Lab 1 - PolyMorpher Product Description

Kevin Santos

Old Dominion University

CS411W

Professor Thomas Kennedy

21 February 2018

Version 3

Table of Contents

List of Figures

List of Tables

Lab 1 - PolyMorpher Product Description

1. Introduction

Programming is a technical skill that requires one both retain and apply the skill in practice. Programming can be very intimidating for those not committed to learning it. Programming requires a different perspective. It requires a logical angle that laymans in this fields are not used to thinking. Traditional forms of teaching Object Oriented Programming and problem-solving skills at Old Dominion University and other colleges/universities are not producing the desired results. PolyMorpher is a game that is designed to teach the player complex OOP concepts and problems solving skills. For PolyMorpher to succeed, it must address the deficiencies and mistakes traditional academic learning has.

1.1 Team Members

A group of Computer Science majors from ODU students proposed and are currently developing a solution to this particular to this problem under the guidance of Thomas J. Kennedy, who is their mentor. Team Silver consists of:

- Casey Batten
- Peter Riley
- Matthew Tuckson
- Kevin Santos
- Colten Everitt
- Daniel Dang
- Nathaniel DeGrace
- Tyler Johnson

1.2 Problem Statement

As discussed in section 1, programming can be an intimidating skill to learn. ODU programming students often dropout or switch majors because of the rigor required to learn Object Oriented Programming(OOP) and problem solving skills. The traditional academic techniques and tools used to teach OOP do not work effectively. Section 1.3 discusses dropout and major change statistics.

1.3 Problem Characteristics

Team Silver's mentor, Mr. Kennedy provided very astonishing ODU records that show a decreasing trend of CS students moving successfully forward to advance level courses. Furthermore, the volume of students enrolled in 300 level CS courses at ODU has decreased significantly. Figure 1 shows the current data flow diagram of the possible paths CS students take towards their CS degree. It illustrates both conditions if the student successfully understands the OOP concepts and proceeds to the next CS course or vice versa. Figure 2 shows courses enrollment statistics.

[This space intentionally left blank]

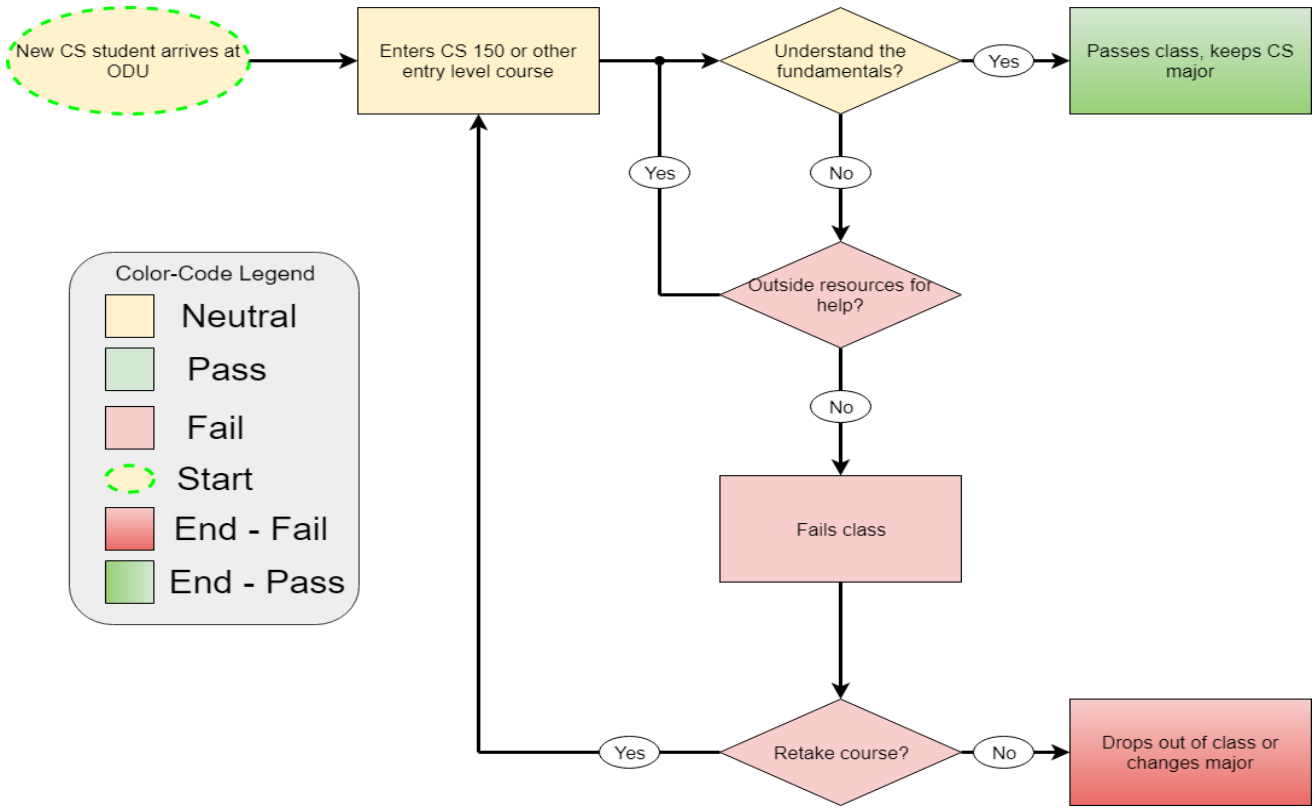
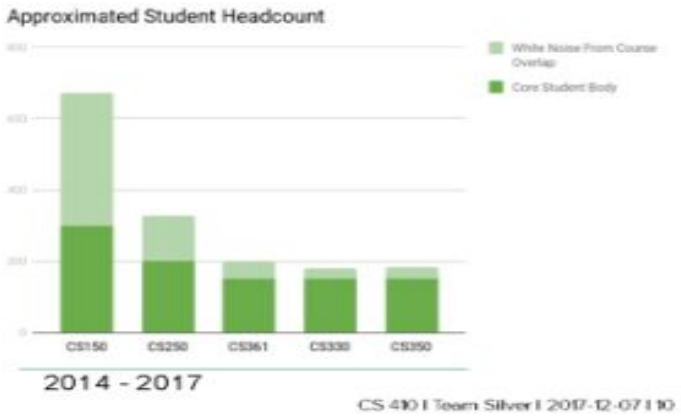


Figure 1: Current Process Flow Diagram 1

[This space intentionally left blank]

	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



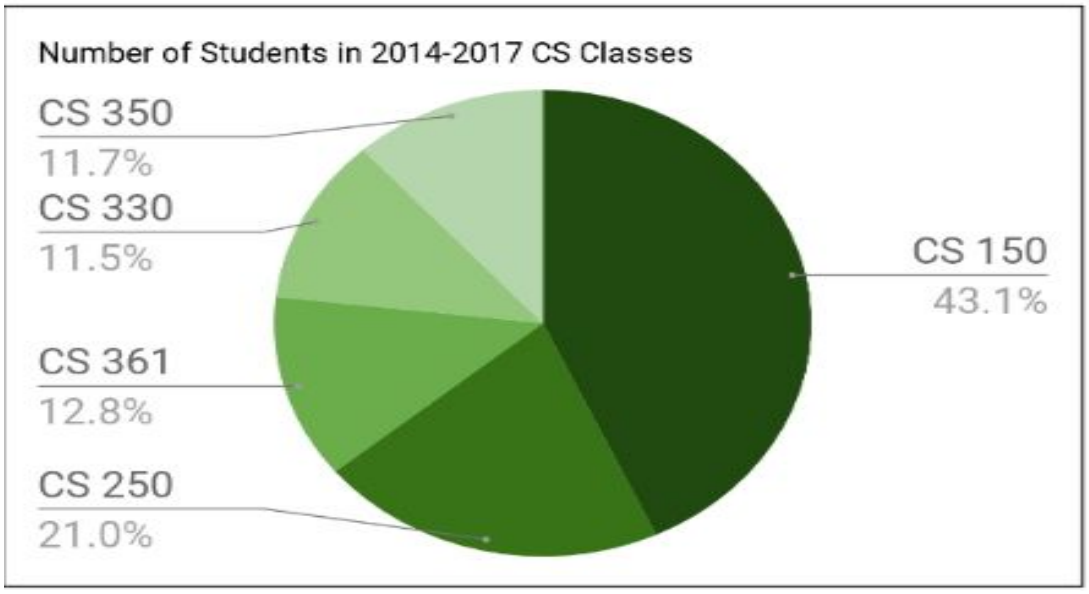


Figure 2: Combined Matrix, Bar Graph and Pie Chart

1.4 Solution Characteristics

PolyMorpher is a game that is intended to teach the player OOP concepts in an engaging and interactive way. The game will make use of an Tangible User Interface (TUI) and a management simulator. Polymorpher will improve on using and teaching OOP concepts that are not strictly addressed by its competitors as illustrated in Table 1. Figure 3 shows the ideal scenario if a student at ODU understands the fundamentals of OOP and continues as a CS major progressing through the next required courses.

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

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

Table 1: Competition Matrix

[This space intentionally left blank]

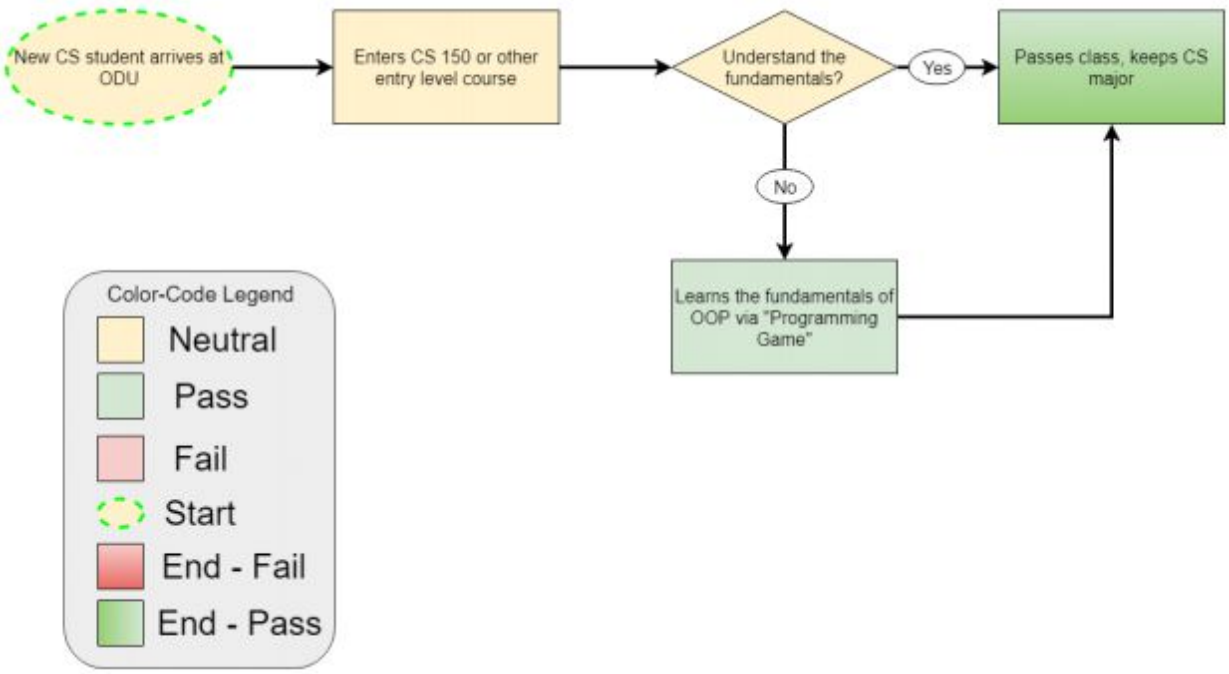


Figure 3: Solution Flow diagram

2 Product Description

The uniqueness of PolyMorpher (introduced in Section 1.4) is that it is a game that teaches OOP concepts in a way that is interactive and engaging. Research conducted by Team Silver through an article published by the Office of Naval Research (ONR) states that games, in general, are interactive and engaging. Thus, PolyMorpher will take advantage of that to teach OOP concepts that traditional academic learning does not address. PolyMorpher will consist of three important concepts:

- Realist Approach
- Improvement on Teaching
- Balanced Gameplay

The Realist Approach allows the game to use applicable and retractable examples to teach the player OOP. Improvement on Teaching will consist on facilitating the teaching of complex OOP concepts to the player. The Balanced Gameplay concept will apply both the gaming and learning experience.

The competition matrix in Table 1 will show that most of the competitors do not primarily focus on OOP. Traditional educational tactics towards teaching OOP concepts are not effective to new learners. PolyMorpher will effectively teach the players the important aspects of OOP. Only 1 of 14 competitors teaches OOP concepts which is the focus of PolyMorpher.

2.1 Major Functional Components (Hardware/Software)

PolyMorpher will be compatible with modern versions of Windows, Linux, and MacOSX. PolyMorpher will be implemented as a 2-Dimensional game that will draw minimal computing resources. Polymorpher will be optimized to perform with Intel's 4th generation i3 processor. The player will need an internet connection to navigate to Team Silver's website to download PolyMorpher. Figure 3 shows the major functional components.

[This space intentionally left blank]

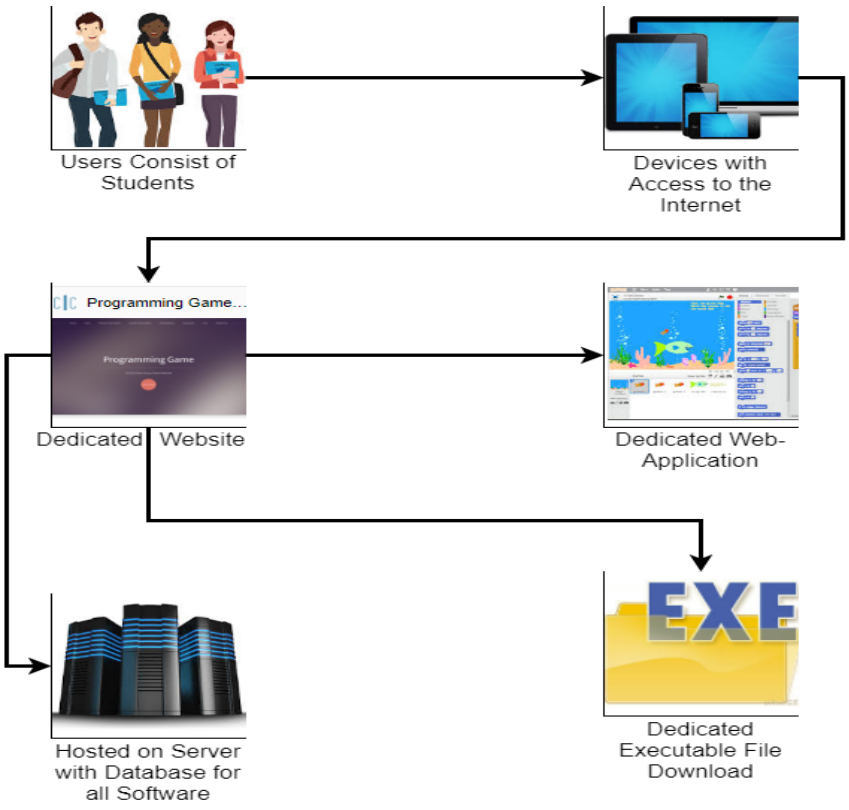


Figure 4: Major Functional Component Diagram

3 Identification of Case Study

The potential end users for PolyMorpher are students from ODU or other colleges/universities. Potential customers for PolyMorpher are in same category as end users but extend to people from different academic and work areas.

3.1 End-User: Students

The end users for PolyMorpher will be primarily students who want to learn programming based on OOP concepts. PolyMorpher is also geared towards helping students that enjoy playing games as a hobby. Thus, PolyMorpher provides the benefit of letting students play and simultaneously teaching complex OOP concepts.

3.1.1 ODU

One of the primary end users are particularly ODU students in the CS department. According to the enrollment data shown in Figure 2, ODU students are negatively affected by the constraints regarding the traditional teachings of OOP.

3.1.2 Other colleges

Other colleges and universities will be potential end users. Particularly, their CS students will benefit greatly from PolyMorpher because it will address the difficulties they are possibly having with comprehending and understanding OOP concepts and problem-solving skills. PolyMorpher will not have to accommodate to particular colleges/universities regarding their CS curriculums because they are similar to the one at ODU. Thus, PolyMorpher will help CS students from other colleges/universities because the same solution will help them as well.

3.1.3 Those generally interested in programming

The data in Figure 2 also shows that students from various majors are struggling with learning OOP concepts. PolyMorpher can help them because it focuses on dealing with the teaching and interactive aspect of the game instead of the individual itself.

3.2 Customers

Potential customers include programming instructors. Besides students, instructors can greatly benefit from using PolyMorpher. The reason is that if they can benefit from learning OOP concepts and problem-solving skills, then they can teach it effectively to their students. Instructors can also use PolyMorpher as part of their programming course curriculum as a new effective teaching tool.

4 Product Prototype Description

The prototype deliverable will be an exact simulation of the possible final product as illustrated in the tech demos by Team Silver. No external hardware is required for using PolyMorpher besides the player's PC.

4.1 Prototype Architecture (Hardware/Software)

PolyMorpher prototype architecture will be identical as the one in final product.

4.2 Prototype Features and Capabilities

Both the PolyMorpher prototype and final product will be able to be downloaded from its website as stated in section 2.1. The prototype and final product will also be compatible with most common operating systems, Windows, Linux, and MacOS. Table 2 summarizes the 16 possible differences between the PolyMorpher Prototype and the final product.

[This space intentionally left blank]

Elements	Description	Real World Product	Prototype
Teaches Polymorphism	Provision of a single interface to entities of different types		
Teaches Abstraction	Technique for arranging complexity of systems		
Teaches Encapsulation	Building of data with the methods that operate on that data		
Teaches Inheritance	When an object or class is based on another object or class, using the same implementation		
Single Language Taught	A single programming language will be focused on C#.		
Single Player	Focused on an experience targeted to interact with only one player		
Downloadable .EXE File	Desktop application version of the game		
Game Assets	Primary components that are used as building block to construct the more complex features and levels of the game		
Developed Story	Narrative used to drive progression or direct player throughout a more guided/linear experience		
Portable Compiler	Code compiler used to run player-made code on the fly in game		
Tutorial Section	Precursor series of levels meant to help the player adjust to the in-game toolset given to them and also prep them with knowledge of the language(s) they will be working with		
Multiple Platforms	Version support for multiple operating systems (Windows, Mac OS, Linux)		
Sandbox Level	Open level where the player has access to all tools at once and can build their own level sequences and puzzles		
Player-Made Content	Variant of Sandbox Level, potentially allows the player to share custom levels with one another		
Multiple Player	An experience geared toward multiple players interacting with a game environment together		
Web Application	Web based version of the game running in-browser		
Multiple Languages Taught	Alternative programming languages for the player to use and learn in-game		

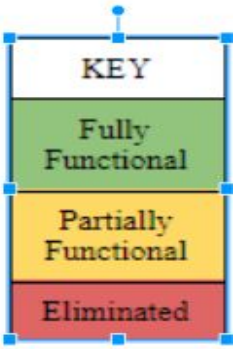


Table 2: Features of the Completed Product against the Prototype

It will be guaranteed that PolyMorpher will be able to teach some of the most important OOP concepts that includes Abstraction and Encapsulation. In addition, the player will be able to benefit from a story progression and a tutorial section so that they can see their amelioration as they move from one level to another while they play through the game. One of the partially functional features that might be implemented on PolyMorpher is a sandbox level. This is where the player is allowed to have access to all the tools available in the game so that they can design

their game levels and puzzles. Lastly, the eliminated features of PolyMorpher for both the prototype and the final product are multiplayer, multiple languages taught, and the web application. All three of these applications requires a constant internet connection.

4.2.5 Algorithms

PolyMorpher prototype will consist of three important algorithms that will contribute to its capabilities and features. The three algorithms that will used for this game are the following:

- Core Algorithm
- API Book Algorithm
- Compiler Algorithm

Sections 4.2.5.1 through 4.2.5.3 will go into further details explaining how each of three algorithms will work.

4.2.5.1 Core Algorithm

The Core Algorithm will consist of one component option named “Morph” that will deliver three sub options. The procedure of this algorithm starts with allowing the player to select the main option of “Morph” which allows objects in the game be edited. Then, the algorithm through the game will create a text box with a game loop that will give the player three options. The player can choose to type their own code directly into the text box to modify the script, choose a function from the API Book, or reset the script to its original content. This algorithm with the aid of the compiler, will determine if there were any compilation errors from the input entered. Figure 5 illustrates the Core Algorithm in detail.

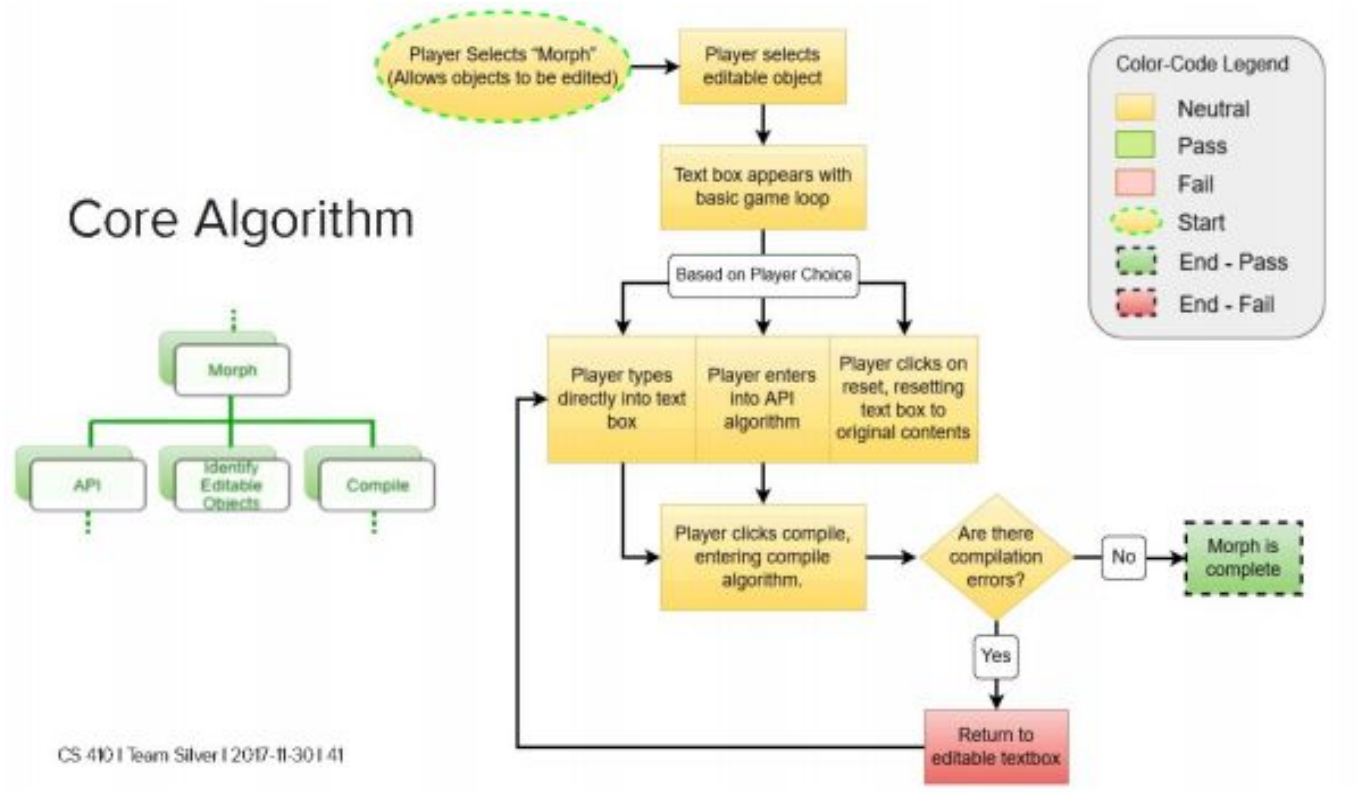


Figure 5: Dataflow Diagram for the Core Algorithm

4.2.5.2 API Book Algorithm

The API Book Algorithm will consist of allowing the player to access the API Book that PolyMorpher provides. This algorithm administers the educational information that Team Silver had decided to implement in their design and grants that to the player. The API Book provides a variety of tools to help them complete the puzzles and challenges in each level. This algorithm consists of three simple procedures. First, the player selects the API icon which will be an image of a book that appears in a textbox. Then, they select a function inside the provided API Book. Lastly, the function the player selects appears in the textbox which it can be editable and manipulated. Figure 6 shows API Book Algorithm diagram for further details.

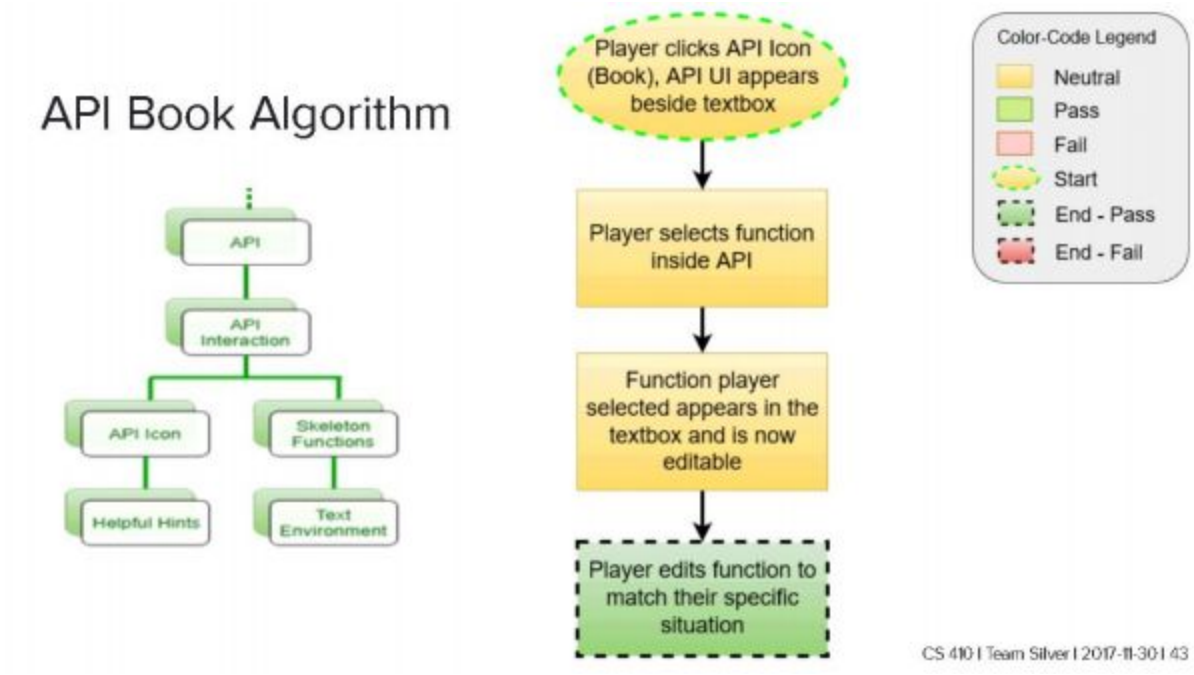


Figure 6: Dataflow Diagram for the API Book Algorithm

4.2.5.3 Compiler Algorithm

The Compiler Algorithm will act as a portable assembler to process the scripts the player enters as they are solving a puzzle or challenge. The procedure starts with saving the player's script in the StreamingAssets Directory. Next, a Runtime Object is created, and it functions along with a premade LoadScripts.cs file. This file pulls the player's script and marks it as "Source Code".

The LoadScript.cs file calls another premade file called ScriptBundleLoader.cs to perform the portable compilation on the source code. ScriptBundleLoader.cs has a CodeCompiler class that performs the compilation of the player's stored script. ScriptBundleLoader.cs will determine if there were compilation errors in the player's script. Figure 7 illustrates the Compiler Algorithm.

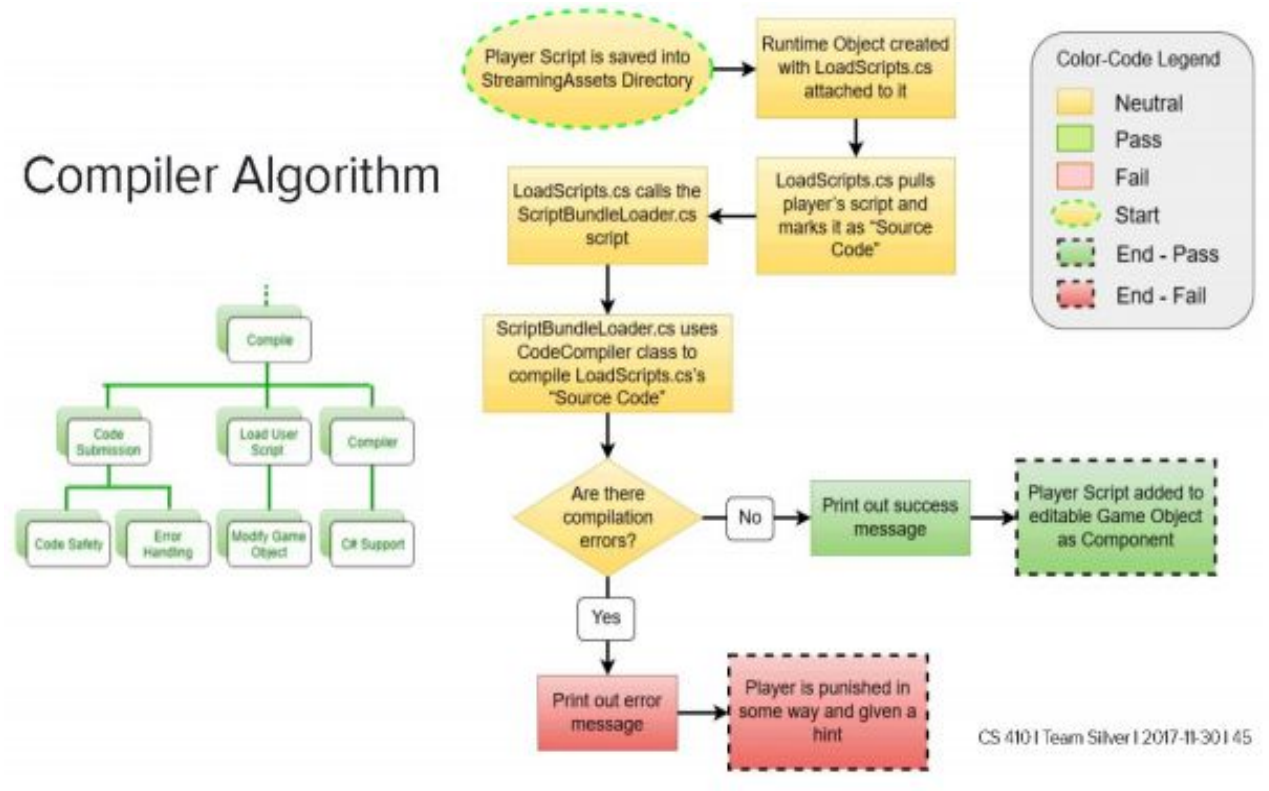


Figure 7: Dataflow Diagram for the Compiler Algorithm

4.3 Prototype Development Challenges

The development of the PolyMorpher prototype by Team Silver will encounter five challenges. The subsections will elaborate further on each of those.

4.3.1 Design continuity

The consistency of the design for PolyMopher is imperative because it promotes the stability for the prototype and final product. Team Silver plans to maintain the design of PolyMorpher consistent through weekly meetings.

4.3.2 Difficulty debugging

A rigorous development challenge that Team Silver will encounter is debugging PolyMorpher. One aspect that Team Silver must consider is debugging the numerous and vague

scripts the player can enter. Thus, Team Silver will strategically find ways to cause errors and glitches within PolyMorpher to debug it effectively.

4.3.3 Playtesting

Playtesting is an important challenge that Team Silver must take into consideration. It requires to test if the main components of PolyMopher are working efficiently. The main components are:

- TUI and GUI functionalities
- Compilation of player's code
- The effectiveness API Book references

The challenge constraints will be noted on the lengthy time it might take to test these components individually.

4.3.4 Maintaining player engagement

Keeping the player engaged and entertained is another major challenge for PolyMorpher. Playtesting results will be a major aid to Team Silver in determining how entertaining this game will be.

4.3.5 Teaching enough

The goal of PolyMorpher will be to teach the main concepts of OOP. However, in programming there are a lot of other important concepts that can be beneficial to the player. Team Silver is faced with a bigger challenge that is if teaching enough is only sufficient with OOP.

4.4 Risk Matrix/Risk Mitigation

The risk mitigation and matrix for PolyMorpher (Table 3) lists the three main scenarios of impact and probability risks. There are two important categories as well which are Customer and Technical. The matrix is divided into three colored sections.

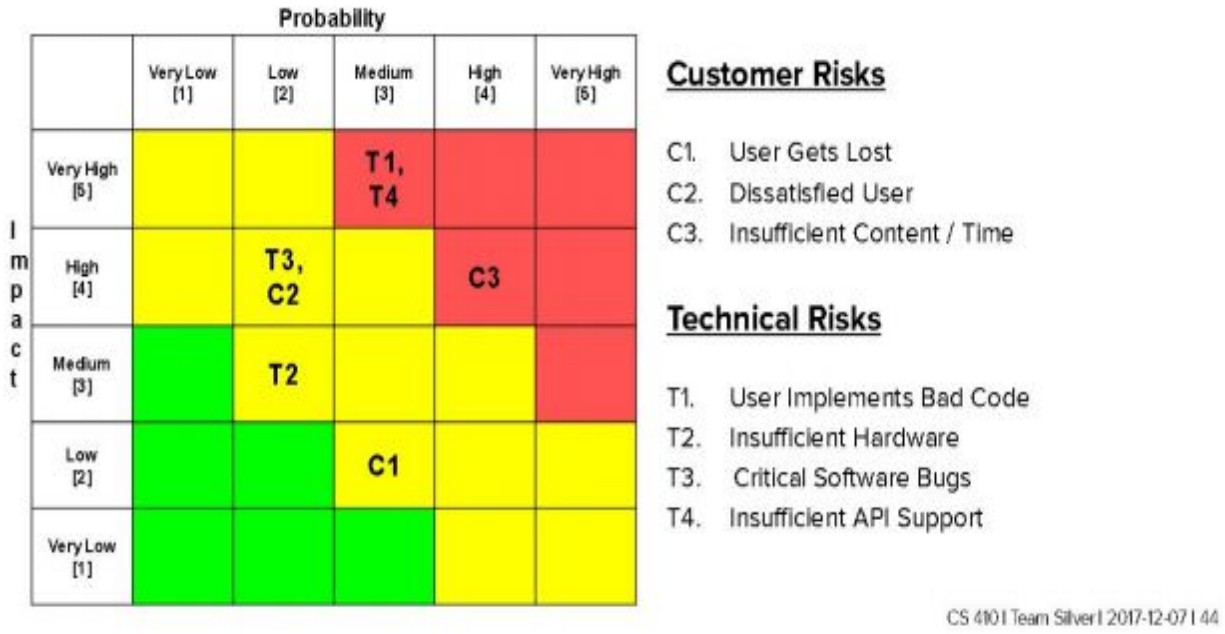


Table 3: Risk/Mitigation Matrix of Customer & Technical Risks for PolyMorpher

4.4.1 Technical risks and mitigations

There are four potential technical risks. The first technical risk is that user implements bad code. The mitigation for this risk is to encourage the user to use the provided API Book for their scripting references instead of them using potential malicious code from online forums. The second technical risk deals with there being insufficient hardware for the game itself. The mitigation for this risk is that PolyMorpher will be implemented as 2-dimensional game minimizing required machine resources.

The third technical risk deals with critical software bugs in the game. The mitigation for this risk is to test and assure that the game is fully optimal and functioning at the desired expectations. The last and fourth technical risk deals with insufficient API support. The mitigation for this risk is to ensure that the built-in API Book provides all of the necessary tools for the user to reference OOP concepts and particular code syntax that can help them get through the levels of the game successfully.

4.4.1 Customer risks and mitigations

There are three main customer risks and mitigations. The first customer risk deals with the user getting stuck in solving a puzzle or challenge. The mitigation for this risk is to ensure that the gameplay will have the necessary hints and resources to allow the user to not get stuck and learn the OOP material effectively. The users who can be affected by this risk either have low or no programming experience.

The second customer risk is if the player is dissatisfied particularly with GUI and UI/UX design. Team Silver is committed though to find and implement a GUI and UI/UX design that can satisfy most of its users.

The third and final customer risk deals with the insufficient content and time that PolyMorpher will be able to teach to its players. The mitigation of this risk is to ensure that PolyMorpher will have the important and sufficient OOP content so that the user can learn the material effectively. Team Silver also took into consideration time constraints. This is to ensure that the user will have the sufficient time to play the game and learn OOP concepts at the same time.

5 Development Pipeline

The tools needed to successfully develop PolyMorpher include Unity SDK, Gitlab, SourceTree, MonoDevelop, and C# programming language.

5.1 Unity SDK

Unity is a popular gaming engine that has the sufficient development built-in tools to construct PolyMorpher as a game. Unity has many premade objects and environments that can be useful for developing the GUI and UI/UX design. MonoDevelop is the IDE that is integrated into Unity to develop games and the programming language that will be used to do it is C#.

Team Silver will use C# to build PolyMorpher. Team Silver decided to use MonoDevelop because it is an integrated feature of Unity.

5.2 SourceTree

SourceTree is a free git client for Mac and Windows. SourceTree uses PuTTY which is a popular pageant client to connect and communicate with GitLab remotely.

5.3 Version Control

Team Silver's version control will require SourceTree, Gitlab, and PuTTY to connect and communicate simultaneously with their repositories. Figure 8 illustrates Team Silver's version control process.

[This space intentionally left blank]

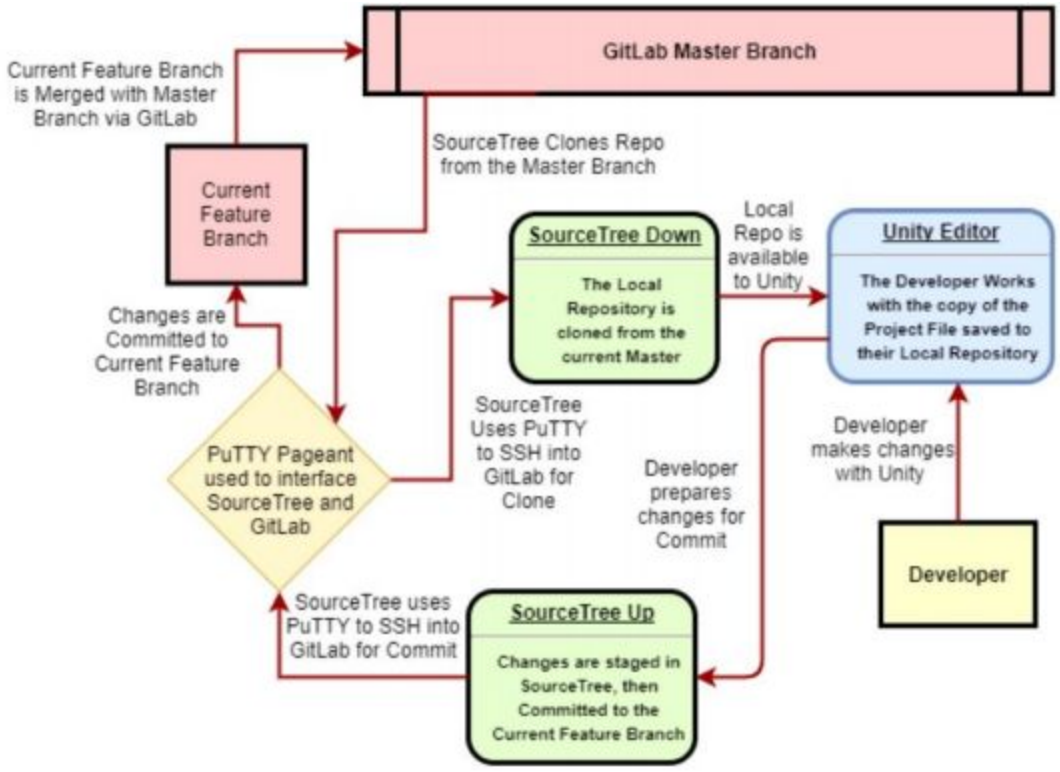


Figure 8: Dataflow Diagram for Version Control

5.4 Development Model

Team Silver will use the Agile Development Model. As illustrated in Figure 9, this development model shows quick versions of prototype models done each of the developers. It encourages a rapid workflow among the developers since if one prototype model is not liked by the team, they move on to next.

[This space intentionally left blank]

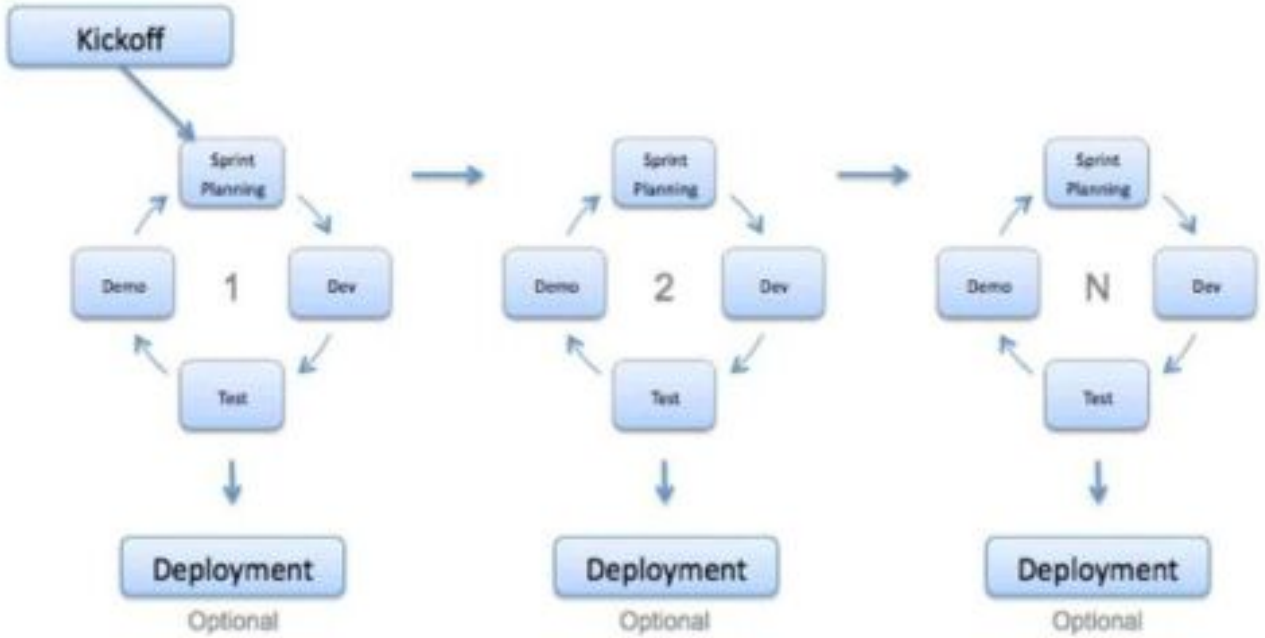


Figure 9: Dataflow Diagram for Agile Development Model

5.5 Work Management

Team Silver will be split into three development groups. Each group will be assigned two OOP concepts and they will be given the task to design a puzzle or challenge for a level. They have to design such puzzle and challenge to effectively teach their assigned OOP concepts. Once all three groups successfully complete their tasks, the levels will be integrated into PolyMorpher.

Glossary

API: Application Program Interface

Computer: a programmable electronic device designed to accept data, perform prescribed mathematical and logical operations at high speed, and display the results of these operations

Computer Programming: a process that leads from an original formulation of a computing problem to executable computer programs

Computer Science (CS): the science that deals with the theory and methods of processing information in digital computers, the design of computer hardware and software, and the applications of computers

Design: an outline, sketch, or plan, as of the form and structure of a work of art, an edifice, or a machine to be executed or constructed

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

IDK: Integrated Development Kit

Management Simulator: a way to simulate the management of a game in an organized fashion

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

Object-Oriented Programming (OOP): A schematic paradigm for computer programming in which the linear concepts of procedures and tasks are replaced by the concepts of objects and messages

ODU: Abbreviation for Old Dominion University

PC: Personal Computer that will be used to describe the player computer/machine

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

PolyMorpher: a programming game that focuses strictly on teaching OOP and problem solving skills

Problem Solving: the process of finding solutions to difficult or complex issues

Programming Game: a video game which incorporates elements of computer programming into the game, which enables the player to direct otherwise autonomous units within the game to follow commands in a domain-specific programming language

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

Software Development Kit (SDK): a set of software development tools that allows the creation of applications for a certain software package

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

Student Progression Dilemma: the problem of CS majors at ODU not advancing through the

CS course schedule in order to graduate with a CS degree

TUI: Tangible User Interface

Ubuntu: open-source Linux operating system

Unity: a popular game development platform

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

Virtual Machines: emulations of computer systems that provide functionalities of physical computers

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

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%220B3Bc95zBWXg9TFZ6X0FMU1NTdEk%22%5D,%22action%22:%22open%22,%22userId%22:%221086920031335905830 47%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%220B-5KdQEdqLUPWnNoSHhIUGg2OTQ%22%5D,%22action%22:%22open%22,%22userId%22:%221086920031335905830 47%22%7D#G0B-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_xBnZ1ge4PlZTVjV3h6Y2pGSWc%22%5D,%22action%22:%22open%22,%22userId%22:%221086920031335905830 47%22%7D#G0B_xBnZ1ge4PlZTVjV3h6Y2pGSWc

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/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

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/1oY9lkSAHvg2OIRkljYJNZWCqVTbiw45STKglsJUQjJI/edit#slide=id.g283e74317a_0_177

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

Team Silver. (2017, November 21). Design PowerPoint Presentation. In *PolyMorpher*. Retrieved from https://docs.google.com/presentation/d/e/2PACX-1vSllslBDmSvRfMI9nbrp0R

mRaPRsHNz7YWDfKNiF5sg15cp7ycQ774MuMgm4G4qhR6hohTiUQrrjRdo/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-PyMVGO8F4CQ7pY49tB_pJecVApruksoGaP_00ozhmR/pub?start=false&loop=false&delayms=3000&slide=id.g25ab9a9d23_0_1542

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