Hello World

Computer Science Senior Design Project

University of Cincinnati – Class of 2018

Matthew Tucker

Faculty Advisor: Gowtham Atluri

Table of Contents

Contents

Project Description 3
Abstract 3
Project Details 3
User Interface Specification 4
Design Diagrams 5
User Stories 5
Test Plan and Results 6
Overall Test Plan 6
User Manual 9
Frequently Asked Questions 9
What is this game about? 9
I have no programming experience - can I play? 9
Who is this game designed for? 9
Spring Final Oral PowerPoint Presentation 10
Final Expo Poster 15
Assessments 16
Initial Self-Assessment (fall semester) 16
Final Self-Assessment (spring semester) 18
Summary of Hours and Justification 20
Final thoughts and GitHub Link 22

Project Description

Description of my senior design project – Hello World.

Abstract

Hello World is a video game designed to teach its players very basic programming concepts. The game allows the players to manipulate objects in the world by modifying pseudocode provided to them. Tutorials are provided to the player that teach four main programming concepts: variables, conditional statements, functions and loops. Each concept taught allows the players to manipulate the world in a different fashion. Players are able to completely redesign the world they exist in by changing the size and position of many objects in the game. Through this manipulation of the environment, players gain experience in basic programming concepts and are able to see how their action affect the world they exist in. Hello World was created in Unity game engine and all of the in-game interactions were coded using C# scripts.

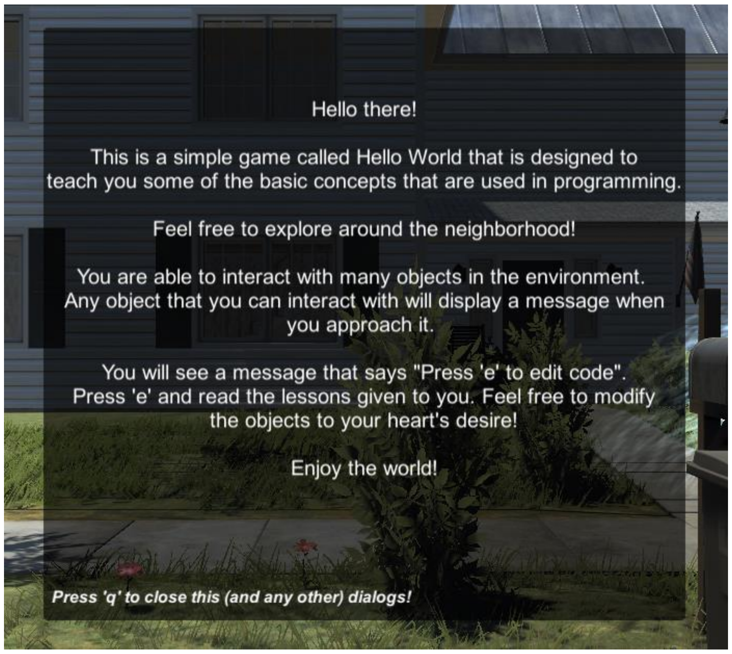
Project Details

- Unity game engine
- First person perspective
- WASD and mouse movement
- Environmental based tutorial game
- 3D
- Single Player

User Interface Specification

The user interface of Hello World is very important because the user is in control of many of the aspects in the game. I wanted to design a user interface that was simple enough for anyone to use and wasn't overwhelming to the player.

The game opens with a short introductory message to tell the player a little about the game and what they can do.



Hello there!

This is a simple game called Hello World that is designed to teach you some of the basic concepts that are used in programming.

Feel free to explore around the neighborhood!

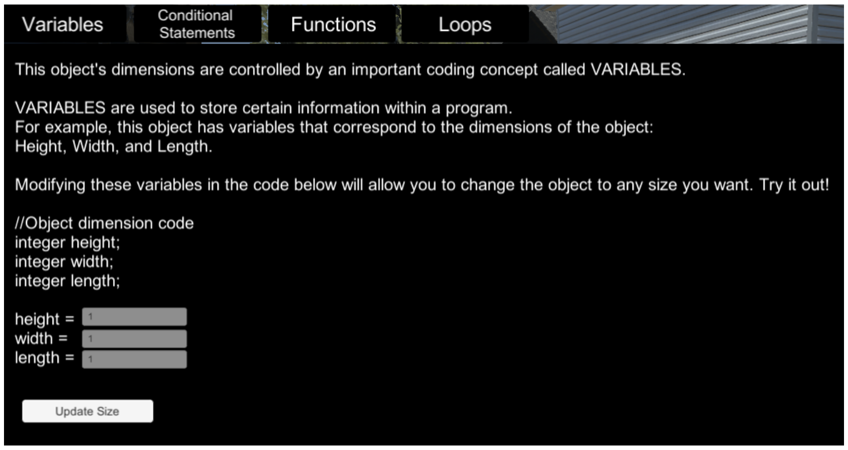
You are able to interact with many objects in the environment. Any object that you can interact with will display a message when you approach it.

You will see a message that says "Press 'e' to edit code". Press 'e' and read the lessons given to you. Feel free to modify the objects to your heart's desire!

Enjoy the world!

Press 'q' to close this (and any other) dialogs!

When interacting with an object, the player sees the screen below:



This object's dimensions are controlled by an important coding concept called VARIABLES.

VARIABLES are used to store certain information within a program.
For example, this object has variables that correspond to the dimensions of the object:
Height, Width, and Length.

Modifying these variables in the code below will allow you to change the object to any size you want. Try it out!

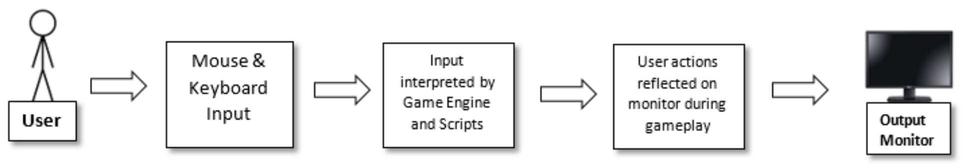
```
//Object dimension code
integer height;
integer width;
integer length;

height = 1
width = 1
length = 1
```

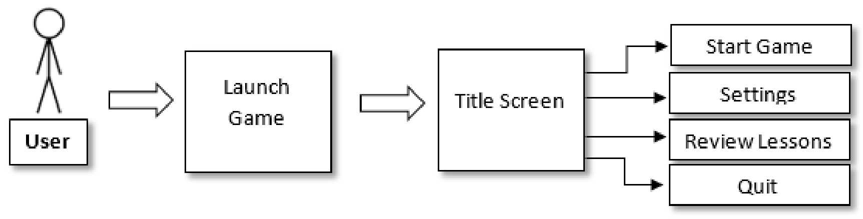
Each of the tabs shown at the top of the screen are the individual lessons for that object. The player can select any of the tabs to see a tutorial about the selected programming concepts. Each tutorial gives a brief description of the programming concept and allows the player to manipulate the object in a specific way.

Design Diagrams

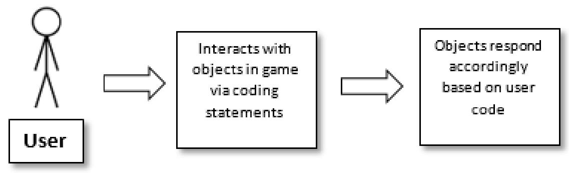
Goal: Design a game that meshing learning programming with an enjoyable gaming experience.

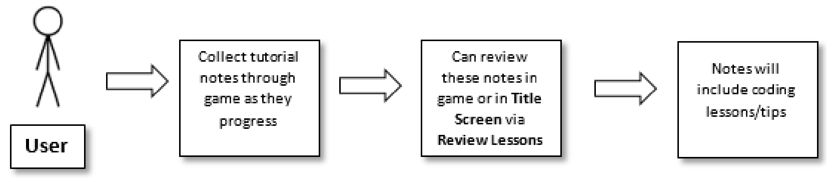
D0: High Level Design Diagrams – shows basic input and output along with high level game design plans


User
Mouse & Keyboard Input
Input interpreted by Game Engine and Scripts
User actions reflected on monitor during gameplay
Output Monitor



D1/D2: Lower Level Design Diagrams – shows more specifics on game design


User
Interacts with objects in game via coding statements
Objects respond accordingly based on user code


User
Collect tutorial notes through game as they progress
Can review these notes in game or in Title Screen via Review Lessons
Notes will include coding lessons/tips

User Stories

As a gamer, I want to explore an interesting environment and have fun playing the game.

As a student, I want to enjoy what I am learning, so that I can look forward to studying and retain the knowledge better.

As a programmer, I want to be able to experiment and show off my skills, so that I can use my creations as examples of my skill level.

Test Plan and Results

The following describes the testing process for Hello World.

Overall Test Plan

I will use thorough play tests to ensure that Hello World functions properly in all situations and that there are no game breaking issues that arise throughout gameplay. A series of guided play tests will be used to make sure the player has no major issues while playing the game. Playtest will include walking around in the environment (Test case 1), interacting with objects in game (Case 2), analyzing the dialog/tutorial messages (Case 3), testing the boundaries (Case 4), and options/main menu functionality (Case 5). A final test case will be administered to ensure the game runs smoothly on a non-development system (Case 6).

Test Case Descriptions

The specifics of the six test cases are described below.

TC1: Walking in environment
TC1.2: The purpose of this test is to make sure the player can successfully walk around in the environment without having any mechanical/graphical issues.
TC1.3: The tester will load into the game and make sure they can walk and look around using the mouse and WASD keys. They will inspect the environment and make sure no graphical issues are present along with ensuring they are always in control of their character.
TC1.4: The inputs for this test are the WASD keys and mouse movement.
TC1.5: The character movements should reflect the inputs of the tester. There should be collision on all objects in the environment with the player (they can't walk through cars/trees). All graphics should remain stable throughout playtest.
TC1.6: Normal
TC1.7: Blackbox
TC1.8: Functional
TC1.9: Integration
Results: The player is able to successfully walk around in the environment without having any graphical issues. There were some mechanical issues where players can 'stick' to certain objects in the world. The majority of these issues have been fixed but there are still some cases of players having mechanical issues.

TC2: Interacting with objects
TC2.2: The purpose of this test is to ensure that the player can successfully interact with interactable objects in the game.
TC2.3: The tester will load into the game and need to walk up and press the E key on certain objects in the world. Certain objects will be interactable and the user will need to make sure that these objects have a response when the user looks at them and presses the E key.
TC2.4: The inputs for this test are the WASD keys, mouse movement, and the E key.
TC2.5: The interactable objects should have a specific response when the tester presses the E key while looking at the object. In most cases – a text dialog will pop up when the user interacts with the object.

TC2.6: Normal
TC2.7: Blackbox
TC2.8: Functional
TC2.9: Integration
Results: The player can successfully interact with appropriate objects in the game.

TC3: Analyzing dialog/tutorial messages
TC3.2: The purpose of this test is to ensure the accuracy and spelling of all the dialog messages in the game.
TC3.3: The tester will load into the game and need to interact with every interactable object in the game. This test should be done after or along with TC2. Every object that provides text will need to be analyzed for accuracy and spelling.
TC3.4: The inputs for this test are the WASD keys, mouse movement, and the E key.
TC3.5: The accuracy of all the dialog/tutorial messages in the game should be correct. There should be no spelling errors throughout the game.
TC3.6: Normal
TC3.7: Blackbox
TC3.8: Functional
TC3.9: Unit
Results: The dialog/tutorial messages are accurate and there are no spelling issues.

TC4: Testing the boundaries
TC4.2: The purpose of this test is to make sure the boundaries of the game cannot be broken.
TC4.3: The tester will load into the game and need to walk to the edge of the environment and attempt to pass through the invisible barriers in the world.
TC4.4: The inputs for this test are the WASD keys and mouse movement.
TC4.5: The boundaries of the game should not be able to be passed through and the tester is confined to the world.
TC4.6: Normal
TC4.7: Whitebox
TC4.8: Functional
TC4.9: Unit
Results: The boundaries of the game are solid and the player cannot leave the play area.

TC5: Options/Main menu functionality
TC5.2: The purpose of this test is to make sure that all the options and main menu options function properly.
TC5.3: The tester will select every option from the main menu and make sure that they function and do what is expected of them.
TC5.4: The inputs for this test are the mouse movement and mouse clicks.
TC5.5: The options should do what is expected of them. All main menu options should be interactable and carry out their specific tasks.
TC5.6: Normal
TC5.7: Blackbox:

TC5.8: Functional
TC5.9: Unit
Results: All of the menu options function properly.

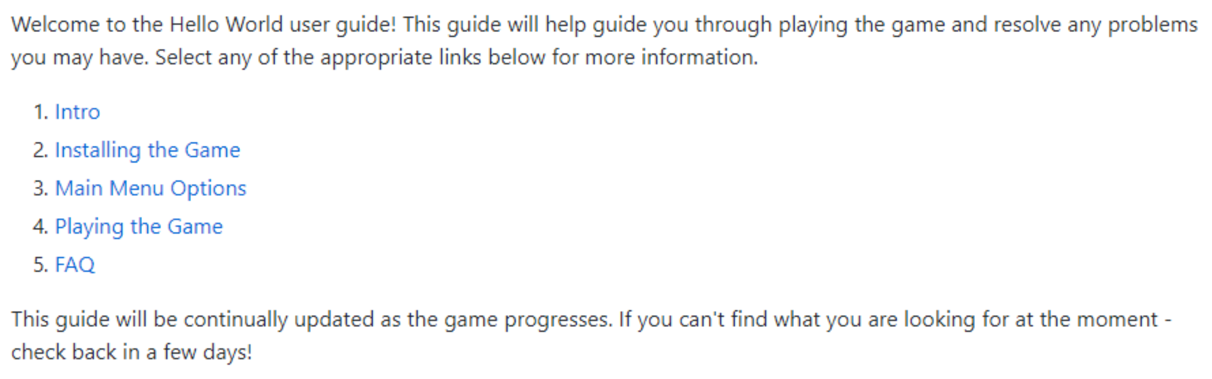
TC6: Game running on non-development system
TC6.2: The purpose of this test is to ensure that the game can successfully run on a non-development system through the executable created by dev system.
TC6.3: An executable of the game will be created and distributed to a non-development system. The game will then be run on this non-development system and tested.
TC6.4: The inputs for this test are the executable being run on the non-development system and all user inputs required to play the game.
TC6.5: The game should run smoothly and have no issues running on a non-development system.
TC6.6: Normal
TC6.7: Blackbox
TC6.8: Performance
TC6.9 Integration
Results: The game successfully runs on a non-development system. There are some issues with the lighting on non-development systems. This is a not a game breaking issue but the game does look slightly different on non-development systems.

User Manual

The user manual for Hello World is hosted on GitHub. A link to the manual is shown below:

https://github.uc.edu/tuckermr/HelloWorld/blob/master/User%20Guide/User%20Guide.md

Screenshot of user manual homepage:



Welcome to the Hello World user guide! This guide will help guide you through playing the game and resolve any problems you may have. Select any of the appropriate links below for more information.

1. Intro
2. Installing the Game
3. Main Menu Options
4. Playing the Game
5. FAQ

This guide will be continually updated as the game progresses. If you can't find what you are looking for at the moment - check back in a few days!

Frequently Asked Questions

What is this game about?

This game is about teaching basic programming concepts. You will walk around in the game and manipulate objects that are in the environment. In order to manipulate these objects - you will use your programming skills learned throughout the game.

I have no programming experience - can I play?

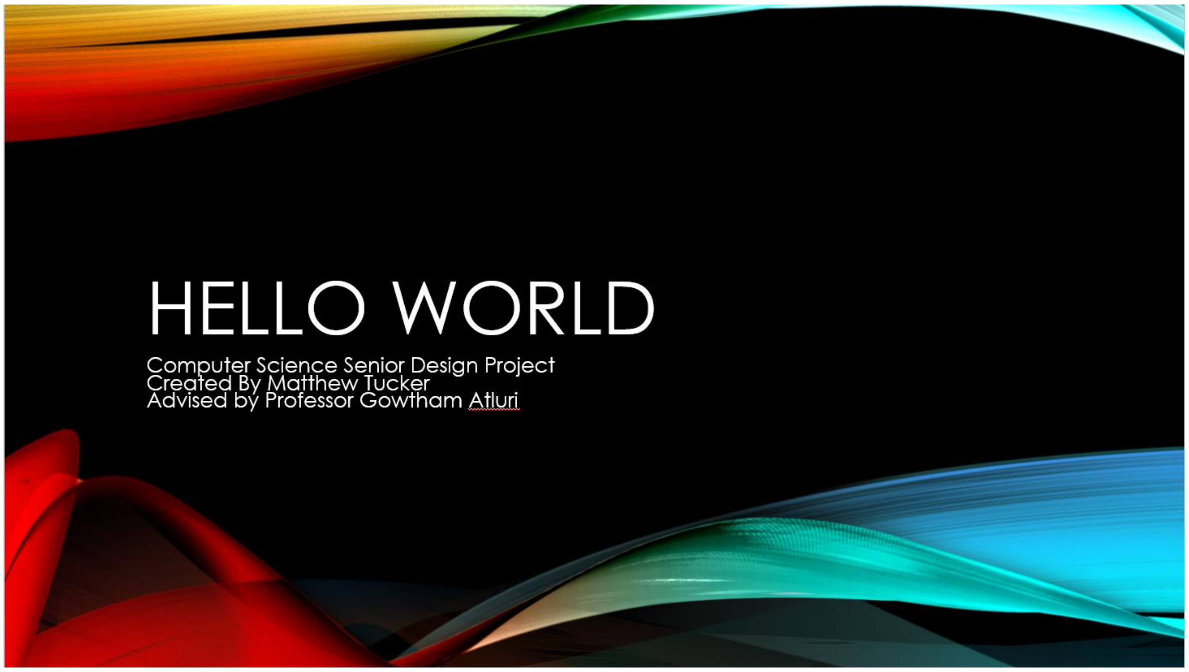
Yes! This game is designed to teach anyone - even those who have no experience programming.

Who is this game designed for?

This game is aimed toward anyone who wants to learn programming. The concepts are very basic and are therefore aimed at younger students - but anyone who wants to learn a little is welcome to play!

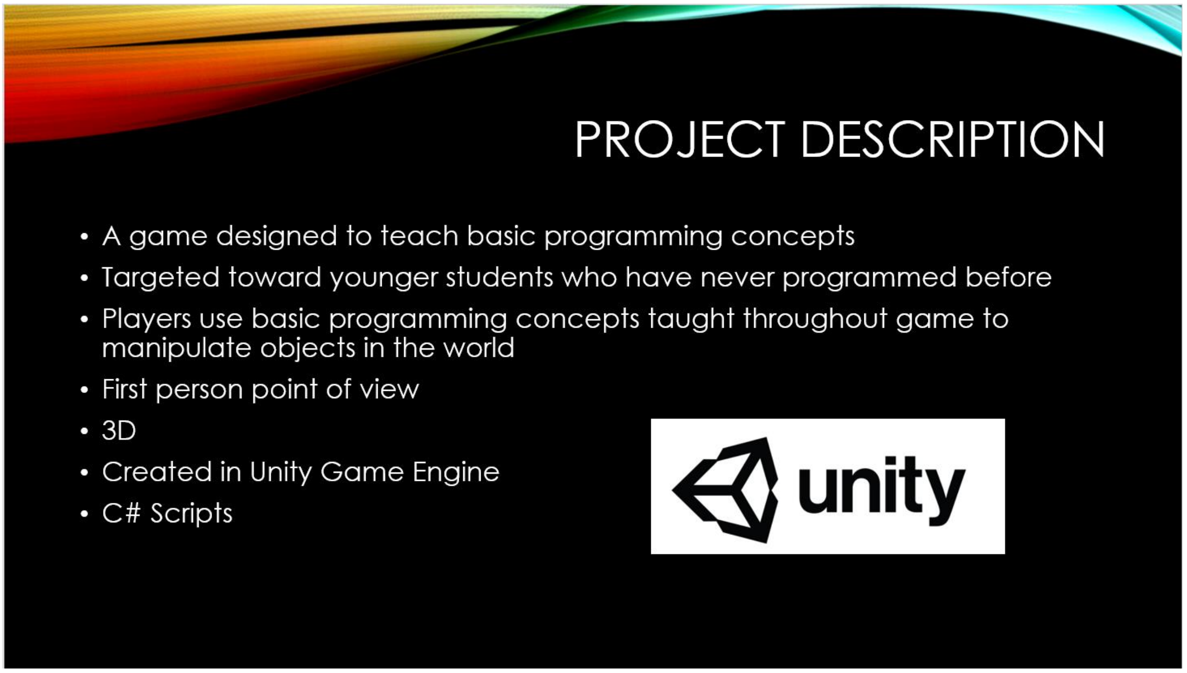
Spring Final Oral PowerPoint Presentation

My final oral PowerPoint presentation is shown below. The presentation was given on March 21, 2018.



HELLO WORLD

Computer Science Senior Design Project
Created By Matthew Tucker
Advised by Professor Gowtham Atluri



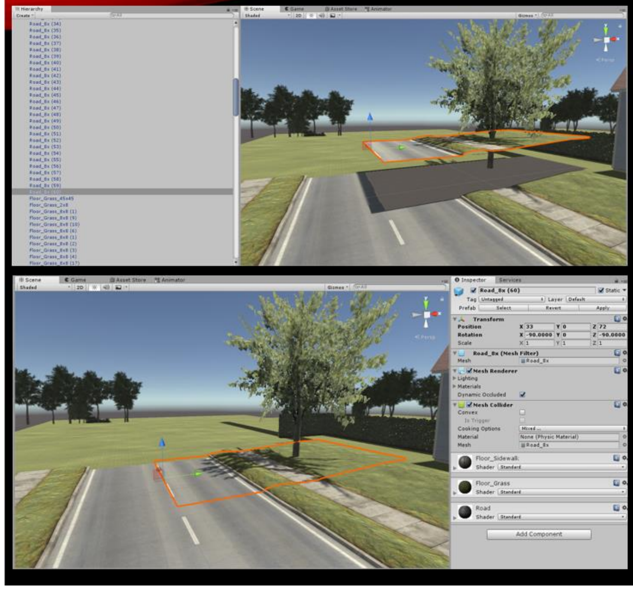
PROJECT DESCRIPTION

- A game designed to teach basic programming concepts
- Targeted toward younger students who have never programmed before
- Players use basic programming concepts taught throughout game to manipulate objects in the world
- First person point of view
- 3D
- Created in Unity Game Engine
- C# Scripts

PROGRAMMING CONCEPTS

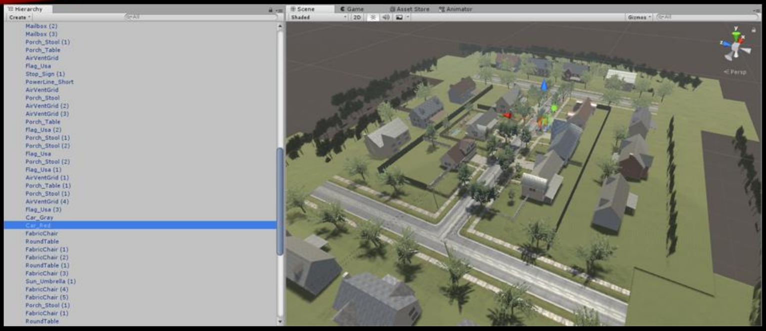
- Focuses on the ideas of programming rather than the specific syntax
- Variables
- Conditional Statements
- Functions
- Loops

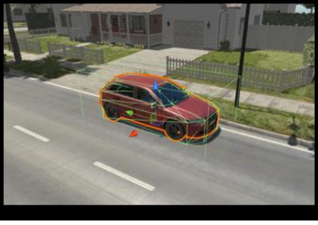
BUILDING THE WORLD IN UNITY



- Each object in the hierarchy was added and placed somewhere specific in the environment
- Objects placed using editor and Transform component
- Eye-opening experience building an environment from scratch

WORLD OVERVIEW



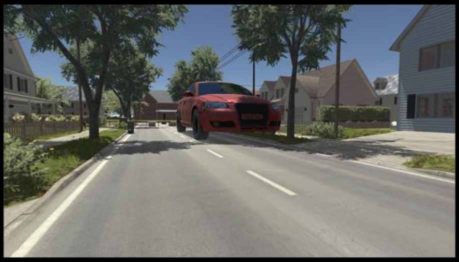
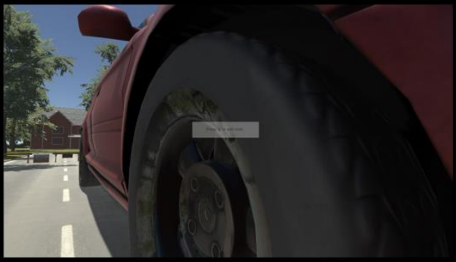
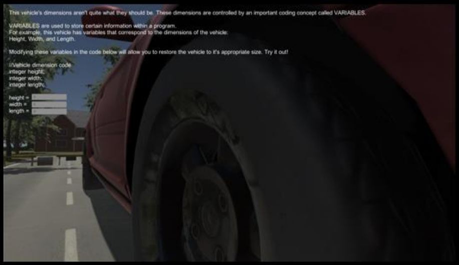


- Zoomed out perspective of world (players would never see this)
- Individual objects can have components attached to them to make them behave appropriately
- Attach C# scripts

FIRST PERSON PERSPECTIVE

- Examples of player perspective
- WASD and mouse movement

VARIABLE EXAMPLE

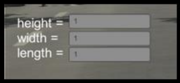



This vehicle's dimensions aren't quite what they should be. These dimensions are controlled by an important coding concept called VARIABLES.

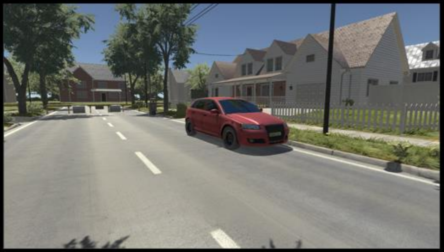
VARIABLES are used to store certain information within a program.
For example, this vehicle has variables that correspond to the dimensions of the vehicle:
Height, Width, and Length.

Modifying these variables in the code below will allow you to restore the vehicle to it's appropriate size. Try it out!

```
//Vehicle dimension code
integer height;
integer width;
integer length;
```

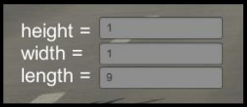
height =
width =
length =

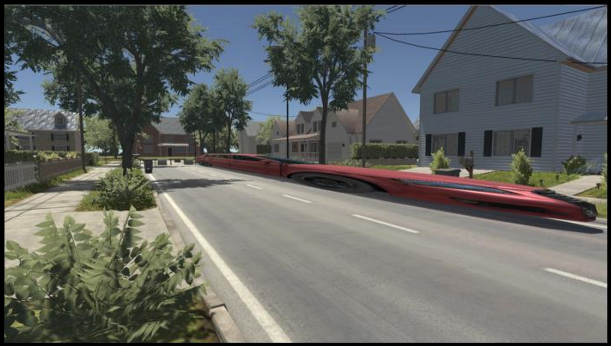

height = 1
width = 1
length = 1



EXPERIMENTATION

- Give players freedom to experiment
- See how their actions affect the objects in the environment
- More fun


height = 1
width = 1
length = 9

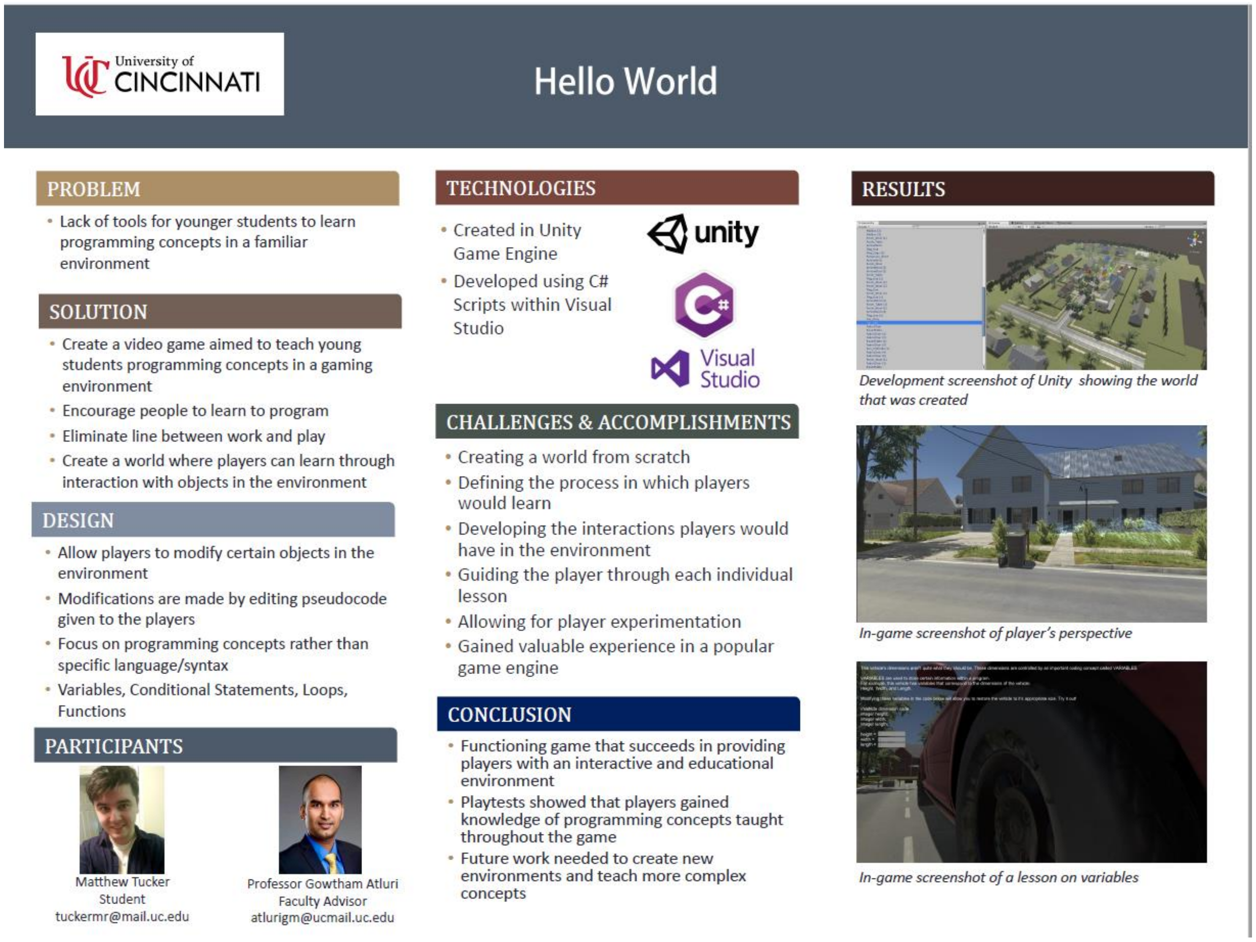


GOALS

- Players manipulate other objects in the world that teach them certain coding concepts
- Players experiment with these objects and see how their changes in the ‘code’ manipulate them
- Have a better understanding of coding concepts when formally taught
- Have some fun exploring the objects and the environment

Final Expo Poster

My final expo poster is shown below. The expo was held on April 3, 2018 and the project was presented to numerous students, professors and local professionals.



University of CINCINNATI

Hello World

PROBLEM

- Lack of tools for younger students to learn programming concepts in a familiar environment

SOLUTION

- Create a video game aimed to teach young students programming concepts in a gaming environment
- Encourage people to learn to program
- Eliminate line between work and play
- Create a world where players can learn through interaction with objects in the environment

DESIGN

- Allow players to modify certain objects in the environment
- Modifications are made by editing pseudocode given to the players
- Focus on programming concepts rather than specific language/syntax
- Variables, Conditional Statements, Loops, Functions

PARTICIPANTS

Matthew Tucker
Student
tuckermr@mail.uc.edu

Professor Gowtham Atluri
Faculty Advisor
atlurigm@ucmail.uc.edu

TECHNOLOGIES

- Created in Unity Game Engine
- Developed using C# Scripts within Visual Studio

unity

Visual Studio

CHALLENGES & ACCOMPLISHMENTS

- Creating a world from scratch
- Defining the process in which players would learn
- Developing the interactions players would have in the environment
- Guiding the player through each individual lesson
- Allowing for player experimentation
- Gained valuable experience in a popular game engine

CONCLUSION

- Functioning game that succeeds in providing players with an interactive and educational environment
- Playtests showed that players gained knowledge of programming concepts taught throughout the game
- Future work needed to create new environments and teach more complex concepts

RESULTS

Development screenshot of Unity showing the world that was created

In-game screenshot of player's perspective

In-game screenshot of a lesson on variables

Assessments

The following are self-assessments at two different times during the development process.

Initial Self-Assessment (fall semester)

The project I will be working on for Senior Design is called HelloWorld and the entire idea of this project is to introduce and teach programming concepts in video game form. There are many games that exist with this idea but most of them feel like studying rather than a game. Our goal is to create a game that is fun and feels like a video game that also teaches programming concepts. Ideally this will encourage the player to continue learning as they progress through the game. The game will be structured to target mid-teenagers and young adults but ideally will be enjoyed by anyone who likes video games and wants to learn to program. Ideally after playing the game, players will understand basic programming concepts and be able to apply these concepts in the real world.

I started my college career with almost zero programming experience. Throughout the years, I took many courses that taught me the basics of programming. Starting with Computer Science 1 (CS1021C) and Computer Science 2 (CS1022C), where I learned the very basics of programming including: variables, functions, loops, pointers, classes, object oriented programming, etc. These concepts will be taught throughout HelloWorld. I have always thought that a good way to demonstrate/test your knowledge on a subject was to try and teach someone else that subject. Another important course for this project is Software Engineering (EECE 3093C) because it taught a lot about objects and inheritance which is another very important topic we will need to cover in HelloWorld. Data Structures (CS2028) and D&A Algorithms (CS 4071) are two other courses that taught more specific programming concepts that we will need to cover some of in HelloWorld.

When I started my first co-op experience, I knew a little about programming (C++) but had never actually coded in the real world. This first experience was somewhat eye-opening in that I realized how little I knew. I spent my entire co-op career at a company called Signalysis and my job title was software developer. At Signalysis, I was taught so much programming and I specifically spent a lot of time learning about pointers and inheritance and how important they are when you are coding in C++. Again, we will strive to cover these aspects within HelloWorld and to teach them effectively we will need to understand them ourselves. One other important concept that I learned while at Signalysis was working with a team to create a project. I will be working with three other students on this project and we will need to share code in an effective way to succeed.

My main reason for choosing Computer Science as my major was because I played video games all the time growing up and continue to do so today. I was always fascinated with how they were made and it always seemed like magic to me. I chose Computer Science so I could understand video games and learn how they were made. This project is a perfect opportunity to test my knowledge of all the computer science topics that go into creating a video game from scratch. I am excited to start from nothing and create a game that people can enjoy and learn from. I believe it will be challenging but will provide invaluable experience in the game development industry.

The first thing we need to do to make this project a reality is figure out what our game will be like and how it will go about educating its players. We want our game to have an interesting storyline that will encourage the players to want to go further in the game and learn as much as possible. Designing this story will be challenging because it needs to be interesting enough for the player but also incorporate learning programming concepts. My expectation is to have a finished game by the end of the year. I hope to have a demo station set up at the expo that can play the game and allow others to see it and enjoy it. I will know I have succeeded if the game successfully tells an engaging story while also educating people on programming concepts. Hopefully it will inspire some people who play it to pursue a career in computer science.

Final Self-Assessment (spring semester)

It is already the end of the semester and I have completed my senior design project. It is interesting to see how ambitious I was at the start of the semester when reading my initial self-assessment. I learned so much about myself and my abilities throughout this project and it was a great experience for me. I started with a group of three other students but due to some personal issues and differing opinions we decided to split up so this ended up being a solo project. Working as a solo student on this project, this meant that the entire progress of the project relied on my work alone. Obviously, I believe being able to work with a team is an important skill, but I felt that working alone was a great experience in its own right. I had to focus on many different aspects of the project and be responsible to hit important deadlines and milestones. I also had to adjust certain aspects of the project that would have been too much to get done in the time given for one person. Being able to adapt when you hit certain roadblocks is another important skill that I felt I learned throughout this project.

The project I created was called Hello World and it was a simple video game that taught its players very basic programming concepts. The game focused on teaching four very common programming ideas: variables, conditional statements, functions, and loops. The way I accomplished this was by creating a world where players could manipulate objects by following tutorials laid out for them. The players would edit some very basic pseudocode that would change the environment they existed in. I created the majority of this game using a popular game engine called Unity and used C# scripts to write my code. This project was the first bit of experience I received in creating a video game and I learned a lot about the process. I learned how much work has to go into every part of a game that I never would have realized if not for this project. I faced many challenges: learning how to use Unity, designing the world the players would exist in, writing good code that would function properly, developing the tutorials in game. I was successful in creating a game that taught its players programming concepts. I was very proud of the world and how the players could change it. I am very excited to continue working on this project and also to start new projects in game development. I believe the experience I gained throughout this project will help me greatly if I ever decide to pursue game development as a career.

Summary of Hours and Justification

The fall semester consisted mostly of designing the game and figuring out exactly what I wanted to do. There was also a lot of work done in becoming familiar with the Unity game engine. I had never used Unity before and I wanted to learn it so I spent a lot of time figuring out how to use it. I also spent time thinking about what concepts would be possible to teach in the type of game I had in mind. I knew I wanted to make a 3D, first person game that put players in a familiar environment. The reason for this was I wanted people who didn't regularly play video games to understand the environment they were in.

The spring semester was where most of my work was done. I first began by creating the world the player would exist in and this was a major part of the project. Constructing a world from scratch is quite a time-consuming process. Once I had my world I was able to experiment and test out which programming concepts would be applicable in the environment. I then began writing the scripts to allow the player to interact with the objects, which was another major part of the project. This was the key mechanic in the game so I needed to make sure it worked properly and the process felt smooth for the player. Finally, I worked on testing and creating the menu screens for the project.

Fall Semester 2017 Total Hours: 105
Spring Semester 2018 Total Hours: 260

November - 45 hours

1. Developed concept for game
2. Gained experience working in Unity

December - 60 hours

1. Created list of possible programming concepts to be taught
2. Created rough draft for tutorials in game that would teach programming concepts
3. Created basic controls for player movement in game

January - 65 hours

1. Began creation of the world in Unity that players would exist in
2. Narrowed down list of programming concepts to be taught

February - 75 hours

1. Finalized creation of the world in Unity
2. Solidified list of programming concepts to be taught (Variables, conditional statements, functions, loops)
3. Created more specific rough drafts tutorials for each concept to be taught

March - 120 hours

1. Created UI for in game tutorials
2. Completed C# scripts that allowed players to interact with objects in the world
3. Created main menu/exit screens
4. Ran test plans and fixed issues

Final thoughts and GitHub Link

I learned so much throughout the process of creating this project. It has always been a dream of mine to create a video game on my own and this project gave me the drive to finally work towards this. There is still plenty of room left for improvement on my game and I plan to continue to improve it in the future. I gained so much experience in the game development world and if I ever do decide to pursue game development as a career I will be so glad to have this experience. You never realize how much work has to go into something until you try it yourself and creating a game is no exception.

GitHub link: https://github.uc.edu/tuckermr/HelloWorld