Feasibility- “Programming Game”

CS 410 - Team Silver
Old Dominion University
November 2, 2017

Team Silver Mentor

Professor Thomas Kennedy

- Lecturer Professor, Old Dominion University
- M.S. in Computer Science, Old Dominion University
- Excels at:
  - Working directly with students
  - Outlining course information
  - Conveying an exceptional level of proficiency in the most current Computer Science practices

Team Silver

Colten Everitt
Website Administrator

Peter Riley
Software Engineering Expert

Casey Batten
Unity SDK Specialist

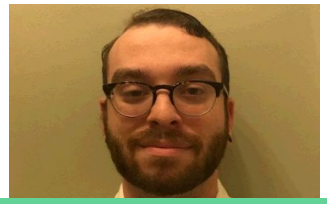
Kevin Santos
Database Manager

Joel Stokes
Game Development Specialist

Matthew Tuckson
Program/Project Manager

Daniel Dang
Web Development Expert

Nathaniel DeArce
Artificial Intelligence Agent



Tyler Johnson
Software Engineering Expert

Table of Contents

“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.

Student Progression Dilemma in Computer Science at ODU

- Students are not following the course series in the expected order
  - This data provides purpose behind the need to reassess methods of assisting students' progress in how they learn the topics at hand
  - When during the learning process this assistance would be most effective
- Changes in class volume, which could indicate that students are leaving the major for less intense fields
- Drop-off in student numbers from CS150 to CS250
  - Related to differing major requirements and course overlap, but the decrease in student body is significant enough to warrant a deeper look into later classes in the major path
- Decreasing class sizes show a steady decline in CS course enrollments as course level difficulty advances
  - Decreases may be indicative of students dropping out of the CS program or changing majors

Statistics to Support the Student Progression Dilemma

According to the **ODU Factbook**:

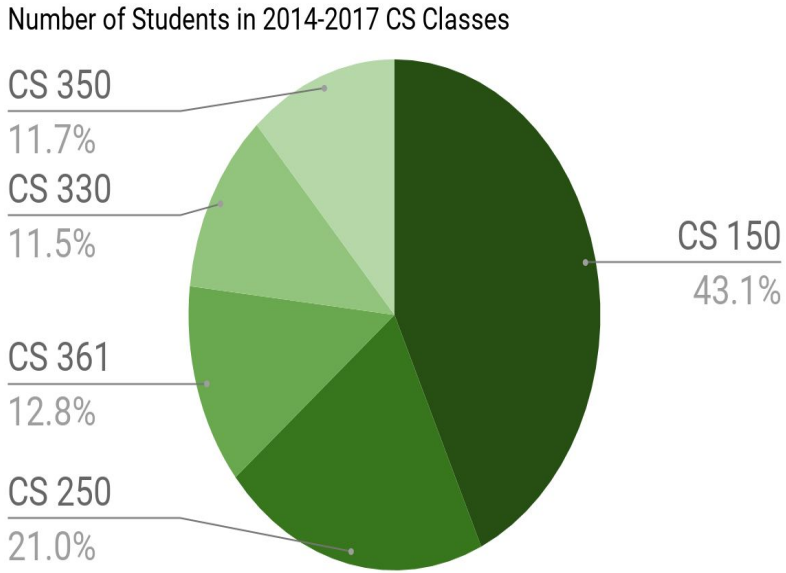
- 2012 - 2016:
  - Number of undergraduate CS majors increased from 284 to 429
    - Showing the high demand in the degree path
- 2014 - 2015:
  - Roughly 672 students enrolled in CS150
- 2015 - 2016:
  - Roughly 327 students enrolled in CS250
- 2016 - 2017:
  - Roughly 199 students enrolled in CS361
  - Roughly 180 students enrolled in CS330
  - Roughly 182 students enrolled in CS350

Computer Science Courses as Requirements

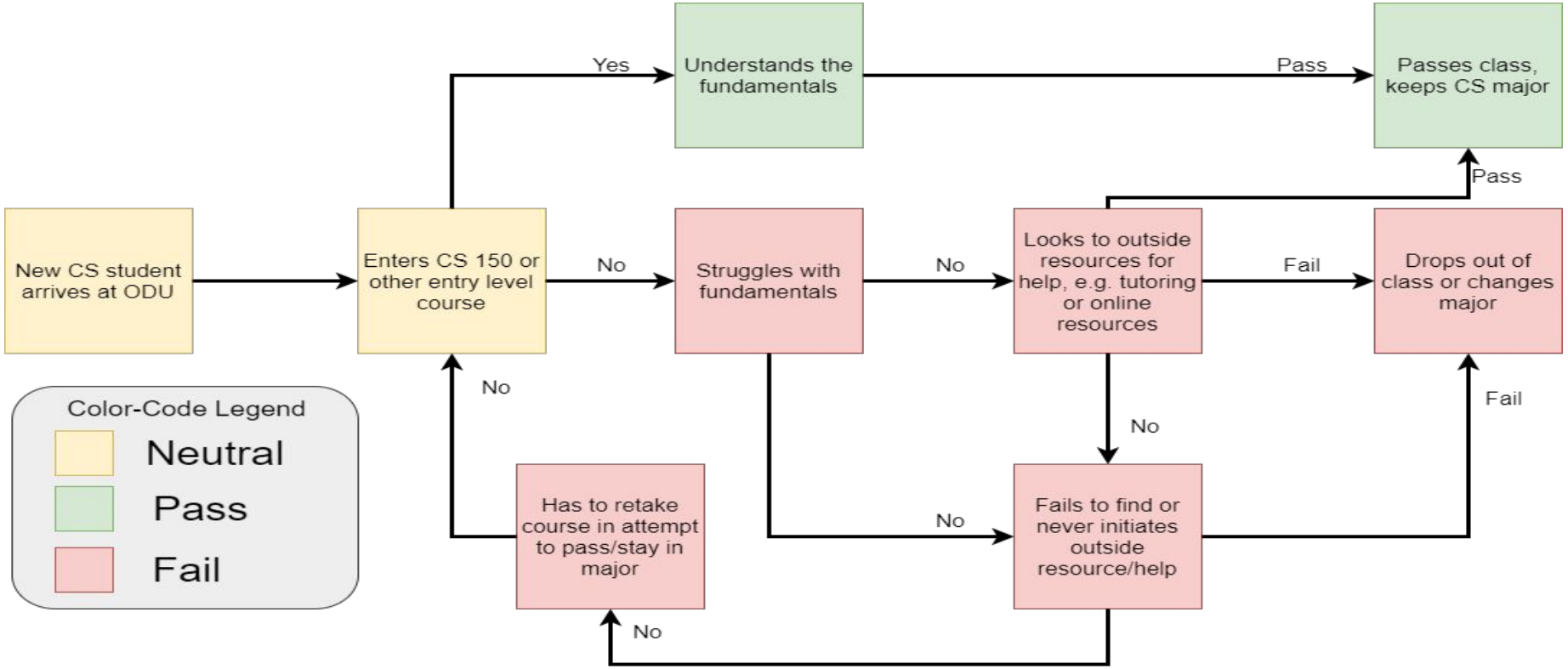
- CS150:
  - “Service Course”
  - Required to be taken by CS, Physics, Math, Engineering, & Mod-Simulation majors
- CS250:
  - Required to be taken by CS, Mod-Simulation, & Computer and Electrical Engineering majors
- CS330:
  - Required to be taken by CS & Mod-Simulation majors
- CS361:
  - Required to be taken by CS & Computer and Electrical Engineering majors
- CS350:
  - Required to be taken by CS & Computer Engineering majors

Student Progression Dilemma - Table & Graph

	CS 150	CS 250	CS 361	CS 330	CS 350
2013-2014	804	327	161	111	93
2014-2015	672	367	208	203	148
2015-2016	937	327	217	195	183
2016-2017	920	337	199	180	182



Current Process Flow



“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).

Target Customers

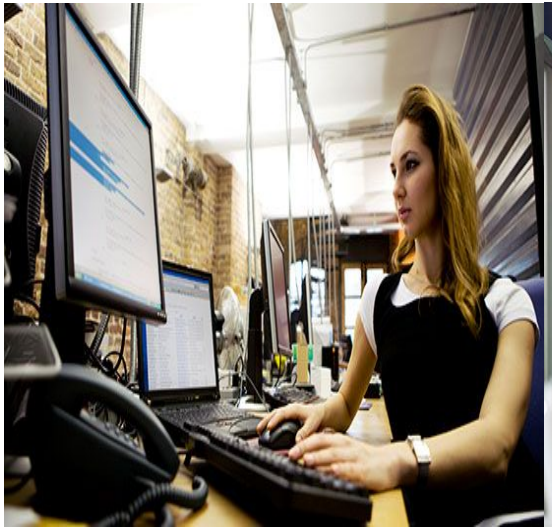
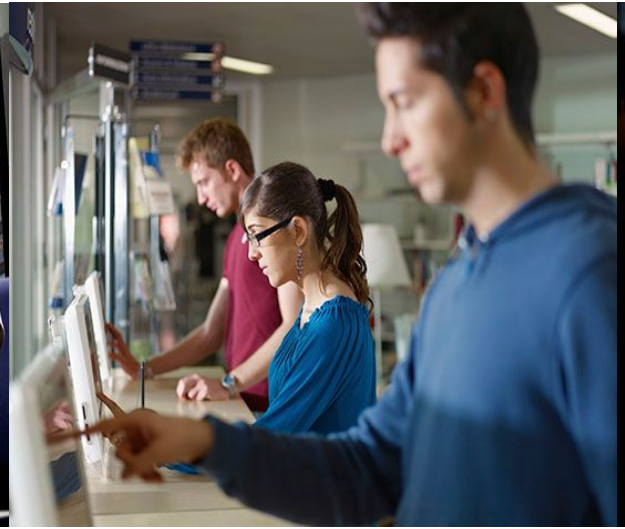
Initial focus will be Old Dominion University, as well as other universities, colleges, and educational institutions that currently offer a Computer Science degree program

Anyone could use this product in order to gain more knowledge in computer programming, Object-Oriented Programming concepts, and problem solving skills

Image Source : http://odu.edu/compsci

End Users

Students who are currently enrolled in a Computer Science degree program at Old Dominion University, or at other universities, colleges, or educational institutions


Left image source: https://online.odu.edu/programs/computer-science-ms
Middle image source: https://online.odu.edu/programs/computer-science
Right image: https://online.odu.edu/programs/computer-science-minor

Why a Game?

- Enhance interest among new learners
- Nature of interaction inherently gives players a more natural way to learn content
- Change the learning style from traditional to more dynamic
- Do not require an instructor present at all times
- Object-Oriented Programming and problem solving can potentially be better grasped and understood

Image 1 source : http://www.cdm.depaul.edu/academics/Pages/BS-in-Game-Programming.asx
Image 2 source: Peter Riley's presentation

Influences of Games on Learning

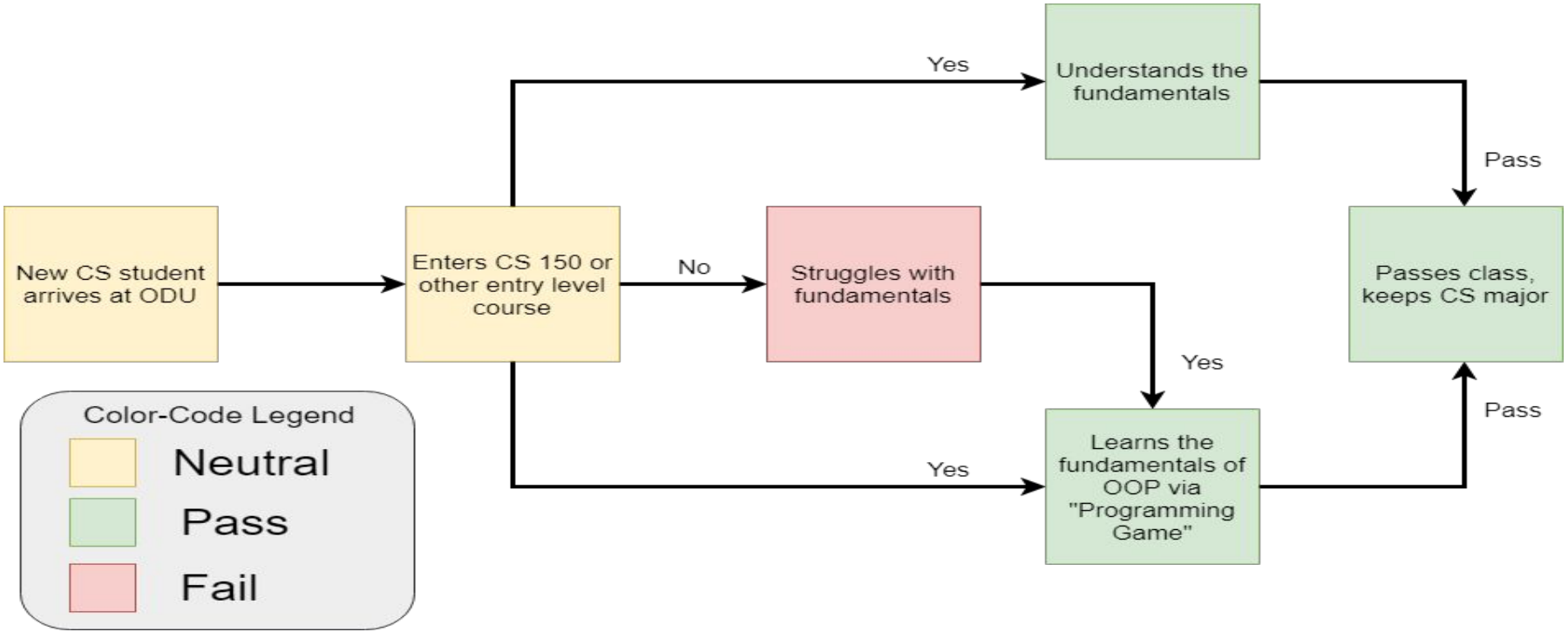
Poor academics and knowledge decrement lead to the stigma of video games being detrimental to the learning process. However, research evidence has shown that traditional learning through textbooks contributes to low engagement when compared to interactive media.

According to the **Office of Naval Research (ONR)** & **WebMD**:

- Video games have positive aspects that help people become more engaged in the learning process
- On average, **56 - 95%** of people who play a particular game to learn a particular subject through tests, demonstrated a better understanding of such subject
- Educational games with a solid foundation and interactive components keep the players engaged and promote enhanced concept learning

Stats Source:
The Benefits of Video Games. (2011, December 26). Retrieved October 19, 2017, from http://abcnews.go.com/blogs/technology/2011/12/the-benefits-of-video-games/

Solution Process Flow



Plans for our Solution

Solution: Game application that teaches users the fundamentals of computer programming and software development

This application will:

- Teach Object-Oriented Programming (OOP) concepts
- Teach problem solving skills
- Strive to teach multiple languages
- Be developed for multiple platforms
- Potentially have multiplayer gameplay to connect players

Structure of “Programming Game”

Image source: https://medium.mybridge.co/12-free-resources-learn-to-code-while-playing-games-f7333043de11

Rapid Prototype GUI Sample - Joel Stokes

Rapid Prototype GUI Sample - Casey Batten

Concepts of Gameplay and Possible Design Choices

- Realistic Approach:
  - Using relatable and applicable examples
- Improvement on Teaching:
  - More complex Object-Oriented Programming concepts that can be easily explained
- Balance Gameplay and Programming:
  - Implementation of the gameplay will not sacrifice the player's experience

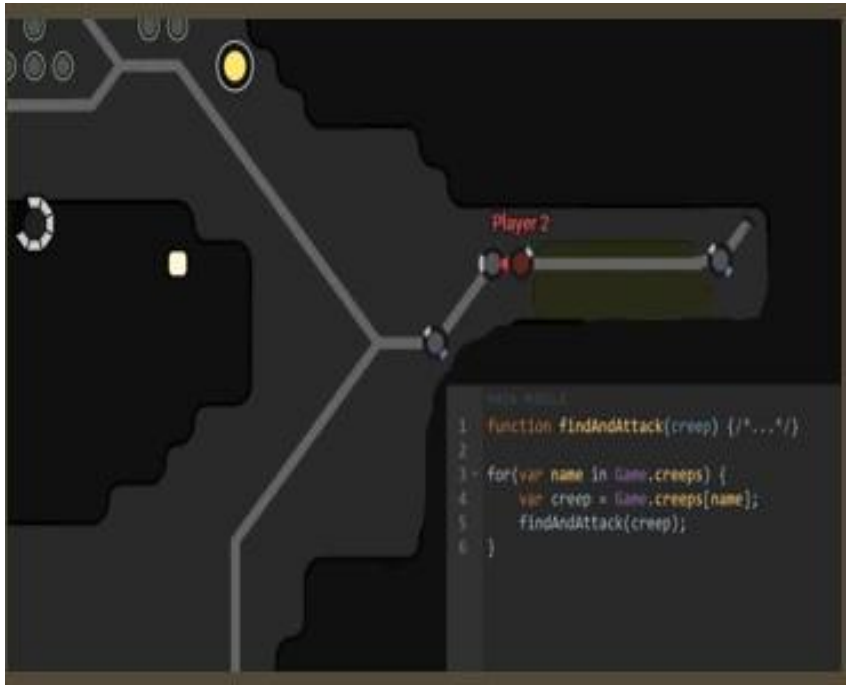


Image source 1: https://medium.com/@techloop.io/funniest-programming-memes-1da50c5229d

What is Unity?

Image source: https://commons.wikimedia.org/wiki/File:Unity_Technologies_logo.svg

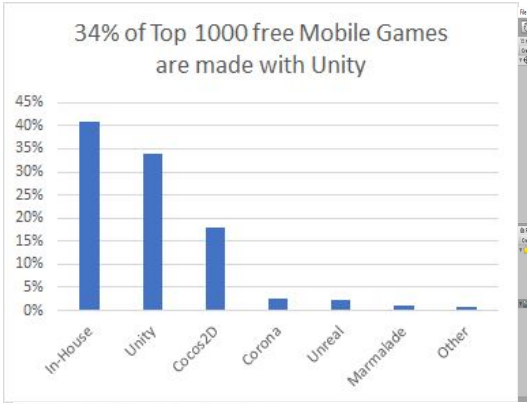
- Cross-platform game engine developed by Unity Technologies
- Primarily used for video game development as well as simulations
- Supports 2D, as well as 3D, graphics and uses C# as its default scripting language
- Creates games by manipulating game objects in 3D, and by attaching varying components to those objects
- Game engine is truly cross-platform:
  - Games can be exported to several different platforms:
    - Desktop, mobile, web, and even game consoles such as Xbox One, PS4

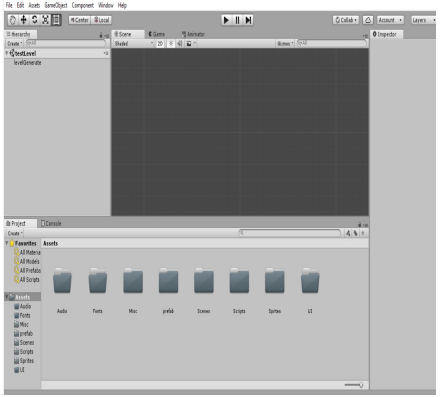
What is C#?

- General purpose, object-oriented programming language created by Microsoft in 2000
- Language design goals:
  - Simple, portable, and suitable for applications hosted and/or embedded, with strong support for internalization
- Enterprise language much like Java
- Programming language for Microsoft's web application framework ASP.NET
- Used to create command-line applications and desktop GUI based applications

Unity Ease of Use and Developer Popularity

- Flexible UI and developer workflow system
  - Allows users to develop a product efficiently
- Tools for software development:
  - MonoDevelop IDE and support for multiple platforms and build environments
- 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





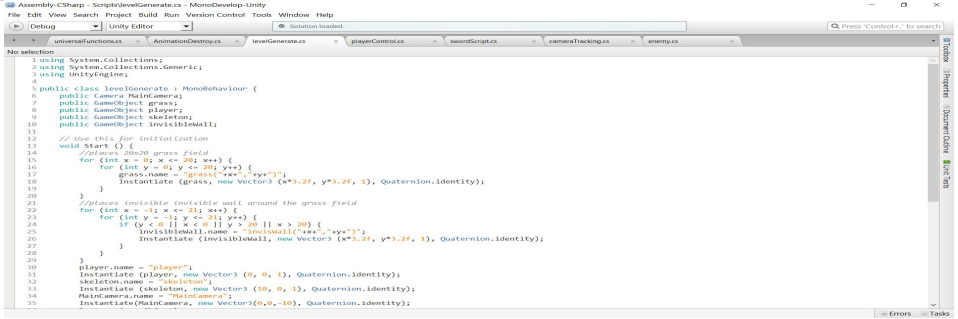


Image Source: https://unity3d.com/

The Advantages of C# within the Unity SDK

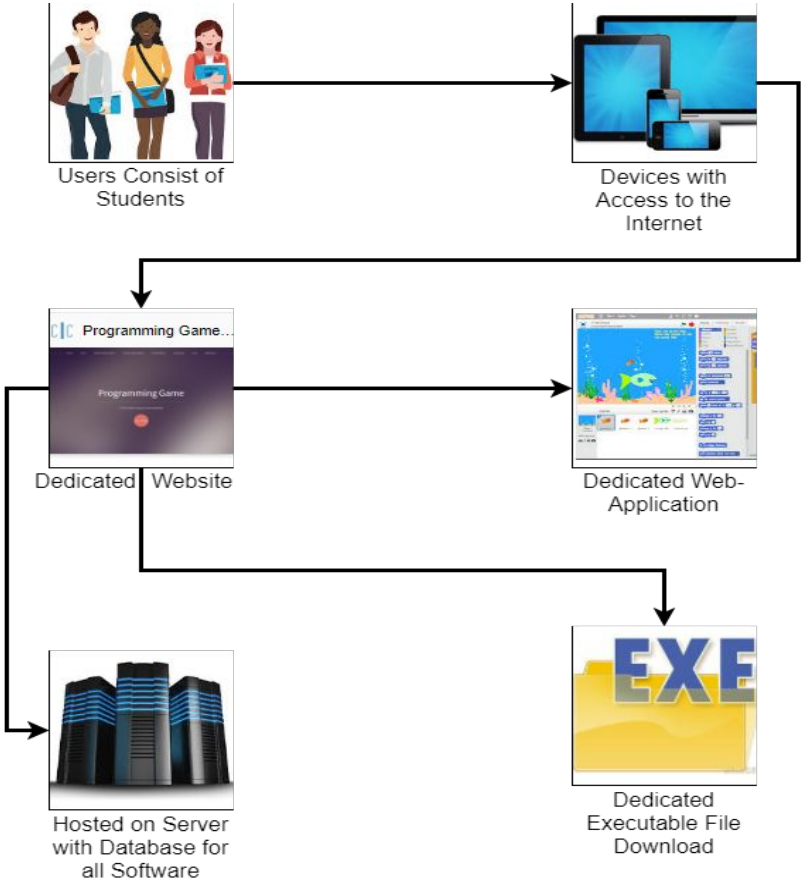
- Most flexible and powerful of the Out-of-the-Box programming languages Unity Supports
  - Large suite of built in functions/methods
  - Unity-specific tools available when used alongside the Engine
- Developer support available through Unity's Scripting API website
  - Code examples
  - Complex breakdowns of Unity-specific functionality within C#

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class universalFunctions : MonoBehaviour {

    public static GameObject universalObjectSearch(string obj_name){
        GameObject[] gameObjects = (GameObject[])FindObjectsOfType(typeof(GameObject));
        for (int i = 0; i < gameObjects.Length; i++) {
            if (gameObjects [i].name.Contains (obj_name)) {
                return gameObjects[i];
            }
        }
        return null;
    }
}
```

Image source: https://docs.unity3d.com/Manual/index.html?_ga=2.37814243.1199564661.1509584726-986472080.1506709735

Major Functional Components

- Users connect to the Internet using their preferred device
- Our game will teach users Object-Oriented Programming concepts as well as problem solving skills



Competition Matrix Part 1

Game	Experience	Uses OOP	Teaches OOP	# Languages	Multiplayer
Our Game	Low-Mid	Yes	Yes	1 or 2	Maybe
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
Codingame	Low-High	Yes	No	25+	Yes

Competition Matrix Part 2

Game	Experience	Uses OOP	Teaches OOP	# Languages	Multiplayer
Our Game	Low-Mid	Yes	Yes	1 or 2	Maybe
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
Ruby Quiz	Mid-High	Yes	No	1	No

Competition vs. Our Solution

Competition Trends:

- Low programming experience
- Focus on one or two languages
- Mainly teaches syntax, rather than OOP
- Some include multiplayer

Our Solution:

- Low programming experience
- Start with one or two languages, but allow for teaching more in the future
- Focus on OOP concepts, with syntax being a secondary objective
- Multiplayer will depend on what gameplay features are implemented

Conclusion

- CS students not introduced to OOP or problem solving skills early on
- Skills are essential to build a solid foundation for understanding CS, including programming
- Web-application or executable downloadable game using the Unity SDK with C# and JavaScript
- User learns Object-Oriented Programming (OOP) concepts and problem solving skills in depth
- Skills allow end user to become more proficient in Computer Science, as well as Object Oriented Design

Our Solution Makes the Process Painless & Fun

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/
- Peter Riley’s presentation. It will be one of the main source of references for our project.
- “kennedyData.” Thomas Kennedy, https://drive.google.com/drive/u/1/folders/0B_xCQd8Vk2BnSU1hNnJwSXB1NEE
- “The Benefits of Video Games.” *abcnews* (2011, December 26). Retrieved October 19, 2017, from http://abcnews.go.com/blogs/technology/2011/12/the-benefits-of-video-games/ Good Morning America
- CS410 Programming Game Pitch, By: Joel Stokes - https://youtu.be/QBvgzFgZaOQ
- CS410 Project Dungeon Demo, By: Casey Batten - https://www.youtube.com/watch?v=ynhdd1lKgps
- CS410 Dungeon Escape Demo (Short Ver.), By: Casey Batten - https://www.youtube.com/watch?v=VnHRaWl8Y8w