

Feasibility- “Programming Game”

CS 410 - Team Silver
Old Dominion University
October 19, 2017

Team Silver Mentor

Professor Thomas Kennedy



- Lecturer Professor, Old Dominion University
- M.S. in Computer Science, Old Dominion University
- Excels at responding promptly to emails, outlining course information, and knowing the most current Computer Science practices

Team Silver



Colten Everitt
Website Administrator



Peter Riley
Cyber Security Specialist



Casey Batten
Unity Game Expert



Kevin Santos
Database Manager



Joel Stokes
Game Development Specialist



Matthew Tuckson
Computer Science Expert



Daniel Dang
Web Development Expert



Nathaniel DeArce
Artificial Intelligence Agent



Tyler Johnson
Software Engineering Expert

Outline

1. Change Major/Dropout Rate for Computer Science at ODU
2. **Problem Statement**
3. What is Unity?
4. What is C#?
5. Unity Ease of Use and Developer Popularity
6. Unity Development Resources
7. The Advantages of C# within the Unity SDK
8. Why a Game?
9. Concepts of Gameplay and Possible Design Choices
10. Possible Influences on Using a Game
11. The Current Process Flow
12. **Plans for Our Solution**
13. Our Customers
14. Revised Process Flow
15. Major Components
16. **Competition Matrix**
17. References

Change Major/Dropout Rate for Computer Science at ODU

The Computer Science major at ODU is usually difficult and hard to grasp for most students. Thus, the rate of students either changing their major or dropping out of college altogether has increased in the past few years. (Additional info. . .).

In (year), approximately (# students) students changed their major from Computer Science to something else at ODU. (Cite source).

In (year), approximately (# students) Computer Science students dropped out of ODU altogether. (Cite source).

“Programming Game”

Our Problem Statement

Programming is intimidating for the uninitiated. As a result, first time ODU programming students drop out or switch majors. Existing tools fail to teach Object-Oriented Programming (OOP) concepts and problem solving skills.

What is Unity?

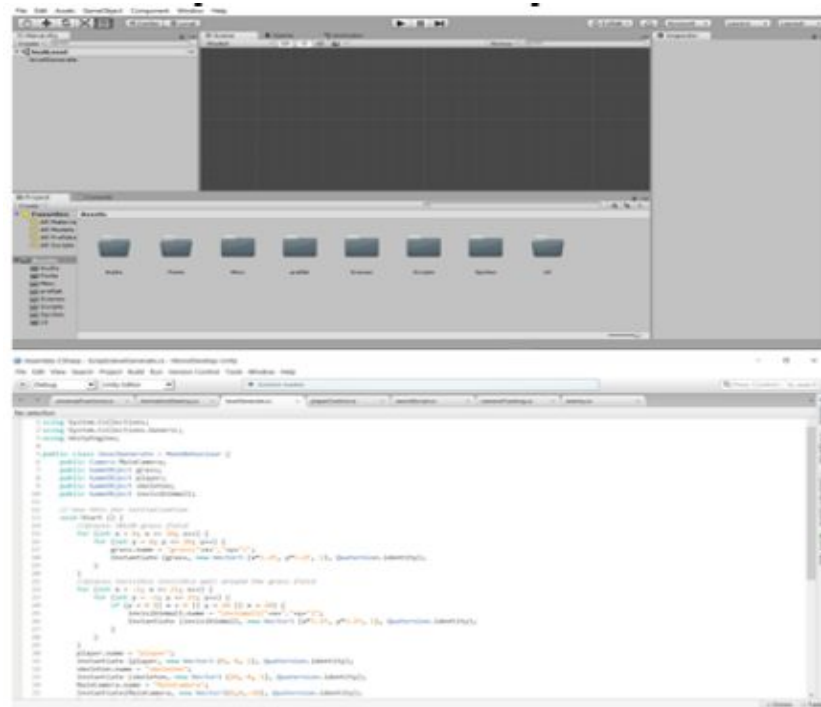
Unity is a cross-platform game engine that is developed by Unity Technologies. It is primarily used for video game development as well as simulations. Unity supports 2D as well as 3D graphics and uses C# as its default scripting language.

What is C#?

C# is a general purpose, object-oriented programming language created by Microsoft. Some of the design goals of the language were that it was supposed to be simple, portable, suitable for applications hosted and/or embedded and had strong support for internationalization.

Unity Ease of Use and Developer Popularity

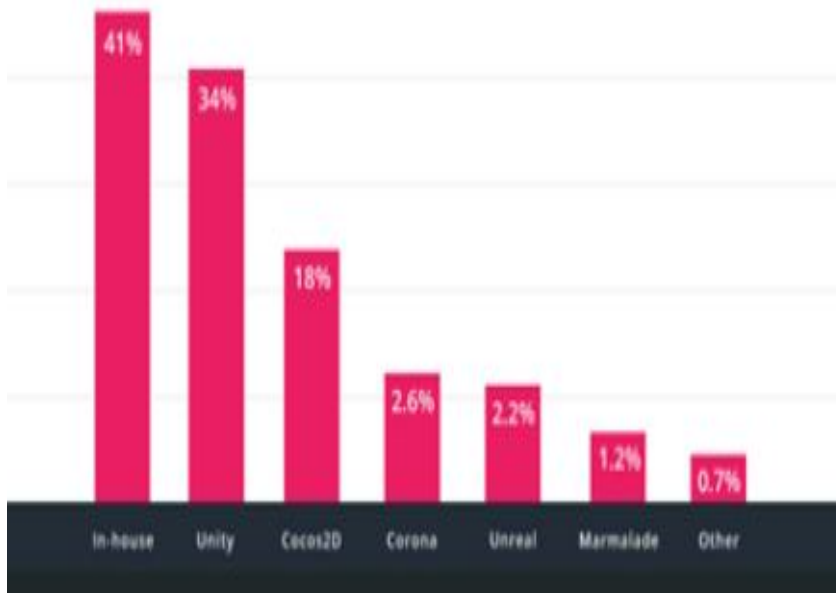
- Unity uses an incredibly flexible UI and developer workflow system that allows users to easily begin development on a product
- Options exist for both 2D and 3D development, porting to multiple platforms easily, and build settings for specific environments
- Unity has tools available for both game development and generalized software development, including the excellent MonoDevelop IDE



Unity Ease of Use and Developer Popularity, (Cont.)

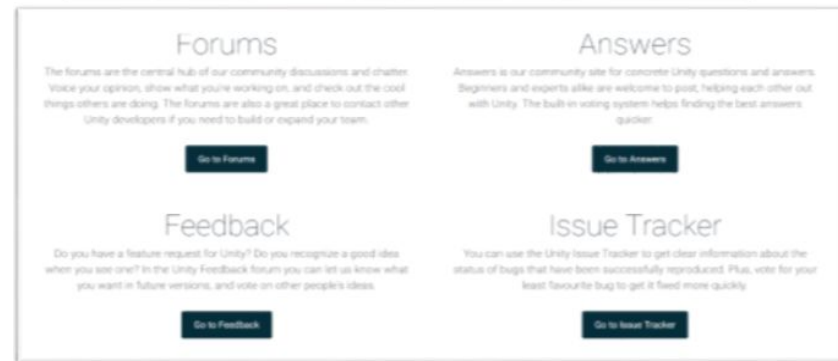
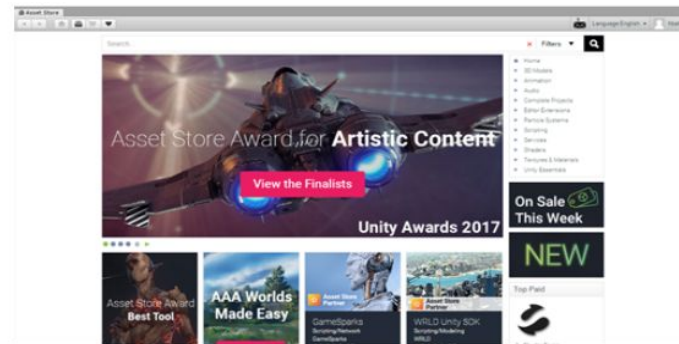
- Unity is extremely popular among both small independent developers and large corporations due to its flexibility and breadth of functionality
- Though hard numbers have yet to be recorded, according to Unity Technologies there were over 5 billion downloads of products made with Unity in quarter one of 2016, with an extra 2.4 billion in mobile product downloads

34% of top 1000 free mobile games are Made with Unity



Unity Development Resources

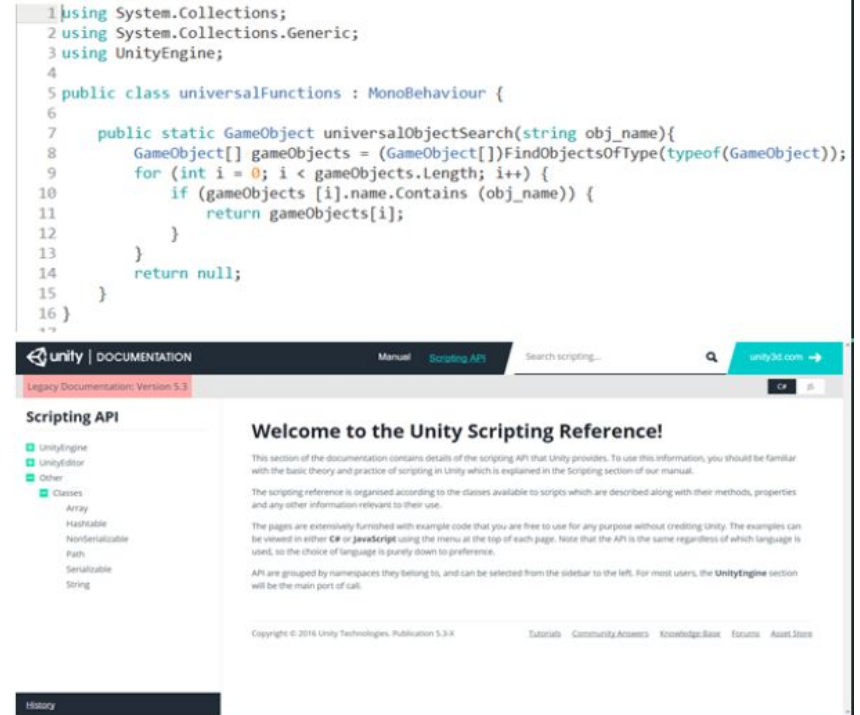
- One of the biggest advantages to the Unity SDK is its development resources
- Included with the Engine is access to the Asset Store, which hosts a catalog of free and paid services, such as 3D models, music, graphics packs, UI elements, sub-engine features such as specialized physics, and tutorial packs
- Also, there is the excellent Developer Support Community, which has various tutorial resources, live support, and a highly active developer forum where developers interact and offer advice and support to help ease the Unity development process



The Advantages of C# within the Unity SDK

- Out of the box, the Unity SDK supports C#, JavaScript, and Boo
- C# is by far the most flexible and powerful of these three, hosting both a large suite of built in functions/methods and having a series of Unity-specific tools available when used alongside the Engine
- Also, there is excellent developer support available through Unity's Scripting API website, which hosts code examples and complex breakdowns of Unity-specific functionality within C#

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class universalFunctions : MonoBehaviour {
6
7     public static GameObject universalObjectSearch(string obj_name){
8         GameObject[] gameObjects = (GameObject[])FindObjectsOfType(typeof(GameObject));
9         for (int i = 0; i < gameObjects.Length; i++) {
10             if (gameObjects [i].name.Contains (obj_name)) {
11                 return gameObjects[i];
12             }
13         }
14         return null;
15     }
16 }
```



Unity | DOCUMENTATION Manual Scripting API Search scripting... unity3d.com

Legacy Documentation: Version 5.3 C#

Scripting API

- UnityEngine
- UnityEditor
- Other
 - Classes
 - Array
 - Hashtable
 - NonSerializable
 - Path
 - Serializable
 - String

Welcome to the Unity Scripting Reference!

This section of the documentation contains details of the scripting API that Unity provides. To use this information, you should be familiar with the basic theory and practice of scripting in Unity which is explained in the Scripting section of our manual.

The scripting reference is organized according to the classes available to scripts which are described along with their methods, properties and any other information relevant to their use.

The pages are extensively furnished with example code that you are free to use for any purpose without crediting Unity. The examples can be viewed in either **C#** or **JavaScript** using the menu at the top of each page. Note that the API is the same regardless of which language is used, so the choice of language is purely down to preference.

APIs are grouped by namespaces they belong to, and can be selected from the sidebar to the left. For most users, the **UnityEngine** section will be the main port of call.

Copyright © 2014 Unity Technologies. Publication 5.3-4 Tutorials Community Answers Knowledge Base Forums Asset Store

History

Why a Game?

There are several reasons:

- Games can enhance interest among new learners
- New learners are typically teenagers in middle and high school
- A programming game can change the learning style from traditional to more dynamic
- Programming games do not require an instructor present at all times
- Concepts like Object-Oriented Programming and problem solving can possibly be better explained while playing a game



Concepts of Gameplay and Possible Design Choices

- A realistic approach: using relatable and applicable examples
- Improvement on teaching: more complex Object-Oriented Programming concepts can be easily explained
- Using a management simulator
- Move away from main-character oriented games that do not contribute to how programming is taught

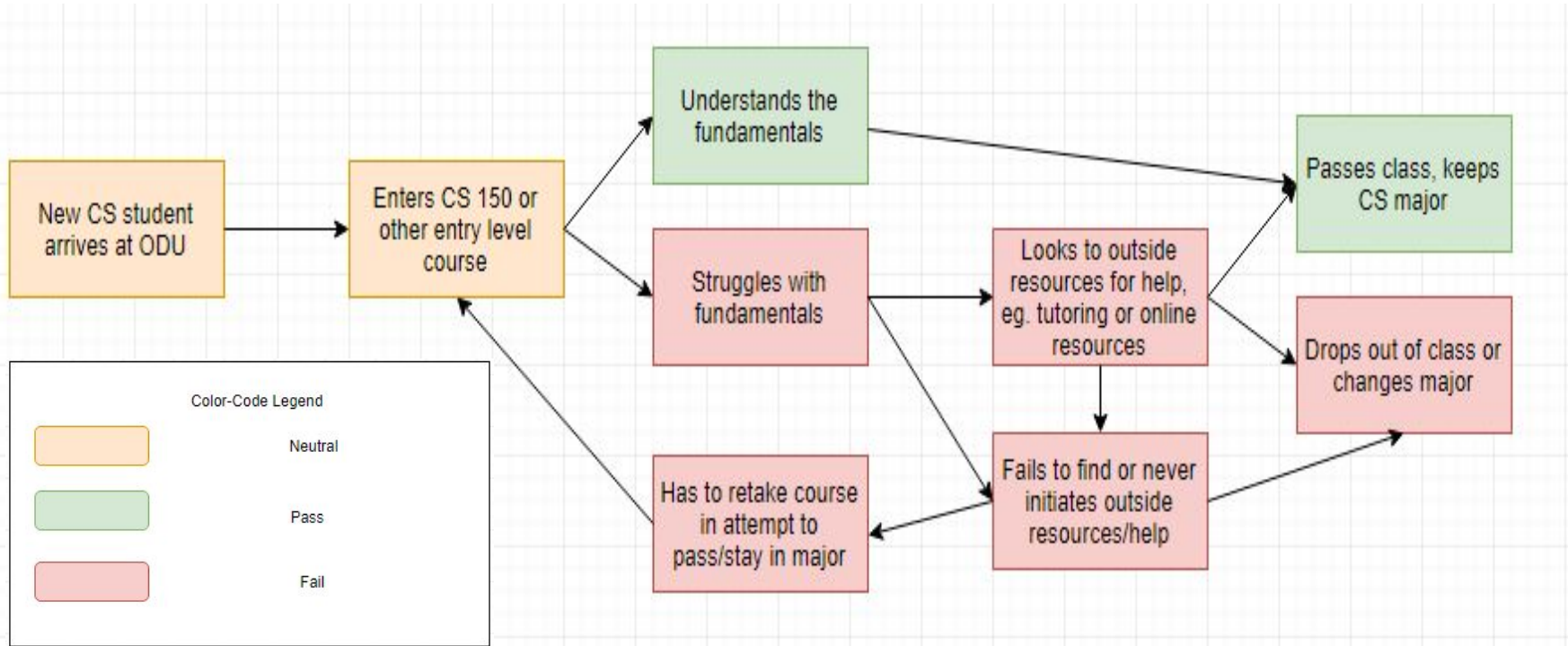


Possible Influences on Using a Game

- According to *computerscienceonline.com*, it is a pragmatic effort to teach youngsters to program before going to college
- Teaching youngsters programming while they are still in elementary school can embolden their coding interest early



Current Process Flow (Flawed)



“Programming Game”

Our Solution Statement

"Programming Game" will address Object-Oriented Programming (OOP) concepts and problem solving through the use of a management simulator and a Tangible User Interface (TUI).

Plans for Our Solution

Our solution will be a web-based game application which allows a user to become familiar with understanding the main core of computer programming, as well as understanding how beginner software is constructed.

- This application will teach Object-Oriented Programming (OOP) concepts
- This application will teach problem solving skills essential to understanding the foundation of computer programming
- This application will be able to teach multiple languages and be played on multiple platforms in later releases
- This application will allow users to interact with each other in a multiplayer setting in order to gain real-world collaborative skills

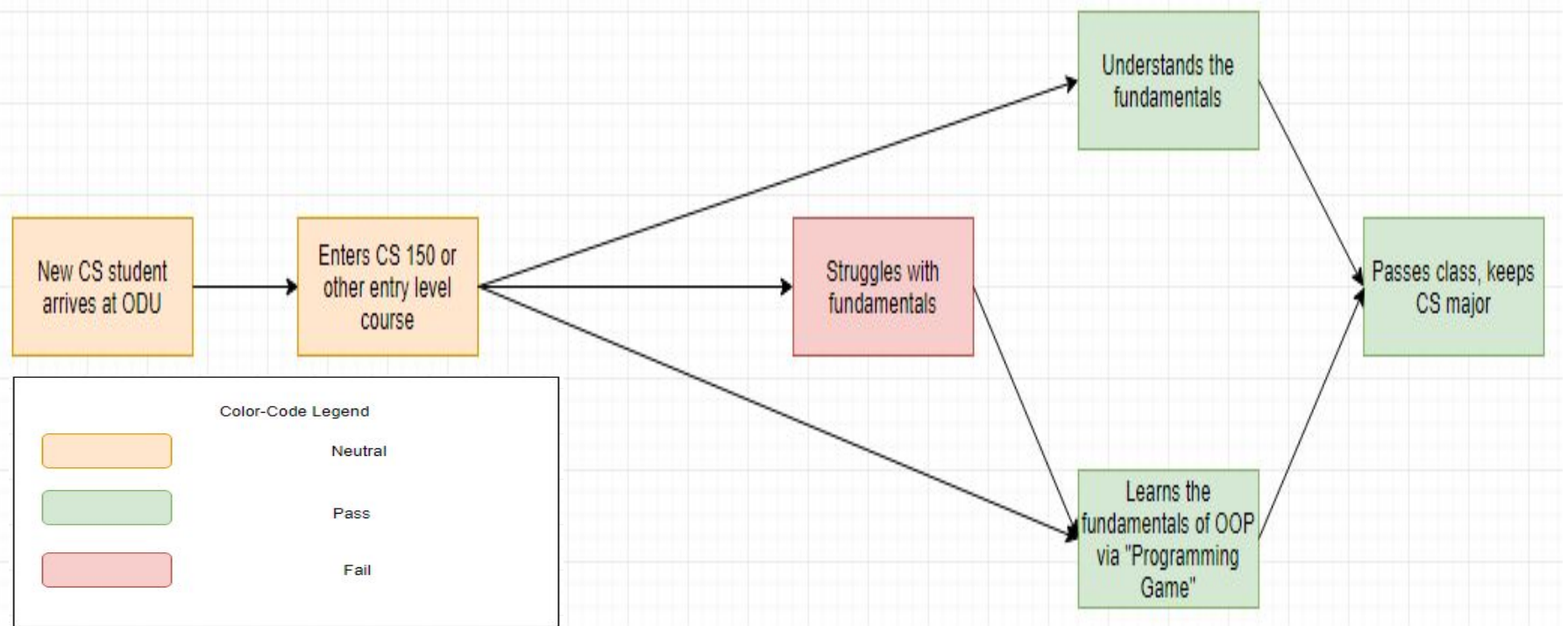
Target Customers

Our initial focus will be Old Dominion University, Universities, Colleges, and other Educational Institutes that currently have a Computer Science Degree Program. However, everyone could use this product in order to gain more knowledge on computer programming, especially to gain a good foundation in Object-Oriented Programming concepts and problem solving skills

End Users

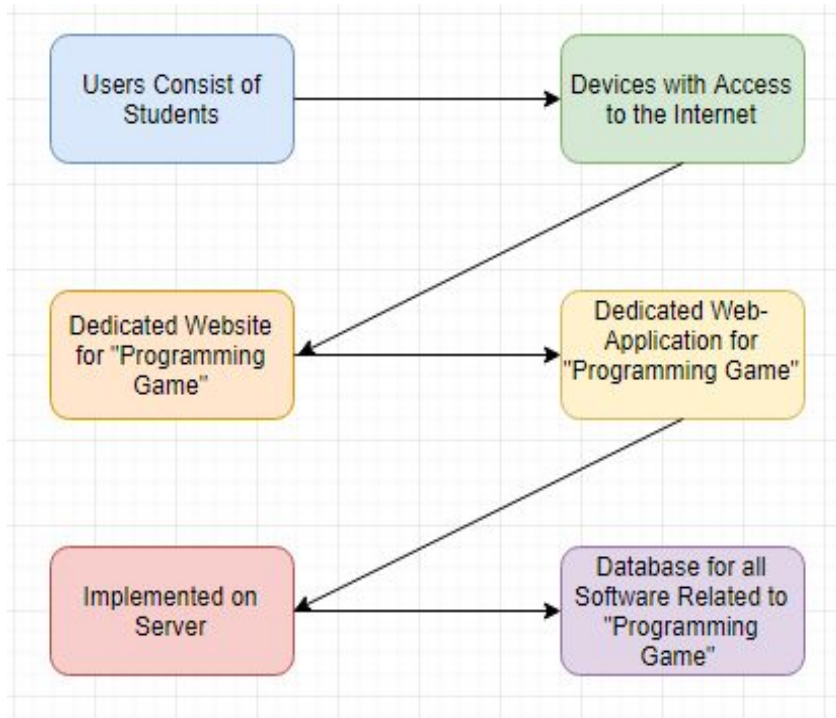
Students that are currently enrolled in a Computer Science Degree Program at Old Dominion University, Universities, Colleges, or other Educational Institutes

Current Process Flow (Revised)



Major Functional Components

- Users connect to the Internet using their preferred device
- Our dedicated web-application on our dedicated website will provide gameplay for users to learn Object-Oriented Programming concepts as well as problem solving skills



Structure of “Programming Game”



Competition Matrix

Game	Experience	Uses OOP	Teaches OOP	# Languages	Multiplayer
Codingame	Low-High	Yes	No	25+	Yes
Code Combat	Low	Yes	No	5	No
Screeps	Mid-High	Yes	No	1	Yes
CheckIO	Low-High	Yes	No	1	Yes
Code Monkey	Low	No	No	1	No
Elevator Saga	Mid-High	Yes	No	1	No
Codewars	Mid-High	Yes	Yes	6	Yes
Ruby Quiz	Mid-High	Yes	No	1	No

Competition Matrix, (Cont.)

Game	Experience	Uses OOP	Teaches OOP	# Languages	Multiplayer
Git Games	Low	No	No	1	No
CSS Diner	Low	No	No	1	No
Flexbox Defense	Low-Mid	No	No	1	No
Ruby Warrior	Low	No	No	1	No
Untrusted	Mid-High	No	No	1	No
Empire of Code	Low-Mid	Yes	No	2	Yes
Our Game	Low-Mid	Yes	Yes	1 or 2	Maybe

Conclusion

- Web-application game using the Unity SDK with C# and JavaScript to create
- User will learn Object-Oriented Programming (OOP) concepts and problem solving skills in depth
- These skills are essential to build a solid foundation for understanding Computer Science as well as computer programming
- These skills will allow students to pass introductory Computer Science classes, thus lowering the switching major/drop out rate for universities
- The current process that Computer Science students go through is flawed by not teaching these fundamental skills early on in the curriculum
- Our solution makes the process painless and fun

Questions?

References

- “Fast Facts.” *Unity*, Unity Technologies, unity3d.com/public-relations.
- *Asset Store*, Unity Technologies, www.assetstore.unity3d.com/en/.
- Technologies, Unity. “Welcome to the Unity Scripting Reference!” *Unity - Scripting API*: Unity Technologies, docs.unity3d.com/530/Documentation/ScriptReference/index.html.
- O'Neill, M. (n.d.). Computer Science Before College. Retrieved October 05, 2017, from <https://www.computerscienceonline.org/cs-programs-before-college/>
- Porter, J. (2017, August 03). 4 Benefits Of Learning Programming At A Young Age. Retrieved October 05, 2017, from <https://elearningindustry.com/4-benefits-learning-programming-at-a-young-age-2>
- Peter Riley’s presentation. It will be one of the main source of references for our project.