password exist in the database. If the method returns true, the errorDisplay changes to a welcome message and getID() is called to store the user ID for future use in module activities. If accountExists() returns false, the errorDisplay notifies the user that the username or password does not exist.

- *Registration*: This script works similarly to LogIn, except it checks if the username and password aren't already present in the database and later writes to the "users" table. The following UML diagram depicts the Registration class.

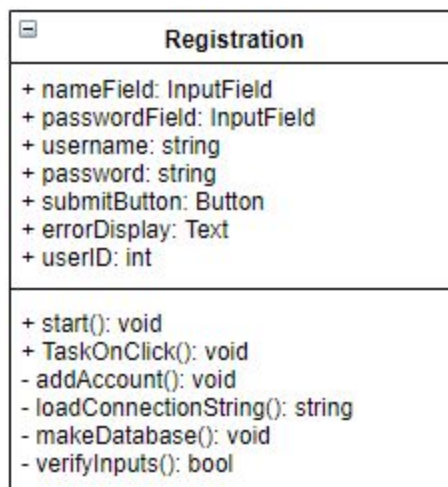


Figure 5: Registration UML

Like in LogIn, TaskOnClick() checks both the nameField and passwordField to make sure neither are null, loads the file path through loadConnectionString(), opens the database, and passes the text from nameField and passwordField into verifyInputs(). verifyInputs() makes sure that the username is between 8 and 30 characters and the password is 5 and 30 characters. It then enters the username and password into the database.

When loading the connection string, this script must now make sure that the database already exists. When the project builds, necessary external tools are installed into the user's local program files. However, Unity does not automatically install the sqlite file into the database, so this must be done manually. loadConnectionString() calls on

makeDatabase() if the database does not already exist, and makeDatabase() creates a new sqlite file with a “users” table with id, username, and password fields. It also creates two more tables, “scores” and “modules,” that are used for module activities.

4.3 Gesture Mapping

For gesture mapping we are using NuiTrack, a full body skeletal tracking software with gesture recognition included. Specifically there is a module titled NuiTrack Hand Tracker, which provides access to user hand tracking capabilities. The main class in this module contains different functions for retrieving the results of the module work, which we can manipulate to perform the desired actions with the help of the API. Currently, the module only supports and can determine the rate of hand clenching and detecting a clicking event. By writing C# scripts and using Hand Tracker’s API we will be able to manipulate the data into our desired outcome.

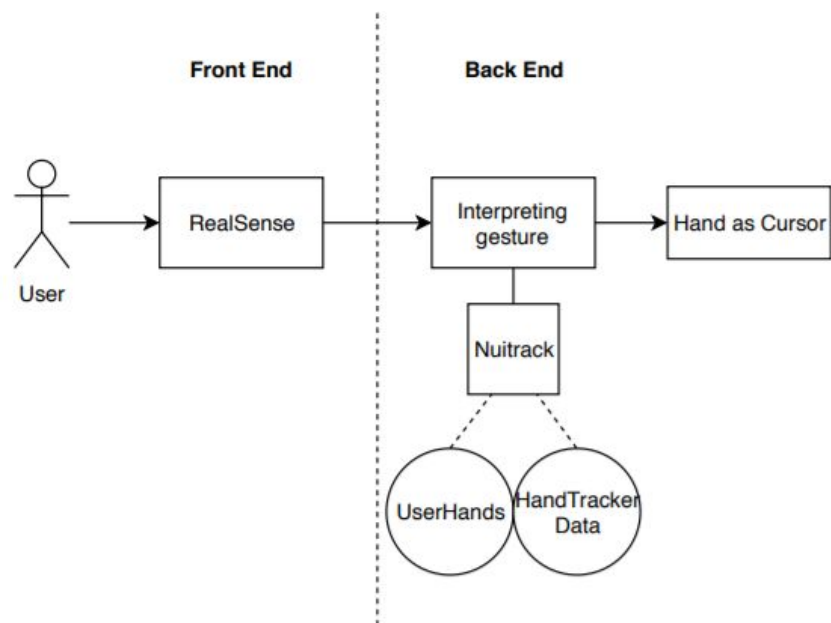


Figure 6 - Gesture Diagram

Figure 6 - Gesture Diagram depicts the process we will be using for gesture recognition. In the NuiTrack Hand Tracker we will specifically be using the UserHands class and the HandTrackerData class. The UserHands class works for storing information about whether the user is using the left hand or the right hand. The HandTrackerData class stores hand tracking results to a specific point in time. At the start of our program our purpose is to identify the users

hands at a certain point in time, and use this data to overlap the mouse cursor using scripts of our own.

4.4 GUI

Unity offers multiple GUI elements that allow us to easily connect buttons, input fields, etc directly to our C# scripts. The main component of our GUI is in the main menu, login, register, and module pages. Each of these pages has the same background, color scheme, and scripts to navigate through the GUI. The following figure, *Figure 7: Menu Layout* shows how each of the pages are connected.

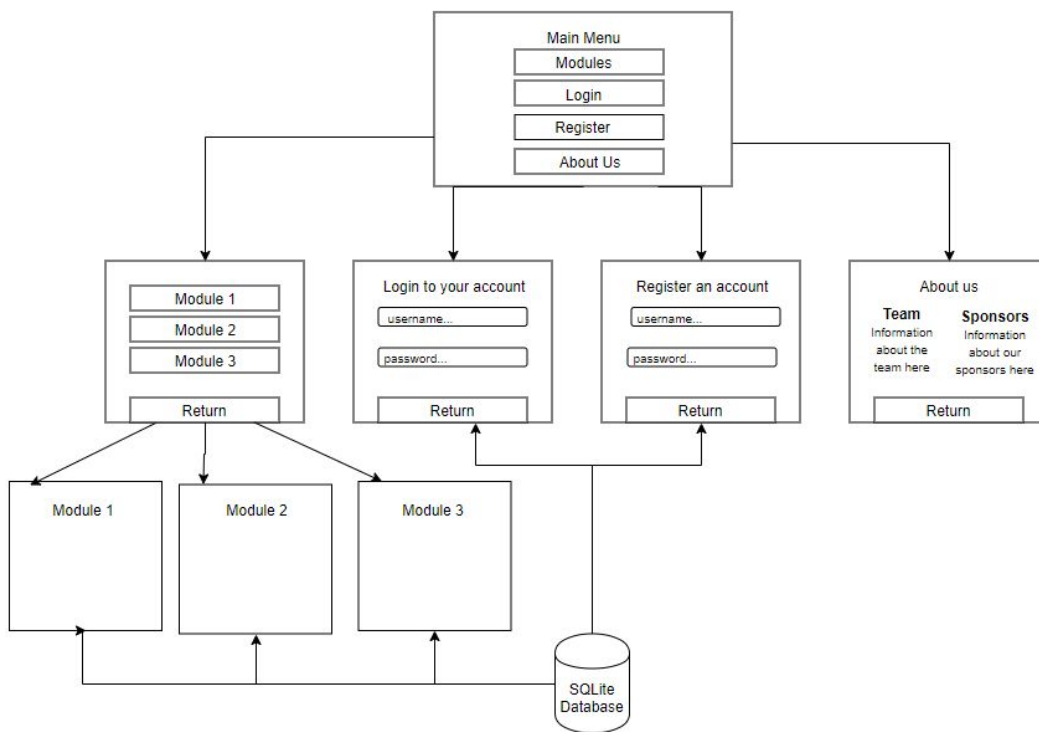


Figure 7: Menu Layout

The scene layout of the GUI are designed to be concise and allow for easy and intuitive navigation by the user. Each panel has the same setup for all of the buttons and the return button is in a similar spot on each scene. The GUI aspect of the navigation system has a consistent look and feel to help the user efficiently get from one scene to another. All scenes are connected with each other indirectly.

5. Implementation Plan

Because this project is being built from the ground up, we must focus on creating the core of the project before expanding it with content and resources. There is a fair amount of overlap in tasks so that we will be able to create a full system by March 11th. Provided below in *Figure 8* is our current progression plan.

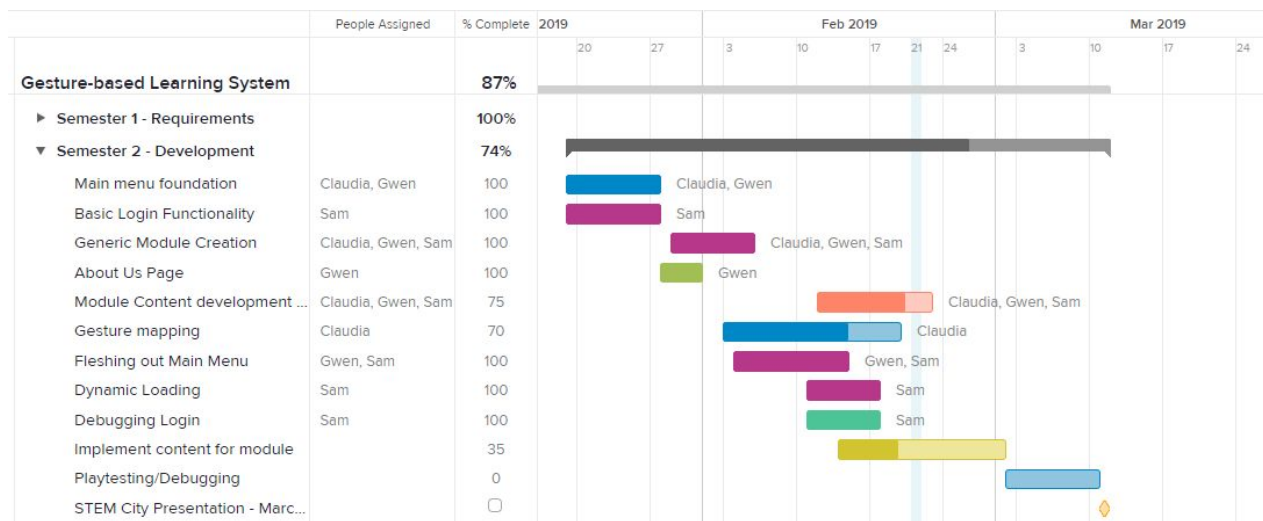


Figure 8 - Gantt Chart

We have first split up the base into 3 main components with a team member working on one component each: the menu navigation and GUI, SQLite database, and gesture mapping. This will together be a basic system that allows the user to register and login to a new account and to navigate the menu through gestures.

The team then plans on developing the main menu with dynamically loading scenes for future developers to simply click and drag scenes into a folder and the scenes to populate the module page. We will be creating more tables in the database to hold user scores and progress for module activities. We will also be optimizing gesture sensitivity to increase usability.

Finally, the team will begin creating more specific modules based off of content supplied by our client. The first module will be an interview-type activity for college students and/or general

community members based on the National Association of Colleges and Employers guidelines. This module most closely relates to our clients profession, so we will be working closely with our client while creating this module.

The second module will be more focused on children from kindergarten to the fifth grade. It will incorporate drag-and-drop gestures in a type of flashcard game. The last module will focus on the grade levels six through twelve. This module focuses more on what careers specifically are available and the introduction to resources for people to develop skills that are important in the job market. These modules will provide resources for the user to learn about STEM concepts and help prepare individuals to be successful finding jobs and in the workplace.

The last thing the team will do is test and debug the system to ensure that we have a minimal viable prototype to showcase at the Flagstaff STEM City public event on March 11, 2019.

6. Conclusion

Scikids is a team that was formed with a goal to provide STEM outreach services to the Flagstaff K-12 and community individuals alongside our sponsor Elizabeth Glass. We are in the process of creating a gesture-based learning system that includes different modules, interactive fun, and personalization. The current problem is that STEM is a quickly growing field without enough individuals going into it. The solution in the Flagstaff community is to create this gesture-based learning system to be easily accessible and portable for K-12 college, and community members. Our client has decided on this solution when seeing how effective gestures can be in helping users retain information and how there is a lack of gesture-based technologies that are currently available. This gesture-based system will allow users to interact with the system with gestures such as open and closed fists as well as grab-and-drag movements (to substitute click-and-drag gestures). They will be able to navigate the menu and engage in age-based modules through these gestures.

With this document we have described the project's tools, overall architectural design, as well as the design of each major module. The technologies being used include the Intel RealSense D435 Depth Camera, NuiTrack, and Unity. Some of the major modules include a SQLite database, scene backend, gesture mapping, and a GUI. By mapping out each of these modules

and their subcomponents we have created a blueprint for our system that we plan to follow. By creating a gantt chart we have strategically allocated the appropriate time for developing all the requirements. By describing a blueprint for our gesture based learning system we are confident in the development as well as the overall outcome to benefit future STEM individuals.