Lab 2 - PolyMorpher Prototype Product Specification

Joel Stokes

Old Dominion University

CS411W

Professor Thomas Kennedy

26 February 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

Table of Contents

List of Figures

List of Tables

1 Introduction

PolyMorpher is an eductional programming game designed to help teach students the fundamentals of Object-Oriented Programming through an entertaining medium. PolyMorpher can be run through most devices which can access the internet, making it highly accessible to potential users.

1.1 Purpose

PolyMorpher is presented as a 2D Platformer, in which the player can click on specific objects to directly edit their code. To complete each level, the player must edit certain objects using the OOP technique taught in the level to complete simple puzzles. For struggling players, additional help information can be accessed to show where specific errors in their code occurred, or to explain how the OOP technique for the level should be applied to the specific situation at hand. Completing the levels not only helps boost the player's programming abilities, but also visually shows them OOP techniques in action.

The original end user are Old Dominion University Computer Science undergraduates who have completed CS150, however depending on the success of the prototype, PolyMorpher will be geared towards a much broader audience, encompassing anyone hoping to learn OOP techniques. PolyMorpher is not meant for users who have no programming experience, as the main purpose centers around helping the user understand OOP rather than teaching introductory logic and syntax.

1.2 Scope

PolyMorpher's primary goal centers around helping Computer Science students struggling with programming overcome their mental roadblocks. By teaching OOP principles in a different way than conventional classes, PolyMorpher helps fill in the gaps of understanding through engaging, fun puzzles, helping those students to succeed.

PolyMorpher's prototype will focus on displaying a working demonstration of the concepts hoped to be taught in a fully-playable format. 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.

1.3 Definitions, Acronyms, and Abbreviations

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.

1.4 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:%22108692000313359058304 7%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

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/

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

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

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

1.5 Overview

This product specification provides an in-depth look at the features of PolyMorpher's prototype, as well as its' underlying interfaces and capabilities. For more information specifically related to the final product rather than the prototype, please view the external document: "Lab 1 - PolyMorpher Product Description".

2 General Description

The primary goal of the PolyMorpher prototype is to have a functional demo that can be tested by ODU CS undergraduates. The prototype will be fully playable, featuring six levels: the first being a tutorial for players to become familiar with the PolyMorpher code manipulation system, as well as a quick refresher on basic programming practices, and the other five will be focused on an individual principle of OOP. Having this finished prototype will be vital in showing the educational capabilities of PolyMorpher.

2.1 Prototype Architecture Description

The architecture of PolyMorpher's prototype will be identical to the architecture of the final product. The three major components of the architecture are:

- The website, which contains an online, fully playable version of the game, as well as an alternate downloadable version to play the game locally on the user's computer.
- The application, which is the stand-alone version which can be accessed through the website.
- The file structure, which contains all the data output through the Unity exporting process. More specifically, this is the "StreamingAssets" folder which allows the application to run, as well as save the code-manipulation changes made by the player within the game.

Figure 1 below shows the architectural breakdown and relation of the components.

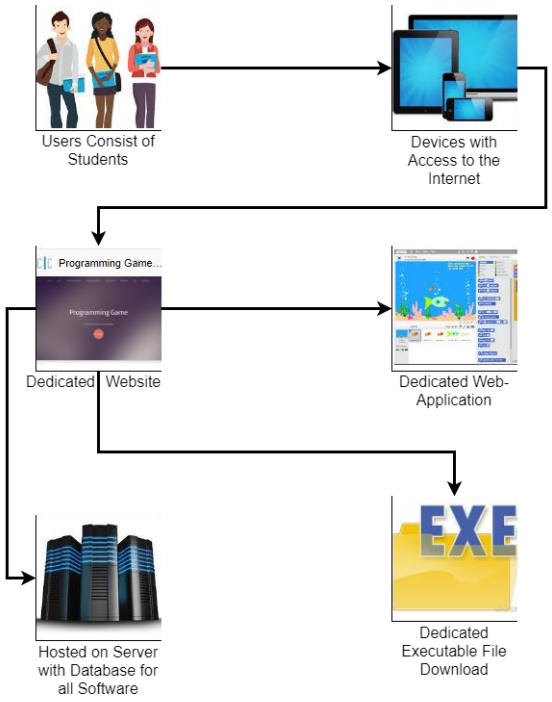

Figure 1: Major Functional Components Diagram (Team Silver, 2017)

2.2 Prototype Functional Description

PolyMorpher will be able to run on most devices which can connect to the internet, including smart phones and tablets. Locally, PolyMorpher will be able to operate on Windows, Mac, and Linux devices.

The most vital function of the prototype will be the code compiler. The compiler will be able to take what is written by the player and compile in real-time during game operation to view instant feedback inside the level. For the prototype, certain features, such as multiple programming languages, multiplayer, and the web-application, will not be available in the prototype version of the game. Also, a few extra features, such as the Sandbox level, which will allow players the freedom to experiment with all editable objects in one place, will only be partially implemented in the prototype. Shown below in Table 1 are all the key differences between the finished PolyMorpher product and the prototype.

KEY
Fully Functional
Partially Functional
Eliminated

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		
Player-Made Content	Variant of Sandbox Level, potentially allows the player to share custom levels with one another		
Sandbox Level	Open level where the player has access to all tools at once and can build their own level sequences and puzzles		
Multiple Languages	Alternative programming languages for the player to use and learn in-game		
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		

Table 1: Features of the Complete Product versus the Prototype (Team Silver, 2017)

See section 3.1 for a more detailed view into the specifics of all functions.

2.3 External Interfaces

The five primary types of external interfaces for Polymorpher is the Hardware Interface, Software Interface, API Book Interface, Compiler Interface, and the User Interface.

2.3.1 Hardware Interfaces

The PolyMorpher prototype will run strictly on the user's local machine. The minimum requirement is a 4th generation i3 Intel Processor. PolyMorpher will not be a resource-intensive game, and does not require a gaming graphics card.

2.3.2 Software Interfaces

PolyMorpher's software will be hosted and downloadable through the CS server at Old Dominion University, which holds Team Silver's PolyMorpher website. Windows (7, 8, 10), Mac, and Linux operating systems all can run PolyMorpher. With Unity's cross-platform development tools, moving from the prototype to the final online version of the product will be made much simpler.

2.3.3 API Book Interface

The API Book Interface is a visual aid for the player which helps display useful information regarding OOP techniques previously learned, and offers helpful code which can be clicked and added to an object currently being edited by the player. The API Book Interface can be brought up at any time by clicking the "API Book" icon in the top right of the screen during gameplay. Because of its nature to add code to currently edited objects, it is designed to work in tandem with the Compiler Interface. Figure 2 below shows the basic procedure of the API Book Algorithm.

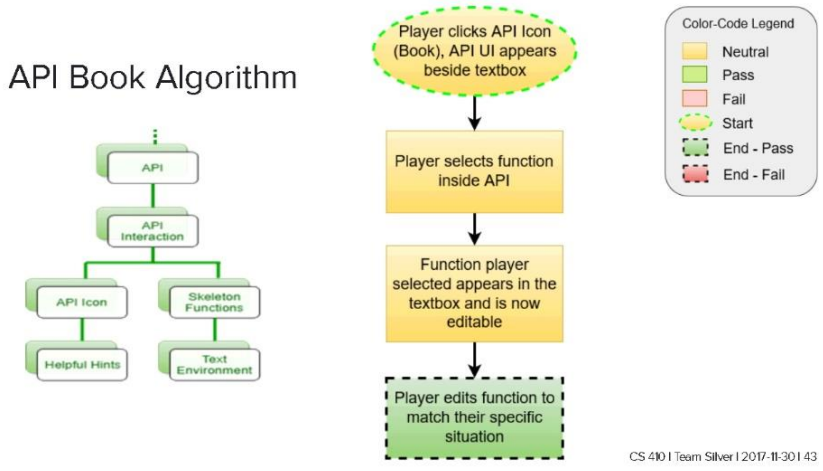

Figure 2: API Book Algorithm (Team Silver, 2017)

2.3.4 Compiler Interface

The Compiler Interface is the display shown while a player is currently editing an object's code. To bring it up, the player must simply click any object which glows when the mouse hovers over it. The previously planned prototype method (using the "Morph" button located at the top left of the screen, then clicking the desired object) was changed in early planning stages due to its unnecessary length. Figure 3 below shows a rapid prototype example of the Compiler Interface in action.

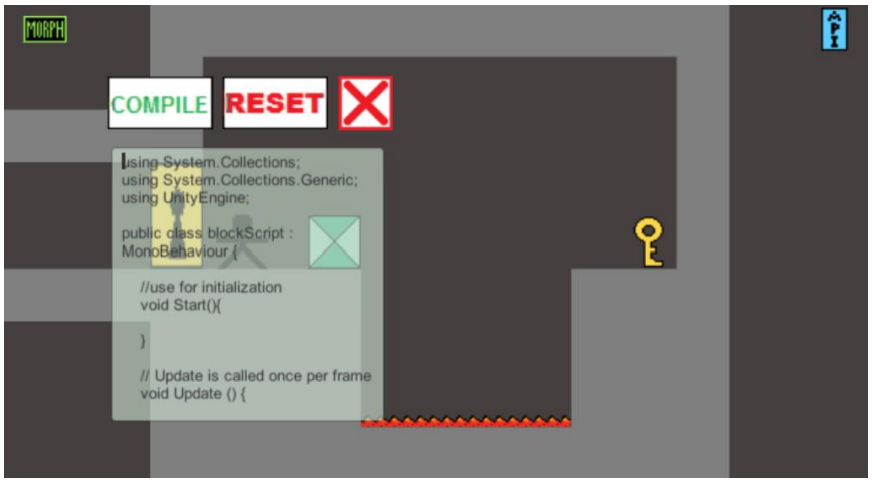

Figure 3: Demonstration Art Style (Team Silver, 2017)

Once the player is finished editing their code within the Compiler Interface, they can hit the green "Compile" button to apply their code to the object, "Reset" to revert the code back to the object's original, or the "X" to close the Compiler Interface without compiling their changes, but still keeping their changes available. As stated in 2.3.3, the API Book Interface can be brought up at any time while editing in the Compiler Interface to help aid the player during coding. Figure 4 below shows the basic logical flow of the compiler interface algorithm.

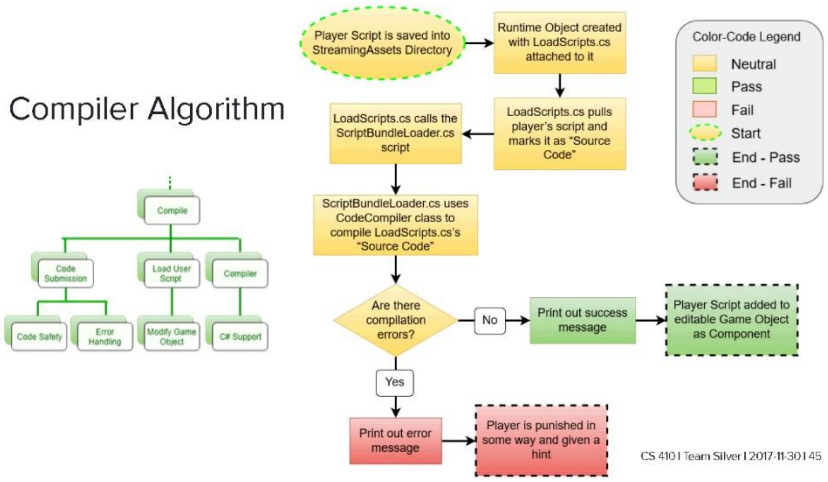

Figure 4: API Compiler Algorithm (Team Silver, 2017)

2.3.5 User Interface

The User Interface is the general play area of the current level, which also includes the Compiler and API Book interfaces. Displayed on the screen of the user's device will be a culmination of art from objects, enemies, and editable blocks, which is output through a camera system which follows the player's movement. The player moves through keyboard inputs (of either Left/Right or A/D for horizontal movement, and spacebar or W to jump) to make their player avatar travel through the level, ultimately having the camera follow their character's movement. Also, the mouse (or touchpad for later builds) is used to click on editable objects to bring up the Compiler Interface or API Book Interface. Then, once either interface is brought up, the player can click within the text region to use their keyboard as a normal word processor. Figure 5 below shows a mockup of the User Interface (not including the player character art).

Figure 5: Prototype Art Style (Team Silver, 2017)

3 Specific Requirements

Despite the hardware requirements specified in 2.3.1 and 2.3.2, along with the previous knowledge requirement for users in 1.1 (PolyMorph is not a product meant to teach introductory programming), there are no other necessary requirements for the user to operate this product. The prototype will be freely available for download from the CS server ODU Team Silver website, which will be operational 24/7. The current estimated release for the PolyMorpher prototype is April 2018.