

Lab 1 - PolyMorpher Product Description

Joel Stokes

Old Dominion University

CS411W

Professor Thomas Kennedy

29 January 2018

Version 1

Author Note

Joel Stokes, Department of Computer Science, Old Dominion University.

This research was done under the supervision and guidance of Thomas Kennedy.

Correspondence concerning this article should be addressed to Joel Stokes, Department of Computer Science, Old Dominion University, Norfolk, VA 23529.

Contact: [jstok018@odu.edu](mailto:jstok018@odu.edu)

Table of Contents

<b>1. Introduction</b>	<b>3</b>
1.1 Team Members	4
<b>2. Product Description</b>	<b>4</b>
2.1 Product Key Features and Capabilities	5
2.2 Major Functional Components	6
<b>3. Identification of Case Study</b>	<b>6</b>
3.1 Current Process Flow	7
3.2 Solution Process Flow	7
<b>4. Product Prototype Description</b>	<b>7</b>
4.1 Prototype Functional Objective	8
4.2 Prototype Architecture	8
4.3 Prototype Features and Capabilities	9
4.4 Prototype Development Challenges	10
<b>5. Prototype Demonstration Description</b>	<b>11</b>
<b>6. Glossary</b>	<b>12</b>
<b>7. References</b>	<b>14</b>
<b>8. Figures and Tables</b>	<b>18</b>

## 1 Introduction

For many first-time students, entering the world of programming can be a daunting task. Requiring a radically different thought process than most degree programs, getting on the right track to understanding fundamentals, such as Object-Oriented Programming, may not come quickly for many students. This faulty fundamental understanding can cause students to fail programming courses, and even drop out of the major altogether.

*(See Student Progression Dilemma Table & Graph, pg 22)*

On a separate note, a powerful learning medium exists today which is vastly-underutilized: video games. Interactive mediums such as games require a higher level of physical and mental participation than traditional learning styles, which have been proven to engage the brain for easier learning. Enticing visuals with a fun atmosphere can “trick” people into learning concepts that may not interest them in traditional classroom or textbook settings. Also, games can allow difficult concepts to be learned faster when done in engaging styles.

This is where PolyMorpher comes in: an educational programming game designed to teach students programming problem solving and the concepts of Object-Oriented Programming. This product, PolyMorpher, takes the powerful and immersive medium of video games into the educational programming field of OOP. Once in consumers hands, PolyMorpher will help bridge the educational gap in CS Major’s programming skills in an addictive, learning environment.

## 1.1 Team Members

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

## 2 Product Description

PolyMorpher is an educational video game designed to help Computer Science students learn the fundamentals and principles of Object-Oriented Programming. The game allows the player to modify the code of objects in the world, and forces them to take on multiple puzzles which require different programming approaches to progress. Along the way, the player is guided with information and tips to help conquer the challenges, and in the process, learn techniques relevant to the strategies of Object-Oriented Programming.

## 2.1 Key Product Features and Capabilities

PolyMorpher will be a game in which the player has the ability to click objects to directly edit the code inside them. The player will be required to change object properties, transform location coordinates, manipulate hitboxes, and change code that will affect multiple objects in tandem. This style will teach the user Object-Oriented Programming skills while also subconsciously teaching problem solving skills as well as how to think in the mindset of a programmer. Depending on the skill level of the user, PolyMorpher may allow them to skip introductory or certain beginner puzzles in order for them to spend more time learning information closer to their knowledge level.

The primary feature which sets the product apart from its competition is the focus on teaching Object-Oriented Programming. The majority of educational programming games focus on the basics, such as syntax or learning a specific language. What sets PolyMorpher apart is the focus on coding techniques and principles, such as encapsulation and polymorphism, which can be carried into many different languages. The product isn't designed to be a beginner's introduction to programming, but a student's continuation into more advanced programming topics.

*(See Competition Matrix, pg. 21)*

## 2.2 Major Functional Components (Hardware/Software)

The product itself is entirely software, and requires the user to provide their own hardware. Users can use the product by either connecting to the internet with their preferred device to play online, or by downloading the software as an executable file to run on their own machine. The product will be able to run in a default keyboard/mouse style on Windows, Mac, and Linux devices, with an alternative touch-screen style for Android and iPhone users.

*(See Major Functional Components Diagram, pg 26)*

The software is being developed in C# using Unity, a free cross-platform SDK. Along with this, GitLab will be used to manage version control within the team. Although the majority of art and sound effects will be coming through asset-purchased or public-domain sources, GIMP and Audacity will be used as free art and music creation tools.

*(See Version Control Flow Diagram, pg 24)*

## 3 Identification of Case Study

As stated in the introduction, the product idea originated from seeing Computer Science students at Old Dominion University struggle and drop out from programming courses, most likely due to poor understanding with the concepts and curriculum. Initially, the product will be focused primarily at Old Dominion University students, as well as other colleges and educational institutions that provide Computer Science courses. As a broader consumer base, anyone interested in programming could use this product.

*(See Student Progression Diagram, pg 22)*

### **3.1 Current Process Flow**

This Process Flow follows a new Computer Science student attending a university. As they enter into CS150 or higher for the first time, it is common for the student to struggle learning the fundamentals. Here, they can either seek outside assistance (which may not always work), or simply fail the class and, potentially, drop out of the major.

*(See Current Process Flow, pg 21)*

### **3.2 Solution Process Flow**

This Process Flow follows the same new Computer Science student, however they now have access to our product: PolyMorpher. Here, if the student has problems learning the fundamentals, they can turn to the product to learn them in an engaging and unique format. Also, students who already understand the fundamentals can bolster their knowledge with the product.

*(See Solution Process Flow, pg 22)*

## **4 Product Prototype Description**

PolyMorph's prototype will focus on displaying a working demonstration of the game. Due to time constraints, the product itself will be shortened in length, however it will contain at least one example of every Object-Oriented Programming technique that will be covered in the official product. The gameplay itself will be very similar to what is seen in the final product.

#### **4.1 Prototype Functional Objective**

The most important addition to the prototype is a fully-functioning code manipulator. This system allows the player to edit the scripts of objects in the game in real time, letting their code come to life. Limits and regulations will be set to prevent the user from implementing broken or potentially dangerous code.

Also, a physics handler for the player controller will allow fluid gameplay, and a gravity system to allow physics-based puzzles for specific OOP challenges. Additional objects will be coded and created to populate the puzzles and give them their Object-Oriented solutions.

*(See API Algorithm Diagrams, pgs 19, 20, 23, 24)*

#### **4.2 Prototype Architecture**

Although the demonstration provided a good example to visualize what the product will do, it left a lot to the imagination, and forced consumers to guess how well the product will educate them. Having the prototype will allow us to put the game into testers hands and genuinely prove the product's educational merit. Although it may be easy to make broad claims about the product's educational value, it's impossible to back up without a genuine, working demo.

*(See Work Breakdown Structure Designs, pgs 19, 20)*

Once the prototype is complete, it will be easy to get user feedback on risks such as "disliking the UI" and "users getting stuck". If they end up being problems, they can easily be addressed when moving towards the final product.

*(See Risk Matrices, pg 18)*



### **4.3 Prototype Features and Capabilities**

Our current scope will lead us to a fully-functional prototype featuring four distinct levels. First is a tutorial, focusing on teaching the player the basics of the code-editing mechanic, as well as basic programming puzzles to ensure understanding before moving the player to more advanced tasks. The other three levels branch into progressively more difficult Object-Oriented Programming techniques, and will aid users with supplemental text information if needed.

Another major feature included in the Prototype will be the introduction of official art and sound. This will bring the prototype leaps and bounds ahead of the visual presentation of the demonstration, and will help entice potential customers to try the product.

*(See Demonstration Art Style & Prototype Art Style Diagrams, pg 25)*

#### 4.4 Prototype Development Challenges

With the team's current development strategy, the product is being designed in thirds by three separate teams. Although certain constants will be in place, such as player scripts and physics, the process of combining the segments into one cohesive game could become difficult. However, with the current team size, three individual smaller teams to create the educational puzzles makes more logistical sense rather than one large group all working on the same exact part of the game simultaneously.

Due to the wide variety of platforms the product can be run on, a lot of time will need to be spent towards cross-platform testing. The primary platforms to be focused are Windows, Linux, MacOS, Android, and iPhones. Although Unity's SDK provides many cross-platform development tools, the product's interface will need to be freely operated with either a mouse and keyboard or touchscreen.

The largest development challenge comes with video games as an art form. PolyMorpher development requires its own art, music, and sound to give the product its own feel and theme. This being said, a two-dimensional art style will be used to remove modeling and lighting necessities, alleviating extra art difficulty.

The most important part of education through games is to immerse the player within the world of the product rather than making the education feel like a forced chore. Tied in with this, a large amount of planning is required to keep the product on the border between entertaining and educational. If the product is not entertaining enough, it could lose the interest of the customers, causing them to no longer use the product. On the other hand, if the product is not educational enough, it fails its purpose as a product.

### **5 Prototype Demonstration Description**

On completion of the prototype, willing Old Dominion University students and faculty in the Computer Science department will test the game in order to receive constructive criticisms in the field. This feedback will help us determine if the product is successfully completing its educational goal in an entertaining fashion, or if tweaks need to be made to increase the product's usefulness. Also, the prototype will be officially demonstrated at the end of the Spring 2018 semester during the CS411 Prototype Demo in April. If feedback is considered positive enough by our team, we will officially move towards completing the finished product.

## Glossary

**API:** Application Program Interface

**Git:** version control system for tracking changes in computer files and coordinating work on those files among multiple people.

**GitLab:** web-based git repository manager the includes wiki and issue tracking.

**Gradle:** an open-source build automation system that was designed for multi-project builds.

**GUI:** Graphical User Interface

**JavaScript:** a programming language commonly used in web development where the the code is processed by the client's browser.

**MySQL:** an open source multi-user database management system.

**Non-Technical Game:** user-friendly gameplay able to be utilized by non-technical users.

**Non-Technical User:** user who lacks formal education or knowledge in computer science, computer programming, object-oriented programming, or problem solving skills.

**ODU:** Abbreviation for Old Dominion University.

**Platform:** an integrated set of packaged and custom applications tied together with middleware.

**Regression Testing:** a type of application testing that determines if modifications to the application have altered the application negatively.

**Student Involvement:** the amount of physical energy students exert and the amount of psychological energy they put into their college experience.

**TUI:** Tangible User Interface

**Ubuntu:** open-source Linux operating system.

**User-Friendly:** easy to comprehend by non-technical users.

**Virtual Machines:** an emulation of a computer system that provide functionality of a physical computer.

**Web Application:** a client-server computer program in which the client (including the user interface and client-side logic) runs in a web browser.

**Wiki:** a website on which users collaboratively modify content and structure directly from the web browser.

## References

Batten, C. (Narrator). (2017). CS410 Dungeon Escape Demo (Short Version) [Online video].

Online: YouTube. Retrieved from <https://www.youtube.com/watch?v=ynhdd1IKgps>

Batten, C. (Narrator). (2017). CS410 Project Dungeon Demo [Online video]. Online: YouTube.

Retrieved from <https://www.youtube.com/watch?v=ynhdd1IKgps>

Batten, C. (2017, November 21). CS410 Tech Demo 2 (Download Source Code). In

PolyMorpher. Retrieved from <http://www.cs.odu.edu/~410silver/references.html>

Batten, C. (2017, November 29). VersionControlFlow. In draw.io. Retrieved December 21,

2017, from [https://www.draw.io/?state=%7B%22ids%22:%5B%221IQj6SYJqC6YLAK\\_qMRVIQkHiUmr9laBu%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G1IQj6SYJqC6YLAK\\_qMRVIQkHiUmr9laBu](https://www.draw.io/?state=%7B%22ids%22:%5B%221IQj6SYJqC6YLAK_qMRVIQkHiUmr9laBu%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G1IQj6SYJqC6YLAK_qMRVIQkHiUmr9laBu)

Batten, C. (2017, October 26). CS410 Dungeon Escape Demo (Download Source Code). In

PolyMorpher. Retrieved from <http://www.cs.odu.edu/~410silver/references.html>

Batten, C. (2017, October 26). CS410 Dungeon Escape Demo (Play Now). In PolyMorpher.

Retrieved from <http://www.cs.odu.edu/~410silver/references.html>

Edraw. (2017, May 12). Standard Flowchart Symbols and Their Usage. In Edraw Visualization

Solutions. Retrieved from <https://www.edrawsoft.com/flowchart-symbols.php>

Everitt, C. (2017, September 6). Current Process Flow. In draw.io. Retrieved December 21,

2017, from <https://www.draw.io/?state=%7B%22ids%22:%5B%220B-5KdQEdqLUPdnBFUnp2V05uMEE%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G0B-5KdQEdqLUPdnBFUnp2V05uMEE>

Everitt, C., & Dang, D. (2017, September 24). currentProcessFlow. In draw.io. Retrieved

December 21, 2017, from

<https://www.draw.io/?state=%7B%22ids%22:%5B%220B3Bc9>

[5zBWXg9TFZ6X0FMU1NTdEk%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G0B3Bc95zBWXg9TFZ6X0FMU1NTdEk](https://www.draw.io/?state=%7B%22ids%22:%5B%220B3Bc95zBWXg9TFZ6X0FMU1NTdEk%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G0B3Bc95zBWXg9TFZ6X0FMU1NTdEk)

Everitt, C., Santos, K. & DeArce, N. (2017, November 27). Work Breakdown Structure (WBS).

In draw.io. Retrieved December 21, 2017, from

[https://www.draw.io/?state=%7B%22ids%22:%5B%](https://www.draw.io/?state=%7B%22ids%22:%5B%220B-)

[220B-](https://www.draw.io/?state=%7B%22ids%22:%5B%220B-)

[5KdQEdqLUPWnNoSHhIUGg2OTQ%22%5D,%22action%22:%22open%22,%22userId](https://www.draw.io/?state=%7B%22ids%22:%5B%220B-5KdQEdqLUPWnNoSHhIUGg2OTQ%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G0B-)

[%22:%22108692003133590583047%22%7D#G0B-](https://www.draw.io/?state=%7B%22ids%22:%5B%220B-5KdQEdqLUPWnNoSHhIUGg2OTQ)

[5KdQEdqLUPWnNoSHhIUGg2OTQ](https://www.draw.io/?state=%7B%22ids%22:%5B%220B-5KdQEdqLUPWnNoSHhIUGg2OTQ)

Everitt, C., Santos, K. & DeArce, N. (2017, October 13). ProcessFlowDiagram\_silver. In

draw.io. Retrieved December 21, 2017, from

<https://www.draw.io/?state=%7B%22ids%22:%5B%220B>

[\\_xBnZ1ge4PIZTVjV3h6Y2pGSWc%22%5D,%22action%22:%22open%22,%22userId%](https://www.draw.io/?state=%7B%22ids%22:%5B%220B_xBnZ1ge4PIZTVjV3h6Y2pGSWc%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G0B_xBnZ1ge4PIZTVjV3h6Y2pGSWc)

[22:%22108692003133590583047%22%7D#G0B\\_xBnZ1ge4PIZTVjV3h6Y2pGSWc](https://www.draw.io/?state=%7B%22ids%22:%5B%220B_xBnZ1ge4PIZTVjV3h6Y2pGSWc)

Few, S. (2008, February 5). Practical Rules for Using Color in Charts. In Perceptual Edge.

Retrieved from [http://www.perceptualedge.com/articles/visual\\_business\\_intelligence/](http://www.perceptualedge.com/articles/visual_business_intelligence/)

[Rules\\_for\\_using\\_color.pdf](http://www.perceptualedge.com/articles/visual_business_intelligence/Rules_for_using_color.pdf)

Kennedy, T. (2017, September 6). kennedyData. In Google Drive. Retrieved from

[https://drive.google.com/drive/u/1/folders/0B\\_xCQd8Vk2BnSU1hNnJwSXB1NEE](https://drive.google.com/drive/u/1/folders/0B_xCQd8Vk2BnSU1hNnJwSXB1NEE)

O'Neill, M. (2017, March 6). Computer Science Before College. In Computer Science Online.

Retrieved from <https://www.computerscienceonline.org/cs-programs-before-college/>

Riley, P. (2017, September 14). Using Games to Introduce Programming to Students

[PowerPoint slides]. Retrieved from <http://www.cs.odu.edu/~410silver/references.html>

Stokes, J. (Narrator). (2017). CS410 Programming Game Pitch [Online video]. Online:

YouTube. Retrieved from

<https://www.youtube.com/watch?v=QBvgzFgZaOQ&feature=youtu.be>

Stokes, J. (2017, October 9). CS410 Programming Game Pitch (Download Source Code). In

PolyMorpher. Retrieved from <http://www.cs.odu.edu/~410silver/references.html>

“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

Santos, K., Riley, P. & Dang, D.(2017. December 7) Risk matrix and description tables in

Design Presentation. Retrieved from

[https://docs.google.com/presentation/d/1oY9lkSAHvg2OIRkljYJNZWCqVTbiw45STKgIsJUQjJI/edit#slide=id.g283e74317a\\_0\\_177](https://docs.google.com/presentation/d/1oY9lkSAHvg2OIRkljYJNZWCqVTbiw45STKgIsJUQjJI/edit#slide=id.g283e74317a_0_177)

Unity Technologies. (2017, August 10). Company Facts. In Unity. Retrieved from

<https://unity3d.com/public-relations>

Unity. (2016, July 6). Unity - Scripting API. In Unity. Retrieved December 21, 2017, from

<https://docs.unity3d.com/530/Documentation/ScriptReference/index.html>

Unity. (2017, October 11). Asset Store. In Unity. Retrieved December 21, 2017, from

<https://www.assetstore.unity3d.com/en/>



12 Free Games to Learn Programming. (2016, April 25). In Mybridge. Retrieved from <https://medium.mybridge.co/12-free-resources-learn-to-code-while-playing-games-f7333043de11>

#### Feasibility, Design, & Prototype PowerPoint Presentation - References

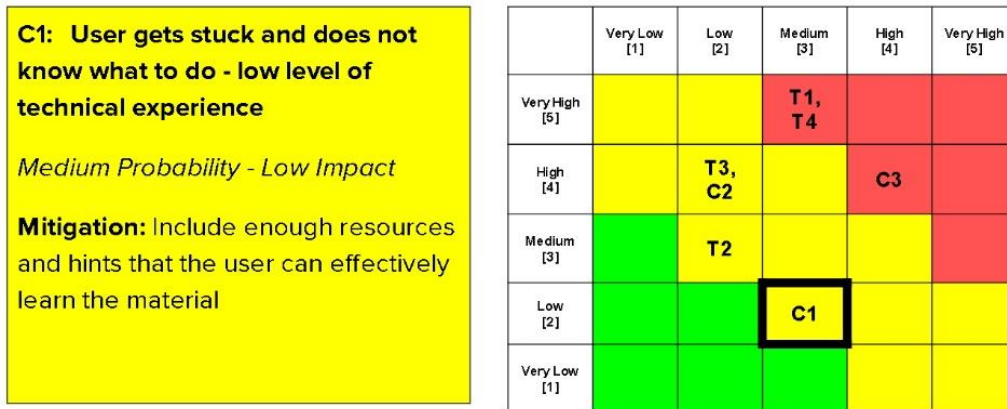
Team Silver. (2017, December 13). Prototype PowerPoint Presentation. In *PolyMorpher*. Retrieved from [https://docs.google.com/presentation/d/e/2PACX-1vSidnjCKAuVEtKshHkyO7A-OfW3qWIKRkxcp0em412WwL1ig6SFmnqrMUyHr8-FMvzvaRjmcKYiCytq/pub?start=false&loop=false&delayms=3000&slide=id.g25ab9a9d23\\_0\\_1542](https://docs.google.com/presentation/d/e/2PACX-1vSidnjCKAuVEtKshHkyO7A-OfW3qWIKRkxcp0em412WwL1ig6SFmnqrMUyHr8-FMvzvaRjmcKYiCytq/pub?start=false&loop=false&delayms=3000&slide=id.g25ab9a9d23_0_1542)

Team Silver. (2017, November 21). Design PowerPoint Presentation. In *PolyMorpher*. Retrieved from [https://docs.google.com/presentation/d/e/2PACX-1vSllslBDmSvRfMI9nbrp0RmRaPRsHNz7YWDfKNiF5sg15cp7ycQ774MuMgm4G4qhR6hohTiUQrrjRdo/pub?start=false&loop=false&delayms=3000&slide=id.g25ab9a9d23\\_0\\_1542](https://docs.google.com/presentation/d/e/2PACX-1vSllslBDmSvRfMI9nbrp0RmRaPRsHNz7YWDfKNiF5sg15cp7ycQ774MuMgm4G4qhR6hohTiUQrrjRdo/pub?start=false&loop=false&delayms=3000&slide=id.g25ab9a9d23_0_1542)

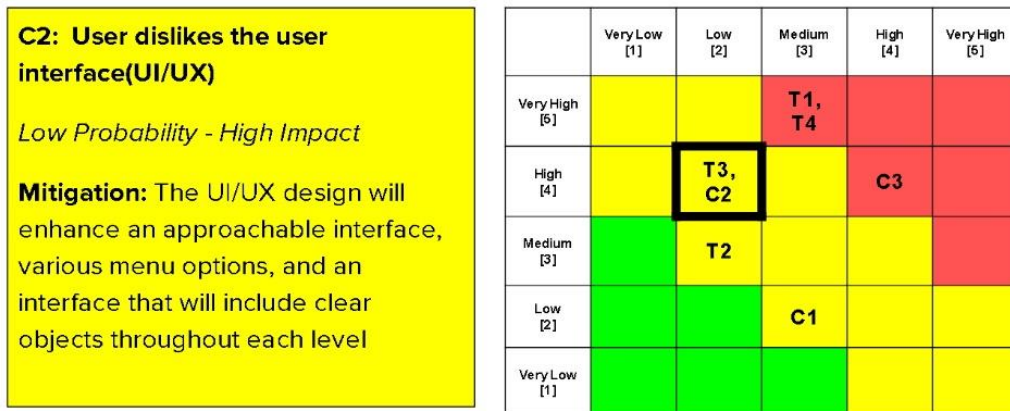
Team Silver. (2017, October 25). Feasibility PowerPoint Presentation. In *PolyMorpher*. Retrieved from [https://docs.google.com/presentation/d/e/2PACX-1vReG6Sodx-gVFro1ByYMOYHSyiSRiU5HW-Su-PyMVG08F4CQ7pY49tB\\_pJecVApruksoGaP\\_00ozhmR/pub?start=false&loop=false&delayms=3000&slide=id.g25ab9a9d23\\_0\\_1542](https://docs.google.com/presentation/d/e/2PACX-1vReG6Sodx-gVFro1ByYMOYHSyiSRiU5HW-Su-PyMVG08F4CQ7pY49tB_pJecVApruksoGaP_00ozhmR/pub?start=false&loop=false&delayms=3000&slide=id.g25ab9a9d23_0_1542)

Figures and Tables

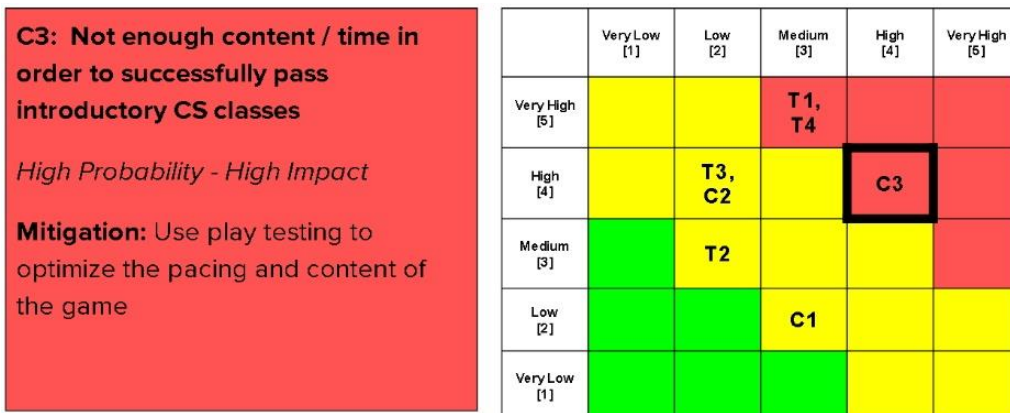
Risk Matrix C1:



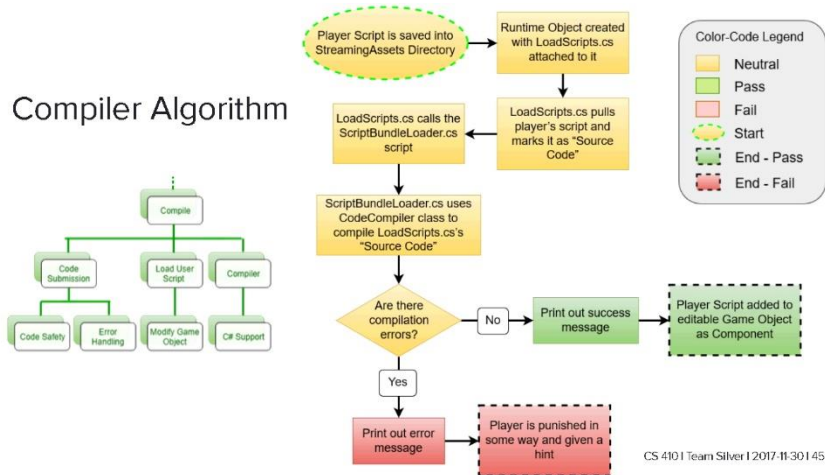
Risk Matrix C2:



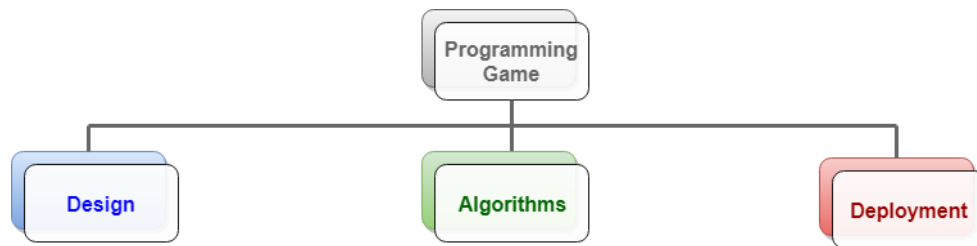
Risk Matrix C3:



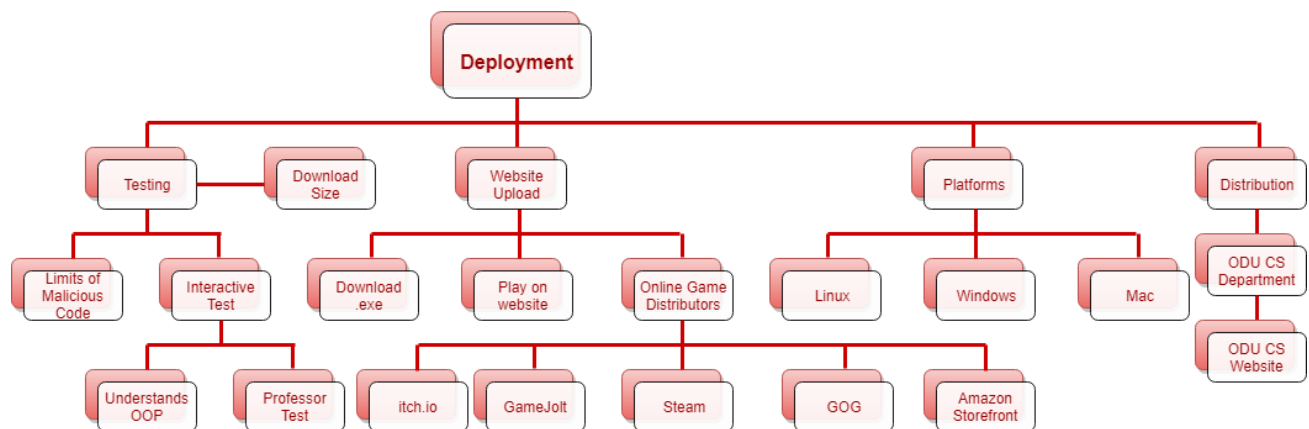
Compiler Algorithm:



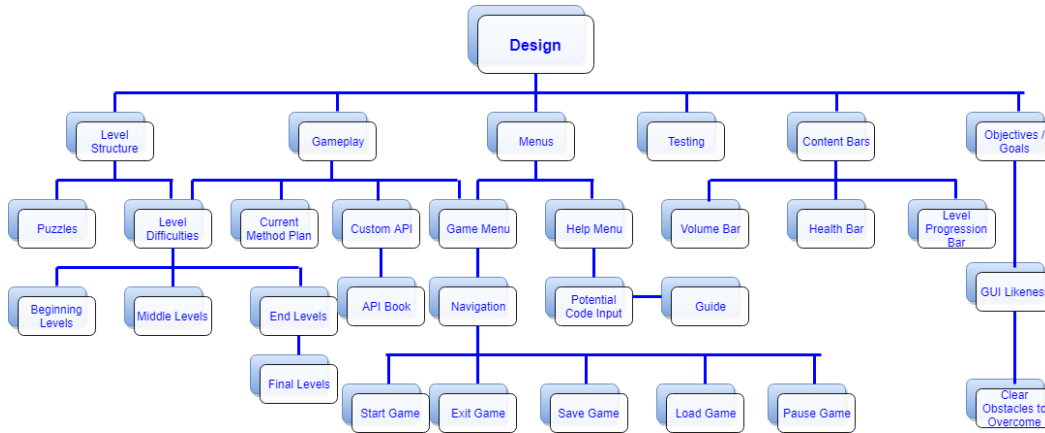
Work Breakdown Structure - Overview:



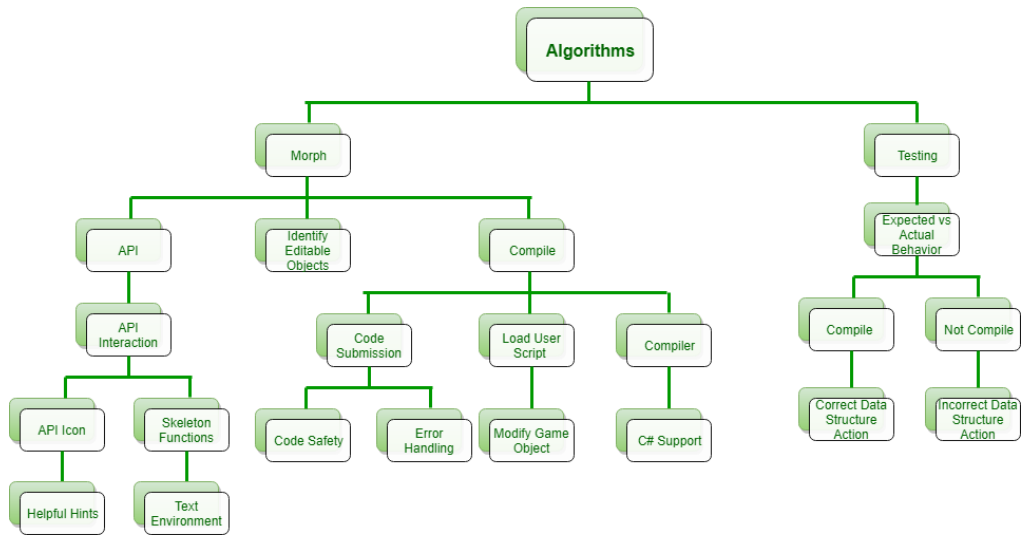
Work Breakdown Structure – Deployment:



Work Breakdown Structure - Design



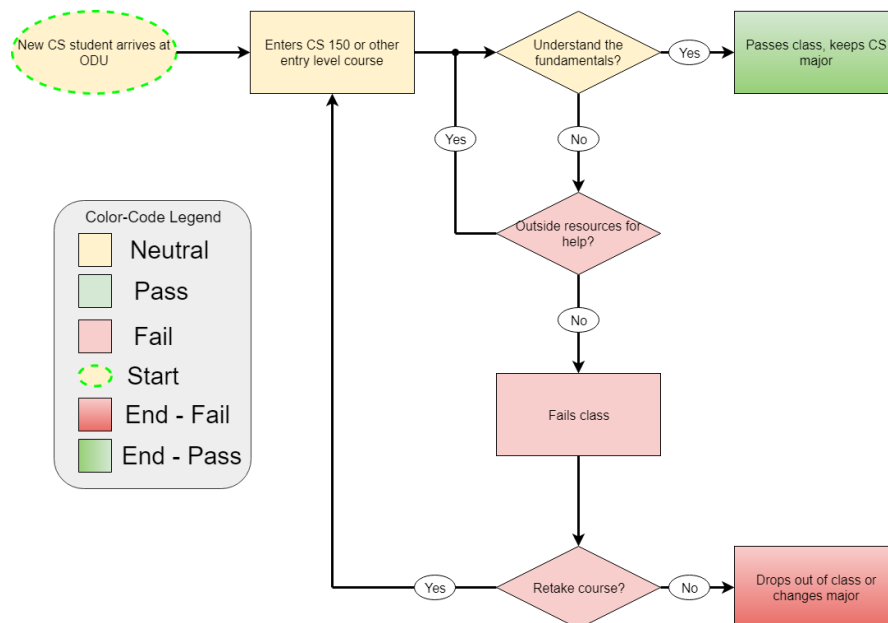
Work Breakdown Structure – Algorithms:



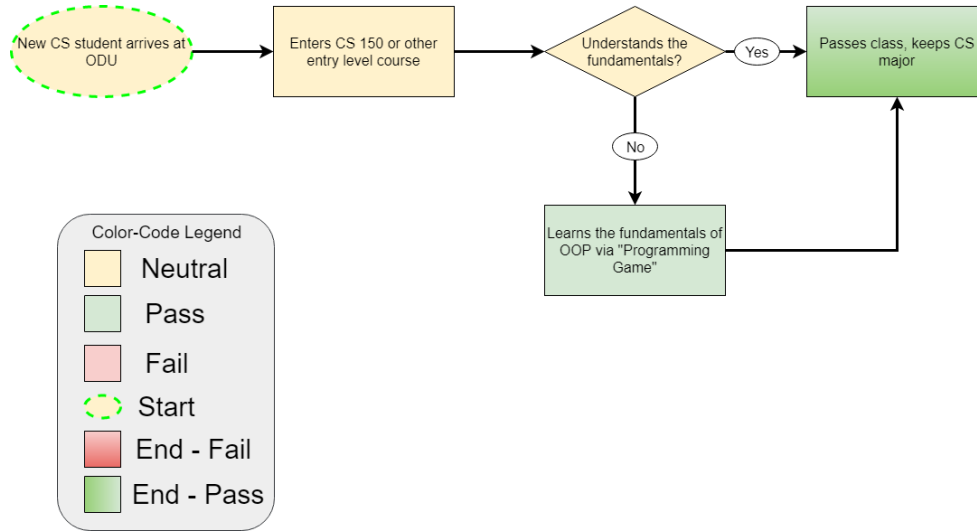
Competition Matrix:

Game	Experience	Uses OOP	Teaches OOP	# Languages	Multiplayer
PolyMorpher	Low-Mid	Yes	Yes	1	No
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

Current Process Flow:

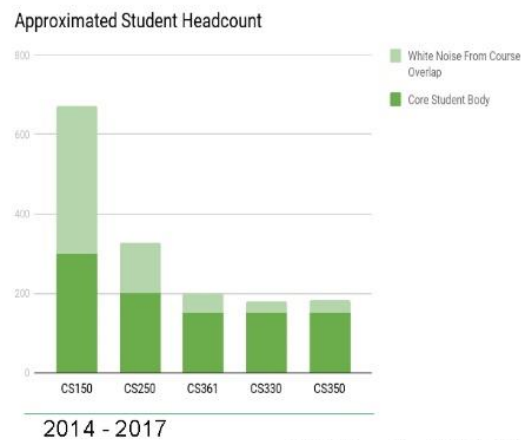


Solution Process Flow:



Student Progression Dilemma:

	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



User Stories – Player:

### Appendix A : User Stories - Player

1. As a player, I would like to interact with an intuitive UI.
2. As a player, I would like to be able to track my progress.
3. As a player, I would like to be able to freely manipulate game objects.
4. As a player, I would like challenges that are meaningful and rewarding (quality).
5. As a player, I would like challenges that are engaging (difficulty).
6. As a player, I would like save my progress.
7. As a player, I would like to access to various game menus.
8. As a player, I would like help when I fail at a challenge.
9. As a player, I would like a tutorial on how to move in this game.
10. As a player, I would like a tutorial on how to edit objects in this game.
11. As a player, I would like a tutorial on how to use the coding environment in this game.

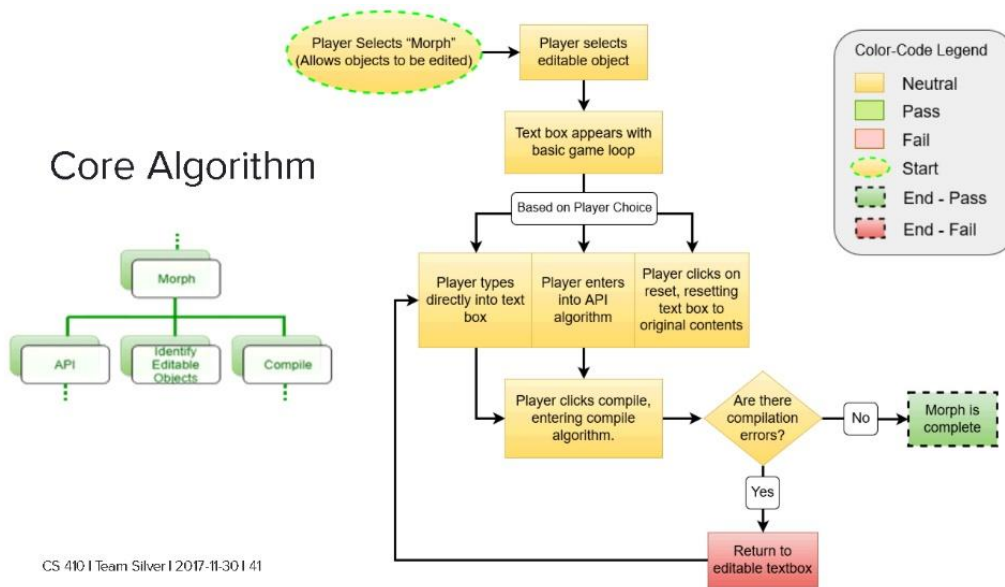
CS 410 I Team Silver I 2017-12-07 160

### User Stories – Student:

### Appendix A : User Stories - Student

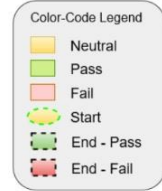
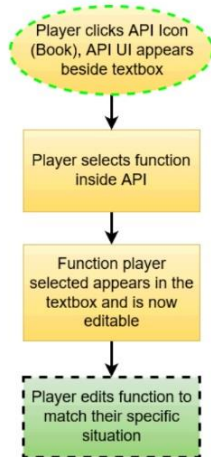
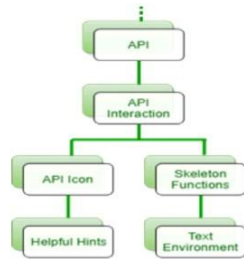
1. As a student, I would like to be taught inheritance.
2. As a student, I would like to be taught abstraction.
3. As a student, I would like to be taught encapsulation.
4. As a student, I would like to be taught polymorphism.
5. As a student, I would like to be taught functional design patterns.
6. As a student, I would like to be taught how to use an API.
7. As a student, I would like to be taught what all is contained inside an object.

### Core Algorithm:



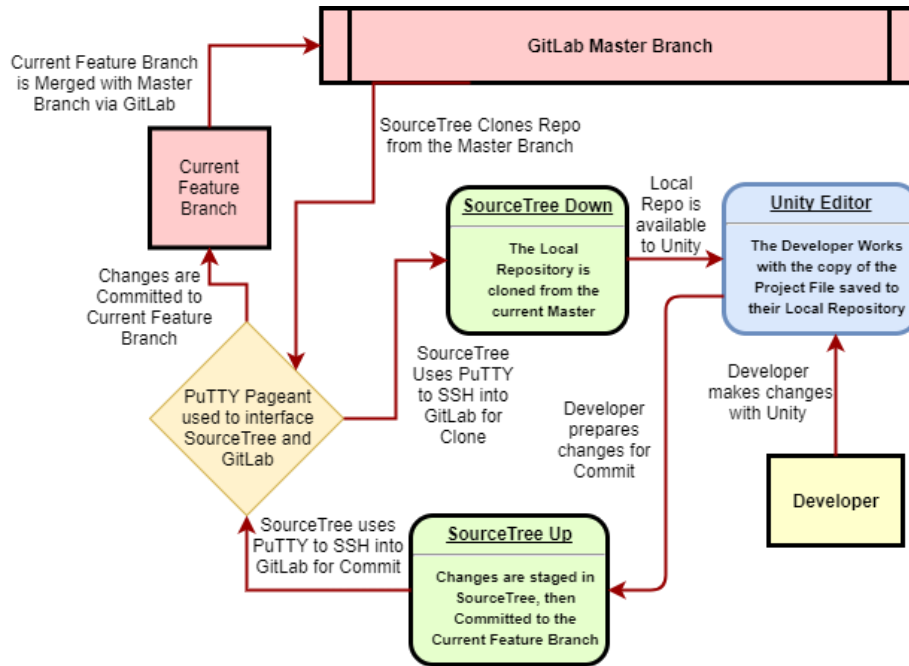
### API Book Algorithm:

API Book Algorithm



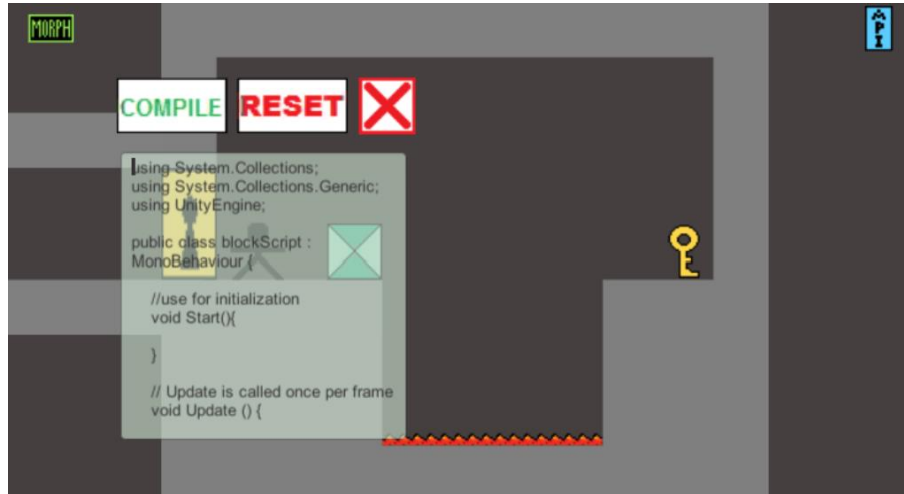
CS 401 Team Silver | 2017-11-30 | 43

Version Control Flow Diagram:

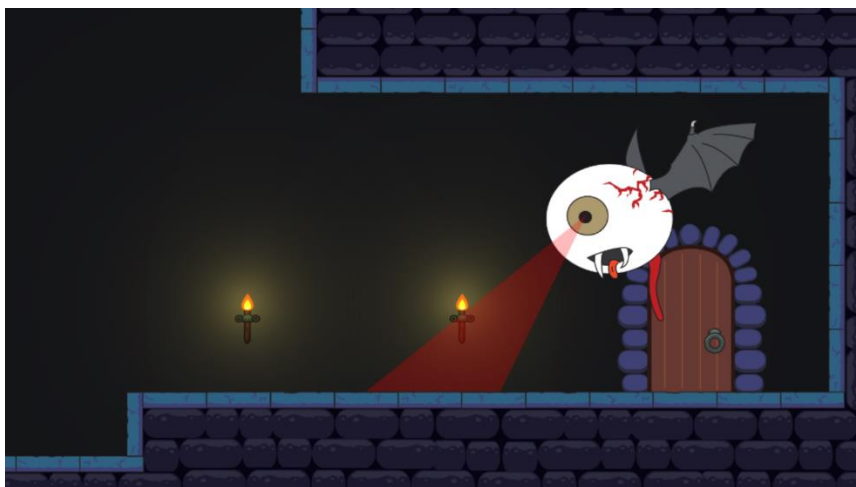


Demonstration Art Style:





Prototype Art Style:



Major Functional Components Diagram:

