

## CIS/CEN 4914 – Senior Design CISE Department – Spring 2022

### Advisor's Evaluation of Student Project & Performance

It is respectfully requested that the advisor return a score for the final report and project combined. The composite score contains the following components, explained below:

<b>Completeness</b> of Project	.....	10 points
<b>Complexity</b> of Project	.....	10 points
<b>Quality</b> of Results	.....	10 points
<b>Quality</b> of Final Report	.....	<b>20 points</b> ← 20 POINTS, NOT 10 POINTS
-----		
<b>Total Score</b>		50 points

*Completeness* refers to the percent completion. For example, a project that is 60 percent complete would be scored 6/10 in the completeness category.

*Complexity* - Projects should be significantly complex. A *significant* project is one that is novel, well-developed, and useful to the project advisor. For example, a project that involves merely rewriting code or making a single Web page with few innovative features is not considered complex. In contrast, a major database system with Web interface as well as analysis and telecommunications modules would be considered complex.

*Quality of Results* - Do the project results hold together and perform well? Is the code buggy? For example, do Web pages crash or hang? Do the network functions work?

*Quality of Report* - It is recommended that student reports should be graded according to the advisor's understanding of a clear technical communication.

**The Advisor is requested to complete the following form, to be attached by student to student's Final Report prior to submission to course supervisor.**

*This page does not need to be submitted to the course supervisor by the student, but the next page must be attached to the Final Report prior to submission to the course supervisor at the Final Presentation.*

# CIS/CEN 4914 Senior Design - Advisor Scoring Form/Questionnaire

Please Attach This Form to Student's Final Report or Email to mssz@cise.ufl.edu

Semester (circle one): FALL **SPRING** SUMMER

Year: **2022**

Student Name Eric Clayton

UF-ID[5:8] 1 6 5 1

**Part 1. Student's Project.** Please score student's effort numerically. Do not use "A", "B", "C", etc. **Note that the Quality of Final Report is on a scale of 20 points, not 10 points.**

Completeness of Project 10 / 10 points

Complexity of Project 10 / 10 points

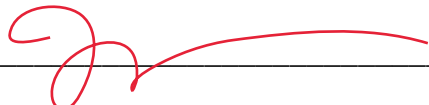
Quality of Results 10 / 10 points

Quality of Final Report 20 / 20 points ← **NOTE: 20 POINTS MAX, NOT 10 PTS**

**Part 2. Student's Performance.** Please circle a number (1 = poor, 2 = fair, 3 = adequate, 4 = good, 5 = excellent) in response to the following questions about your student and your interaction with him/her either individually, or in a design group. **Note: Circling N means that question is not applicable, or that you don't know the answer:**

- |  |   |   |   |   |          |   |
|--|---|---|---|---|----------|---|
| 1. Applied knowledge of math, science, or engineering:               | 1 | 2 | 3 | 4 | <u>5</u> | N |
| 2. Designed/conducted experiments or organized/processed data:       | 1 | 2 | 3 | 4 | <u>5</u> | N |
| 3. Designed systems, components or processes to meet needs:          | 1 | 2 | 3 | 4 | <u>5</u> | N |
| 4. Functioned on multidisciplinary team or interacted well with you: | 1 | 2 | 3 | 4 | <u>5</u> | N |
| 5. Was responsible, diligent, and ethical on this project            | 1 | 2 | 3 | 4 | <u>5</u> | N |
| 6. Used techniques, skills, and tools of modern engineering:         | 1 | 2 | 3 | 4 | <u>5</u> | N |
| 7. Appeared to understand hardware-software interaction :            | 1 | 2 | 3 | 4 | <u>5</u> | N |

Advisor Name Jeremiah Bunch Date 2022/04/19

Advisor Signature 

**Thank you for advising a CISE Senior Design project!**

# Critter Collector

Joshua Bautista, Eric Clayton, Matthew Iwanowski

CIS4914 Spring 2022

Advisor: Dr. Jeremiah J. Blanchard, *email:* [jjb@eng.ufl.edu](mailto:jjb@eng.ufl.edu)  
Department of CISE  
University of Florida, Gainesville, FL 32611

## **Abstract**

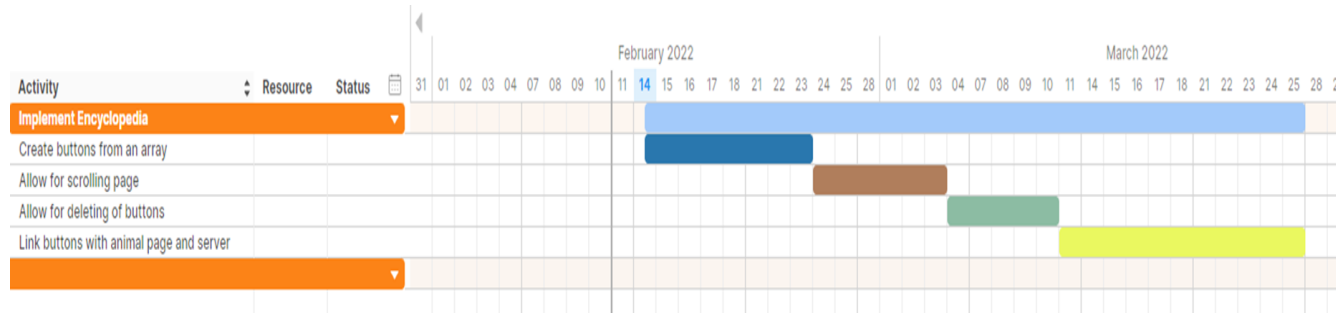
Currently, there are not many options for interactive learning for k-6 about animal taxonomy and classification. Our group has had the opportunity to work on a mobile game that may be a solution. The problem with education games is they are either too boring to students or do not provide any educational value. The method to creating a useful tool for learning is that the students must feel some sense of accomplishment and want to keep playing the game. When looking for the perfect formula for this method we look to what works and try to emulate it. One such game that became extremely popular was Pokémon Go™. Pokémon Go™ is a game about catching Pokémon, little monsters with special abilities. Pokémon can be caught by walking around and finding them on a map of the player's real-life location. The goal of this game is to catch as many Pokémon as they can to fill their Pokédex which is just a collection of Pokémon.

With the success and popularity of the game Pokémon Go™, we looked to create a game that uses its formula to apply to real-life animals instead. This game is called Critter Collector. Critter Collector is a game made for k-6 students that is primarily a mobile game. In the game, the player will find animals that live near them, and catching them will give the player more information about that animal. Thus, keeping the formula of a great game while also providing education for the player. The game was first created with the Unity engine, but our team was tasked to transfer this game to Unreal Engine 4. The game uses a backend server with the VaRest plugin to store animals and player information as well as spawn data for the animals.

### **1. Introduction.**

In this project, we set our goal to successfully transfer the work done from previous groups in Unity to Unreal Engine. Over several months, we managed to bring the state of the game adjacent to its version on Unity. The motivation for this project was my passion for video games. The popularity of games with young students has been increasing, and a game that also encourages education is very promising. We felt the direction of the project was solid and aligned with the author's motivations as a future game developer. The group members for this project are Matthew Iwanowski, who is a computer science major and worked on the backend of the project, Joshua Bautista, who is a computer

science major and worked on the frontend and Eric Clayton the author of this report and a computer science major who worked on the backend.



Above is the author's schedule of the work the author completed, he worked on the encyclopedia page of the app, which he will cover later in this report. His first task was to create buttons dynamically from an array of animals which took a week or so. His next task was to implement these buttons in a scroll box so that the buttons could be scrolled through. The next task was to allow for deleting of these buttons. Finally, he had to link the front end with the backend and the animal page. The problems faced were how to implement the scrolling box as we had to decide on a design that would work best with the app.

### 1.1. Problem Domain.

The problem of this project comes from the intersection of game development and education. The problem of education that it addresses is how to get students k-6 interested in animal taxonomy and classification. With education games, there is a delicate balance between fun and a valuable learning experience. Critter Collector addresses this problem by using the popular formula from Pokémon Go™ of collecting animals and trying to catch as many as possible. Using animal information as the reward for catching these animals. This leads to the game development aspect which is the use of Unreal Engine and the client/server architecture to create a mobile game that is also educational.

### 1.2. Literature Overview.

When the author began working on this project, he was very new to Unreal Engine 4 and needed additional information to help him complete the work. The Unreal Engine forums are a great resource that was used often. The Unreal Engine forums are filled with like-minded people who are also working on another project and might encounter the

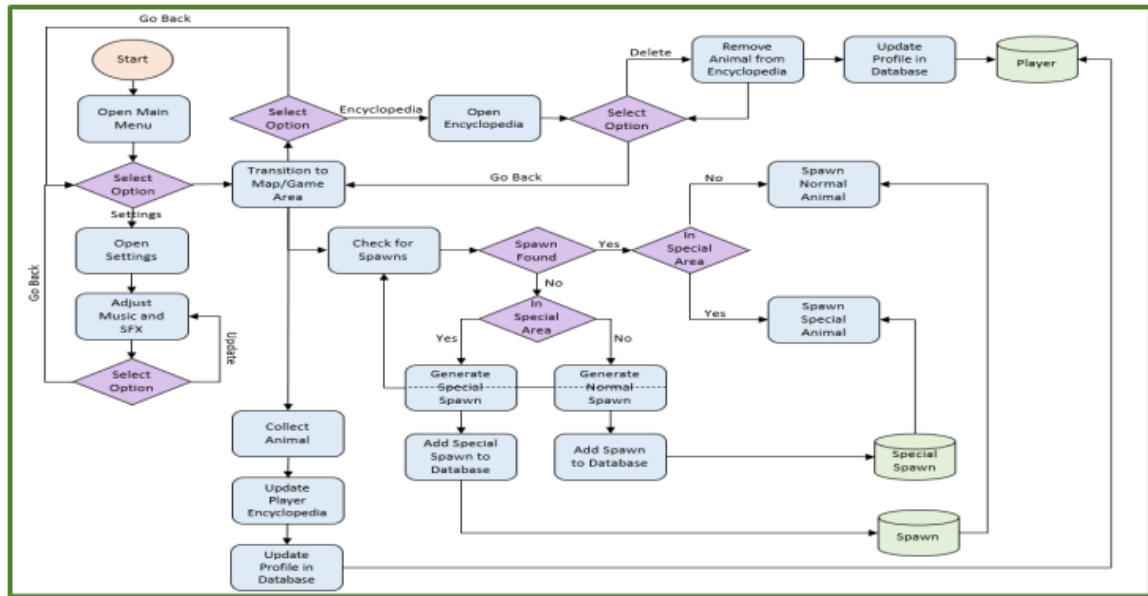
same errors. When learning a new piece of technology such as Unreal Engine it's always helpful to have someone who understands it better than you to help. The second type of source required to understand Unreal Engine was reading through the documentation.

The first source that was used was a post by xlar8or[3] on the Unreal Engine forums called "How Can I Add Buttons Dynamically on My UMG Widget?" To create a page with buttons on every animal the player collected, there was a need to make buttons from an array that changes as the player adds and deletes animals. Creating static buttons wouldn't help here so dynamically added buttons were required. This post goes into detail on how buttons can be added dynamically to a project using an information struct and blueprints. The second source used was another forum post by Victor\_Matos[2] on the Unreal Engine forums. A way to be able to scroll through the buttons created was essential. This post details the method of linking the buttons to a scroll box. The final source that was used to familiarize the author with Unreal Engine was the Unreal Engine documentation on the official website.

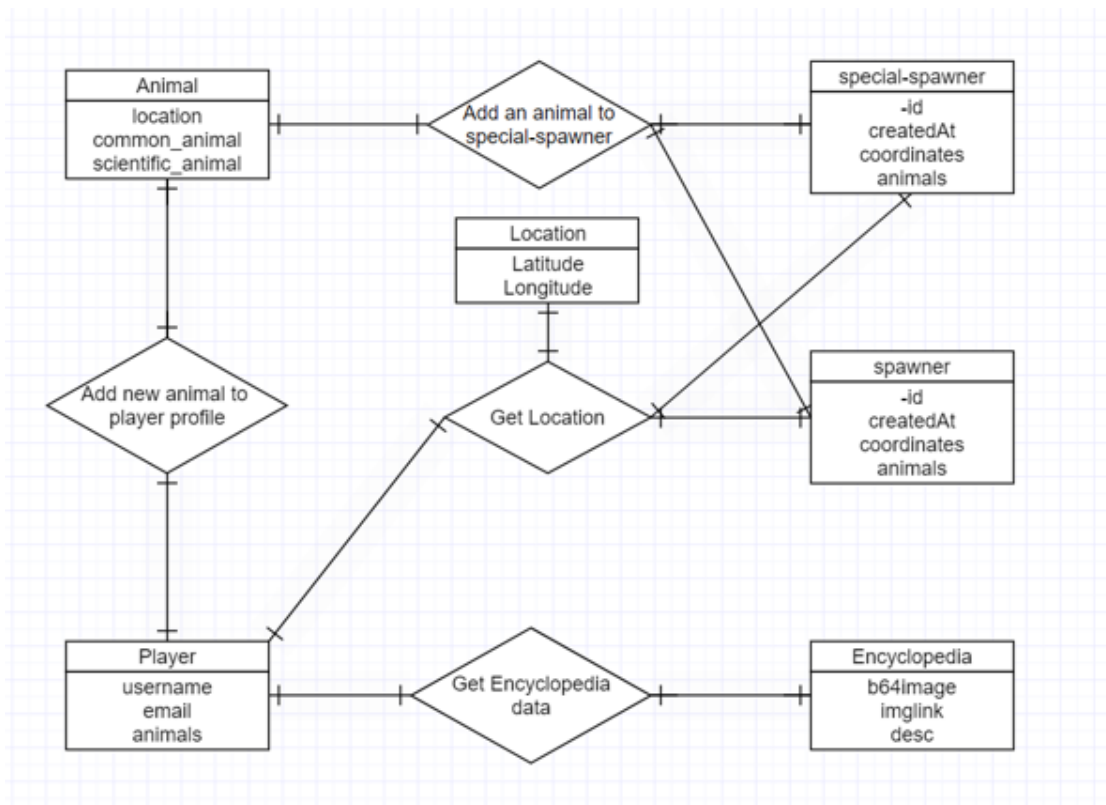
## **2. Technical Approach (Solution).**

To create an educational game for students k-6 to learn about animal taxonomy and classification Critter Collector is a mobile game that allows players to learn while having fun collecting animals in their area. Critter Collector is created in Unreal Engine with C++ and has been successfully implemented for android devices with location services and real-time map integration. Critter collector allows for finding animals based on the user's location that they could find by going out on their own and looking for them outside. When an animal is caught, it is sent to that player's encyclopedia page where they can click on an animal to learn more about it.

The game uses a backend server to generate the animals that are located on the player's map. The game also has some basic settings like sound effect volume and music volume. The animals on the map are generated from a backend server from a default spawner and a special spawner. The default spawner creates new animals around the player's general location while the special spawner creates animals based on a special area. Special areas are typically points of interest like "Lake Alice." The flow chart of the project is shown in Figure 1.



**Figure 1.** Flow chart of Critter Collector, the player starts at the top left where the main menu begins. From there the user has the option to open the settings to adjust their settings or go to the game area where animals can be found. From the map, the player can choose to open the Encyclopedia where all the names of the animals collected can be found. The player has the option to delete any of these animals they want.



**Figure 2.** The E/R diagram for Critter Collector

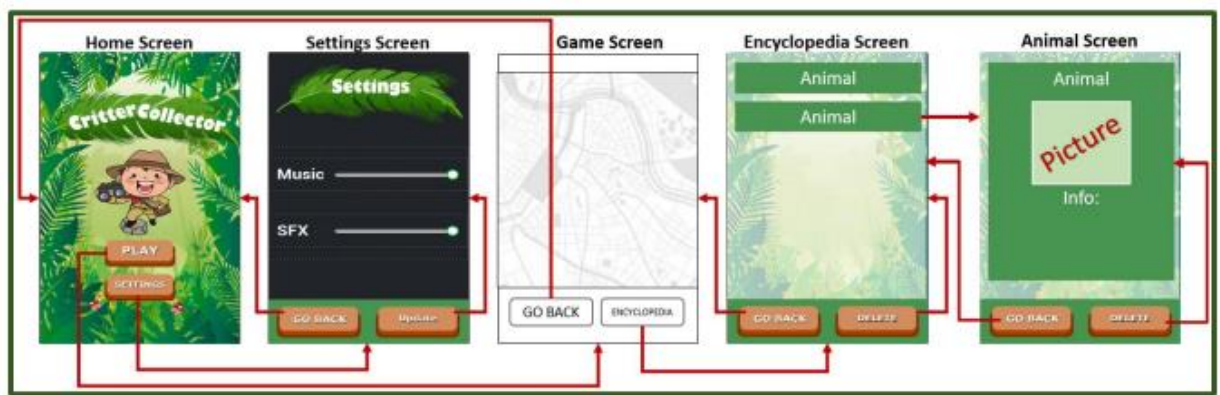
In figure 2 there is the animal entity that comes with the information that the player can learn more about like the common animal's name and scientific animal name. We plan to link this to the Wikipedia page for the animal to provide even more information about the animal. The animal entity also holds the location of where that animal was caught. The animal entity can be added to the special spawner and the default spawner. The spawners have the same information except the special spawner has a location name when requested by the player. Location is comprised of latitude and longitude

Testing and debugging of Critter Collector included trying several animals and testing deleting them from the encyclopedia page and checking whether all information was synchronous with the backend. We also did testing on whether the spawner and special spawner are generating animals with the correct information.



### 3. Results

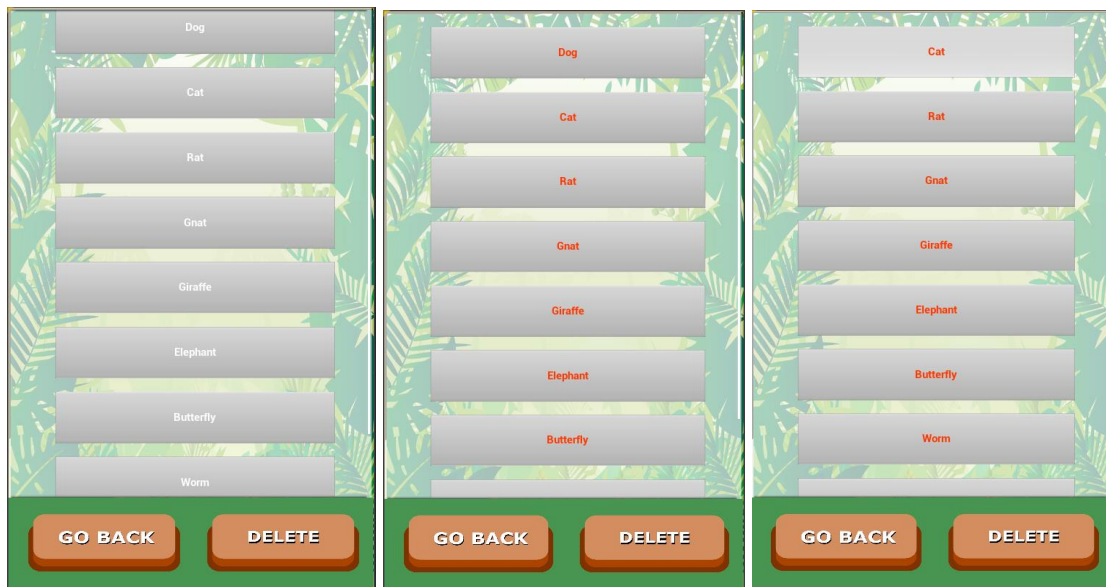
After testing and debugging the game, we arrived at the final product while there is still more work to be done in the future, we are pleased with the results we have accomplished. In the game's current state, the animals do not have textures, but we plan to implement 2d animal sprites for each type of animal e.g., bird, snake, insect, etc. Currently, animals that spawn are represented as a block but can easily be replaced by the new models. The user interface for the game can be seen below in figure 2.



**Figure 3.** GUI of critter collector

In figure 2, the left side is the beginning of the game when the player opens the app from there the player has several options. The first option is “play” which takes the player to the game screen. The other option is “settings” which takes the player to the settings page. In the settings screen, there are two volume sliders where the player can change the volumes of the music and sound effects and press the update button to save the changes or go back to the main menu. The game screen is where the player can see the map of their location including 3D buildings and the animals that spawn around them clicking on the animal allows the player to catch the animal which sends it to the encyclopedia screen. From the game screen, the player has the option to go back to the main menu or go to the encyclopedia screen. In the encyclopedia screen, the animals the player caught appear here which are saved in the backend and are converted to an array of animal names and are thus converted to

buttons that display here. When a user taps an animal name it takes them to an animal's page where a picture and information are taken from Wikipedia and displayed. The animal page is currently not developed yet and will have to be listed under future work to be done for the project. From the encyclopedia page, the user has the option to delete an animal from their saved animals. To demonstrate this Figure 3 has test values for the animals that are shown to be deleted.



**Figure 4.** Process of deleting an animal

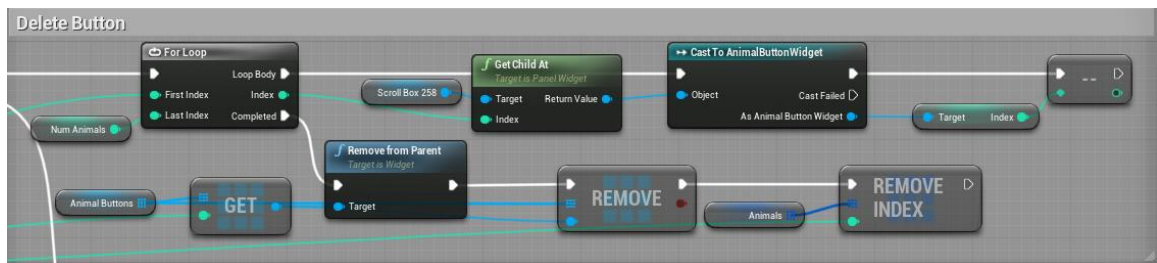
Above the process of deleting an animal from the encyclopedia screen is shown. First, we can see the unaltered list of animals on the left then the player presses the delete button and the names turn red indicating that it is in delete mode then the player presses “dog” and the “dog” button is removed from the list. These screenshots come from the game running in real-time using test values for the animal names.

#### **4. Technical Challenges and Solutions**

The first technical challenge we faced was trying to find a suitable solution to implementing the textures for the animals that spawn in. There are two subproblems in this challenge the first was whether we wanted to do specific textures for each animal or just a general non-specific texture for each type of animal. The other subproblem was whether we wanted to use 3D or 2D textures. There are a limited

number of textures for animals that are free to use and since we cannot make more, we decided to use 2D nonspecific textures as those were the most readily available.

The second technical challenge was implementing the deleting of the animal buttons when deleting the animal, the reference to the index that the animal that is deleted is removed but the animals after that do not have their index updated to be one less. So, when deleting there would be a bug that causes the wrong button to be deleted. To fix this problem I would update the index of each button that came after the button was deleted to a value one less than before to keep the indexes the same between the array of buttons and the button itself. This process can be seen implemented in Unreal Engine in figure 5.



**Figure 5.** Process of updating the indexes after deleting an animal.

## 5. Conclusions

During this project I worked solely on the implementation of the encyclopedia screen creating the buttons, fixing them to a scroll box, allowing for deleting and changing the text of the buttons to red in delete mode. I began this project knowing very little of Unreal Engine and through documentation and forum posts I learned more about how blueprints function. I had to use my knowledge of C++ to understand blueprints it was like learning a new programming language with the same systems as loops, objects, and conditional statements. I also learned more about HTTP calls to the backend server although most of my work was frontend implementation. There is still work to be implanted as the animal textures and the animal screen still have yet to be implemented.

## 6. Standards and Constraints

*Constraints:* Critter Collector was made with Unreal Engine version 4.25.4, Android Studio 4.0, and Java JDK 8

## **7. Acknowledgements**

The author would like to thank his advisor, Dr. Jeremiah Blanchard for his guidance, advice, and encouragement toward the successful completion of this project. Additional thanks go to Samuel Marrero who helped with the project and implemented maps on the frontend, and all the team members who worked on Critter Collector this semester.

## **8. References**

- [1] Anon. "Samples and Tutorials.",  
<https://docs.unrealengine.com/4.27/en-US/Resources/>.  
Unreal Engine Documentation
- [2] Victor\_Matos. "How to Implement Mobile Swipe Scroll for UMG SCROLLBOX.", ,  
<https://forums.unrealengine.com/t/how-to-implement-mobile-swipe-scroll-for-umg-scrollbox/23264>. Unreal Engine  
Forums (as-of 7 Mar. 2015)
- [3] xlar8or. "How Can I Add Buttons Dynamically on My UMG Widget?",  
<https://forums.unrealengine.com/t/how-can-i-add-buttons-dynamically-on-my-umg-widget/292424/2>, Unreal  
Engine Forums, (as-of 30 Aug. 2014)

**Biography**

Eric M. Clayton was born in Clearwater, Florida on February 10, 1999, He completed his secondary education at Countryside High School and is completing his baccalaureate degree in Computer Science at the University of Florida (Gainesville, FL), where he expects to graduate on April 30, 2022. Mr. Clayton is an avid computer programmer, with an interest in the game development field, and is proficient in C++ and Java. Previously he has worked at an ice cream shop called Phenomenom. He hopes to get a job in the video game industry where his knowledge of C++ and passion for gaming can be utilized.

This questionnaire will help us assess how we are improving Senior Design course with respect to your achievements and our accreditation requirements. **Completion is mandatory** for all students enrolled in CIS/CEN 4914, beginning Fall 2005. Items in brackets ([ ]) are for faculty only.

Please use MS-Word to fill in your questionnaire, print it out, and make it the LAST PAGE of your Final Report (does not count toward FinRpt 11-page maximum length). **NO HANDWRITING PLS**

① Enter Your Name Eric Clayton      ② Enter UFID[5:8]: 1 6 5 1

③ Circle Your Major: CPE CSE DAS **(CSC)**

④ UF-Online Student? Circle: Yes **(No)**

**Q1. What was your Senior Project or CISE Design 2 project about?** (1-2 sentences)

*Answer:* My Senior Project is about a mobile game called Critter Collector. It is about teaching k-6 students about collecting virtual animals to learn more about them.

**Q2. What impact do you think a project or topic like this has on your responsibilities re: engineering in particular, and on society in general?** (2-3 sentences) [AB Out f)]

*Answer:* I think it has a huge impact as this project is made to teach a younger generation. If this game becomes popular then my decisions as a programmer and engineer would have considerable weight on the future generations.

**Q3. What impact does your project, or research in the area of your project, have on the global community in general, and on our society in particular?** (1-3 sentences) [AB Out h)]

*Answer:* Similar to the last question, I think that having an impact on people's developmental education is a huge impact that may inspire a student to become a biologist or get them interested in learning in general.

**Q4. How important would it be to continue learning about engineering, upgrading your skills throughout your lifetime of employment? Why is this important?** (1-2 sentences) [AB Out i)]

*Answer:* I think a true master is an eternal student and that there's always room to improve on things and make the world a better place. And making the world faster, more connected, more fun, safer, etc. is a goal humanity never stops chasing.

**Q5. What contemporary issues in society and computer science/engineering do you see as being related to your project, or its topical area?** (1-3 sentences) [AB Out j)]

*Answer:* The issues my project attempts to address are the education of the youth in animal taxonomy and classification and to make learning more fun in general.