As technology advances and evolves at a progressive pace, the capability to meet the demands in the software industry ultimately depends on the number of individuals who decide to join and work in the Computer Science (CS) field. Unfortunately, Computing and Information Sciences majors are the least popular amongst students due to a projected difficulty on the abstraction involved in the learning of fundamental courses. As a result, by 2024, the United States will be unable to fulfill approximately 1 million Computing positions which will slow our technological advancement and heavily affect the United States economy.

### Problem
- As technology advances and evolves at a progressive pace, the capability to meet the demands in the software industry ultimately depends on the number of individuals who decide to join and work in the Computer Science (CS) field.

### Solution
- **Code VR 1.0**, is a free virtual reality application designed to inspire students to join the computer science field which should increase the number of computer professionals and satisfy the increasing demand for software solutions.
- **Code VR 1.0** inspire students by introducing them to the logic of programming in a way that remove the projected difficulty on the abstraction involved in the learning of fundamental courses.
- **Code VR** do so by letting them, create small code components, see the flow of execution of commonly known algorithms through animation and lastly upload a python source code and visualize a 3D representation of the code.

### Current System
- During the fall semester of 2017, with the support of the OpenHID Lab, the previous Code VR team solved the problem by generating 3D objects to represent the elements of a Json tree generated by a parser from a simple Python code which was uploaded to a python server.
- Unfortunately, the current system doesn't have a variety of 3D shapes and colors, it doesn’t have a pause, resume, and quit functionality, grab functionality, algorithm flow of execution animation, 3D object generator, relationship view and lastly code export functionality.

### Verification
- Code-VR object design is based on the data collected from a survey we administered to freshmen and non computer science students to find out what would facilitate their learning.

### System Design

![fig4: Code VR system use case Diagram](image)

- **System Design**
  - Code VR 1.0 was implemented using Agile software development with Scrum as a framework for managing the process.
  - I implemented a set of new 3D shapes using Unreal Engine Static Mesh editor.
  - I implemented Pause and Resume functionality using Unreal Engine widget, horizontal and vertical boxes, button and text block with c++ in the back end for the functions.
  - I implemented a grab/drop functionality using Unreal Engine Static Mesh editor.
  - I implemented a relationship viewer using Unreal Engine Particle System and c++ library.
  - I implemented a Set of functions for a dynamic UX.

### Object Design

- The United States has an increasing demand for Computer Science professional due to a need for software solutions. Unfortunately, Computing and Information Sciences majors are the least popular amongst students. This slows our technological advancement and heavily affect the United States economy.

- **Code VR 1.0** attempt to solve this problem by facilitating the CS experience using VR:
  1. **Code VR** provides 3D representation of code.
  2. **Code VR** provides a Set of functions for a dynamic UX.
  3. **Code VR** provides animation of popular algorithms execution.

- The user understands the logic of programming, joins the CS field, increases the number of CS professional and satisfies the increasing demand for software solutions.

### Implementation

- Code VR 1.0 Prototype

![fig1: Generation of 3D objects based on python file](image)

![fig2: 3D objects manipulation](image)

- **Implementation**
  - Code VR 0.1 Prototype
  - Code VR 0.2 Prototype
  - Code VR 0.3 Prototype
  - Code VR 0.4 Prototype
  - Code VR 0.5 Prototype
  - Code VR 0.6 Prototype
  - Code VR 0.7 Prototype
  - Code VR 0.8 Prototype
  - Code VR 0.9 Prototype
  - Code VR 1.0 Prototype

- **Requirements**
  - As a developer, I shall implement a set of functionalities such as: “a set of new 3D shapes, a grabber, a relationship viewer, a 3D objects generator, pause, and resume” so that the code VR users can have a more dynamic experience.
  - The users shall use the functionalities to create 3D objects to represent coding components, visualize objects relationships, grab objects, move objects, pause or resume session inside of the code-VR environment.
  - The Code VR application environment shall be intuitive for the user so that he/she can easily defeat his/her fear of the computing field.
  - The Code VR application shall respond to user-triggered actions in no more than 1 second.
  - The user shall have a better understanding of the logic behind programming which should inspire him/her to consider computer science as a field for a future career after, using the code-VR application.

- **Verification**
  - We have used the functional and nonfunctional requirements of the product owner’s user stories to generate a set of unit test cases with a 60% minimum coverage.

- **Summary**
  - The material presented in this poster is based upon the work supported by Dr. Francisco Ortega, Dr. Juan Caraballo and OpenHID Lab. I am thankful to the help that I received from my group members, Andres Chalela, Miguel Jardines.