Lab II - PolyMorpher Prototype Product Specification

Matthew Tuckson

Team Silver

Old Dominion University

CS411W

Professor Thomas Kennedy

26 February 2018

Version 1

Table of Contents

Table of Figures

Table of Tables

1 Introduction

PolyMorpher is a game that will give computer science students a resource to help develop their object-oriented programming (OOP) skills. The player will begin as a wizard who has been caught inside a dungeon full of monsters and puzzles. The only way for the player to escape the dungeon is to use OOP skills to “hack” the environment.

1.1 Purpose

The purposes of PolyMorpher are:

- Teach a basic understanding of abstraction and then have the user show the learned skill through application
- Teach a basic understanding of encapsulation and then have the user show the learned skill through application
- Teach a basic understanding of inheritance and then have the user show the learned skill through application
- Teach a basic understanding of polymorphism and then have the user show the learned skill through application
- Help the user develop problem solving skills
- Create a fun gaming experience for the user along with their learning experience

These purposes will be fulfilled through the player having to use various OOP skills to successfully escape the dungeon. This will be done through what will be referred to has hacking their environment. The player will be put through various puzzles where they will need to code scripts that require an understanding of encapsulation, abstraction, inheritance and polymorphism. These concepts will be taught to the player prior to their specific puzzles, allowing the player to learn and then apply these skills in a gaming environment. It is important to note the PolyMorpher will not be teaching basic syntax and that players should be familiar with common concepts of programming languages such as types, functions, loops, and what an object is.

1.2 Scope

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 concepts and problem-solving skills. Object Oriented Programming is an essential skill for students to learn for them to do well in advanced level CS courses. However, at entry level courses students do not get taught OOP concepts which can lead not the result of changing majors or leaving college entirely.

The prototype of PolyMorpher will contain an example level for each of the topics that will be taught in the complete version of PolyMorpher including abstraction, polymorphism, encapsulation, and polymorphism. Additionally, a tutorial will be included to teach the player the basic mechanics of how to play PolyMorpher as well as basic concepts the player will need to know in order to create scripts throughout the game. The topics included in the tutorial of game development be the game loop structure, triggers, and how to reference objects they see in the game in their scripts. The scripts the player creates will be attached to objects in the game which allows the player to “hack” their environment through coding.

PolyMorpher will be optimized for a 4th gen i3 Intel Processor. The minimal operating system needed for PolyMorpher will be Windows 7. These are the only two requirements needed as PolyMorpher will be a 2D game requiring minimal processing power or graphic processing power. Customers will be able to obtain PolyMorpher by going to Team Silver’s website, http://www.cs.odu.edu/~411silver/, and downloading the .exe file under the downloads tab.

1.3 Definitions, Acronyms, and Abbreviations

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

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

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

1.4 References

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

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:%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%220B5KdQEdqLUPWnNoSHhIUGg2OTQ%22%5D,%22action%22:%22open%22,%22userId%22:%22108692003133590583047%22%7D#G0B5KdQEdqLUPWnNoSHhIUGg2OTQ

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:%2210869200313359058304 7%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

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

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

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-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

Team Silver. (2018, February 26). Lab 1. In docs.google.com.

“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

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/

1.5 Overview

This product specification provides the architecture description, functional description, external interfaces, capabilities and features of the PolyMorpher prototype. The following sections of this document include a detailed description of the software and external interface architecture of the PolyMorpher prototype; the key features of the prototype; the parameters used to control, manage, or establish said feature, and the performance characteristics of that feature in terms of output, displays, and user interaction.

2 General Description

The goal of the PolyMorpher prototype is to provide insight into what the completed version of PolyMorpher would be like. The prototype will contain all the required pieces to make the final product but will not teach each topic as thoroughly as the final product will. It will however be playable from start to finish, giving players an understanding of the future product to come.

2.1 Prototype Architecture

The prototype of PolyMorpher will have a similar architecture to the final product of PolyMorpher. Figure 1 contains the architecture of the PolyMorpher prototype. As shown, the user will interact with a downloadable executable of the game, which they will download from Team Silver's website. The three main components of the prototype architecture are:

- PolyMorpher executable file which contains the game itself
- Unity file system containing the compiler as well as key files that work with the compiler that have been integrated into PolyMorpher to allow the player to code inside the game

- Team Silver Website which contains the executable file as well as will contain a forum, tips, and guides to complete PolyMorpher in the final product.

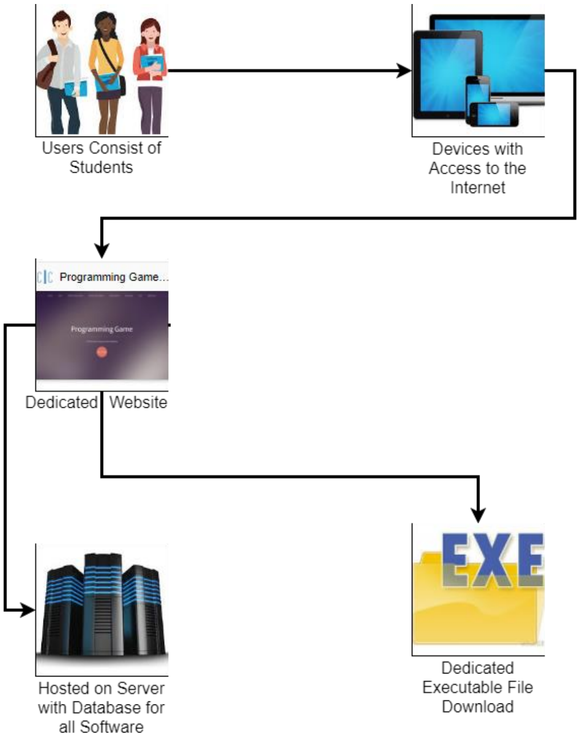
Users Consist of Students
Devices with Access to the Internet
Programming Game...
Programming Game
Dedicated Website
EXE
Hosted on Server with Database for all Software
Dedicated Executable File Download

Figure 1. PolyMorpher Architecture.

2.2 Prototype Functional Description

The functions contained within the prototype can best be explained in Tables 1 and 2. Together, Table 2 describe what will be in the PolyMorpher prototype and what has been removed from the PolyMorpher prototype.

KEY
🟩 Fully Functional
🟨 Partially Functional
🟥 Eliminated

Table 1. Key to Table 2

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)	🟩	🟩

Table 2. Summary of Prototype Functions and Capabilities

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. Summary of Prototype Functions and Capabilities

As shown in Table 2, the .exe will contain the PolyMorpher game itself. The game will teach abstraction, inheritance, polymorphism, and encapsulation. Each topic will have one level in the PolyMorpher prototype to demonstrate what a potential level teaching each topic would look like. Due to each topic only having one level attributed to it in the prototype, each topic will only have one or two aspects taught.

The Unity file structure inside of PolyMorpher contains the compiler. The compiler will be used to allow the player to develop scripts, attach them to objects in the game, and have the game compile the player's scripts while running the game. It is this process which will be used to teach the player how to use the various topics of OOP they learn throughout the game.

For the prototype, the only thing the website will be used for is to download the .exe. If time permits there will also be a sandbox level where the player will be given free reign to develop as they please with all objects in the game.

2.3 External Interfaces

The different external interfaces are the hardware interface, the software interface, the computer screen, mouse, and keyboard. The player will interact with these various interfaces in order to play PolyMorpher.

2.3.1 Hardware Interface

The PolyMorpher prototype will be played on the player's local machine. The machine must have at least a fourth generation i3 Intel processor. There are no other requirements as PolyMorpher is a 2D game, therefore requiring minimal resources from the PC.

2.3.2 Software Interface

Due to the PolyMorpher prototype being a standalone executable file, the only software interface will be the operating system that PolyMorpher runs on. PolyMorpher will be able to run on Windows, macOS, as well as Linux.

2.3.3 User Interfaces

There are three primary user interfaces that will be used while playing PolyMorpher which include:

- Computer screen: The computer screen will be used to show the player the game. The player will be given information via the computer screen. This can be any computer screen due to PolyMorpher having minimal resource requirements.
- Computer mouse: The computer mouse will be used by the player in order to decide which objects to "hack", to click on a variety of buttons, as well as to move from one level to the next. There are no specific requirements to which computer mouse must be used to play PolyMorpher.

- Computer keyboard: The computer keyboard will be used by the player to create scripts and to control the main character. There are no specific requirements to which computer keyboard must be used to play PolyMorpher.