Lab 1 Collaborative Outline

Team Silver Descriptive Paper Outline

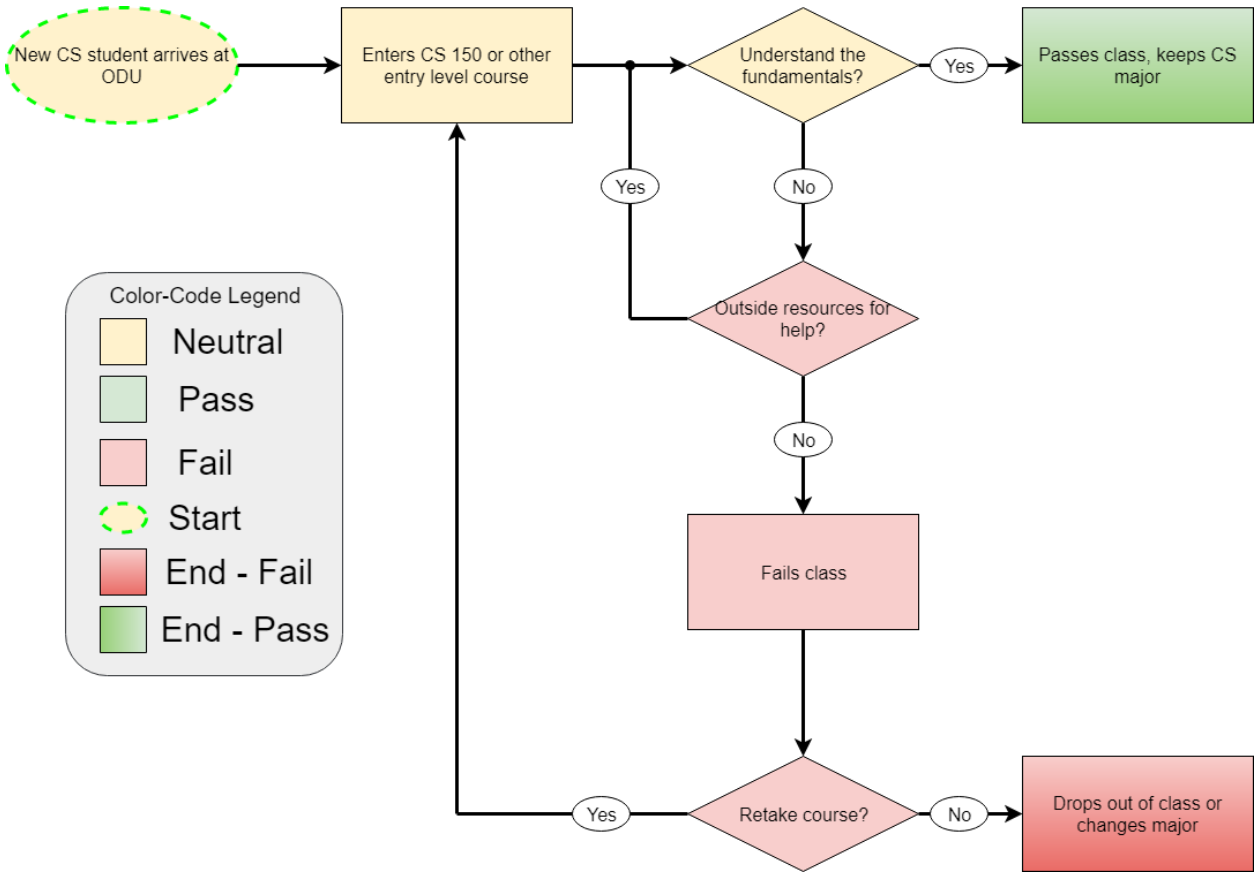
[Insert Table of Contents and Figures/Tables First]

1 Introduction

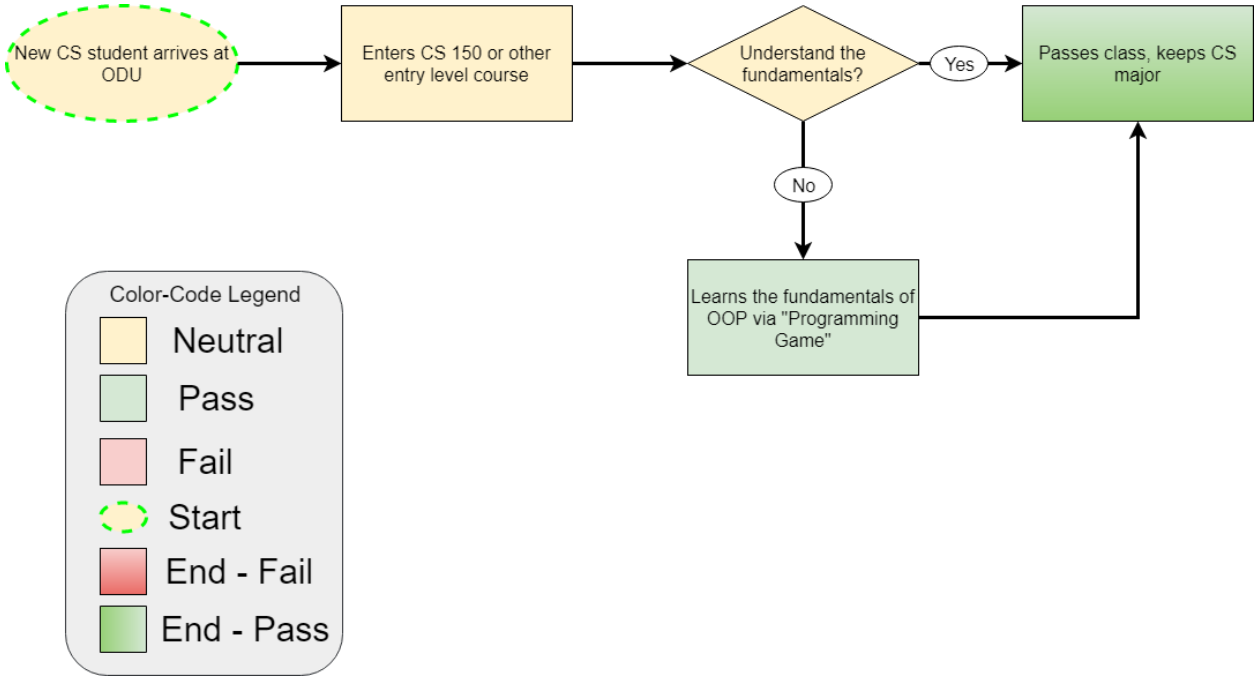
1.1 Team Members

1.2 Problem Statement

1.3 Problem Characteristics



1.4 Solution Characteristics



[Insert Current Solutions (Competition) After Introduction

1. Strengths
2. Weaknesses

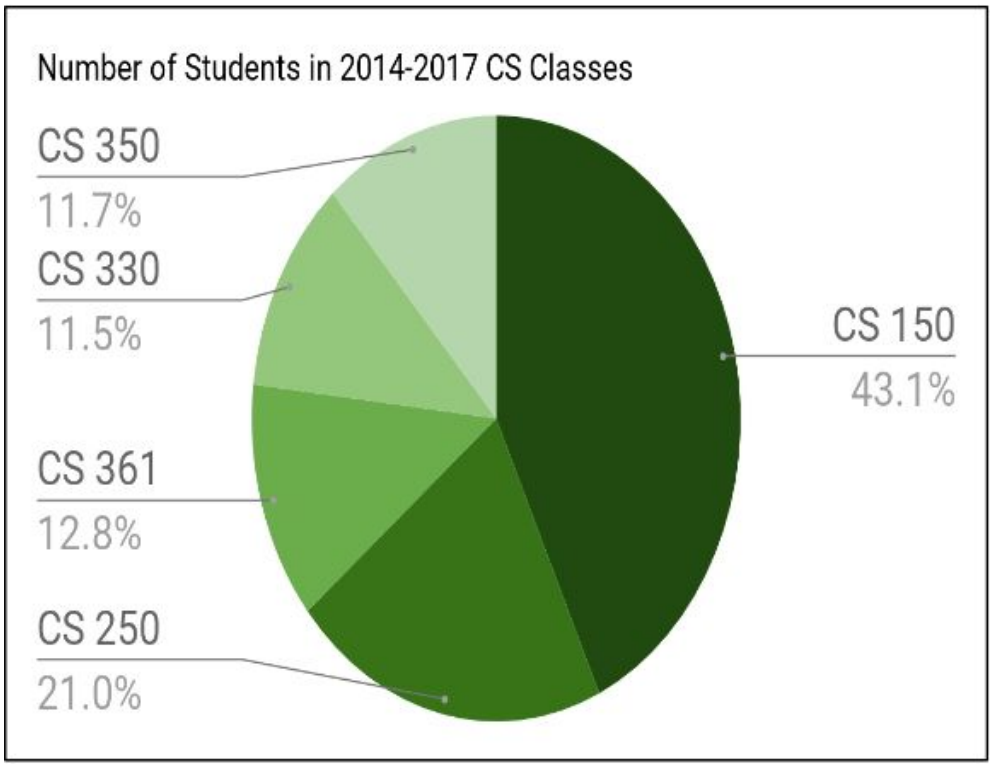
Game	Experience	Uses OOP	Teaches OOP	# Languages	Multiplayer
PolyMorpher	Low-Mid	Yes	Yes	1	No
Git Games	Low	No	No	1	No
CSS Diner	Low	No	No	1	No
Flexbox Defense	Low-Mid	No	No	1	No
Ruby Warrior	Low	No	No	1	No
Untrusted	Mid-High	No	No	1	No
Empire of Code	Low-Mid	Yes	No	2	Yes
Ruby Quiz	Mid-High	Yes	No	1	No

2 PolyMorpher Product Description

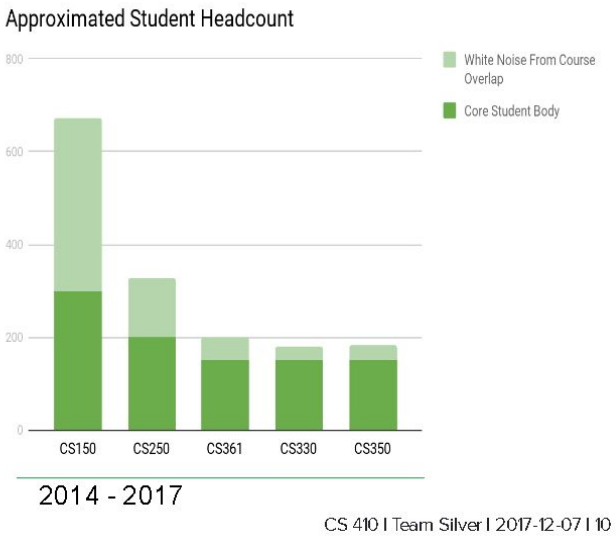
2.1 Goals and Objectives

2.2 Key Product Features and Capabilities

2.2.1 Game concept (Basic idea of the game)



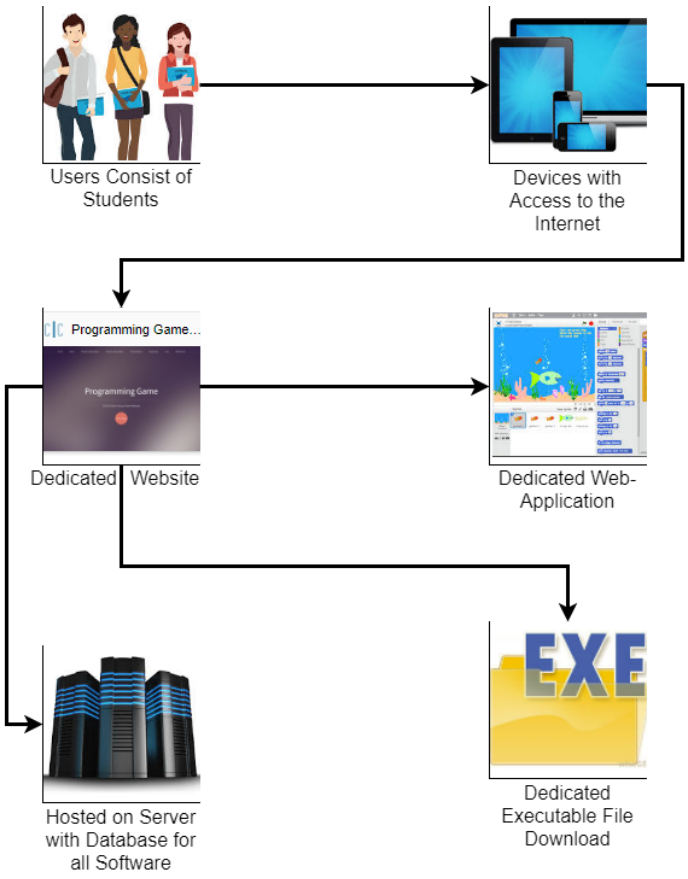
	CS 150	CS 250	CS 361	CS 330	CS 350
2013-2014	804	327	161	111	93
2014-2015	672	367	208	203	148
2015-2016	937	327	217	195	183
2016-2017	920	337	199	180	182



4 Product Prototype Description

4.1 Prototype Architecture (Hardware/Software)

4.1.1 Prototype will contain .exe file

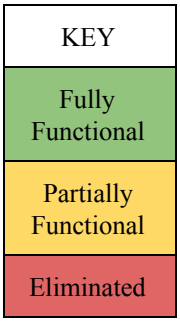


4.2 Prototype Features and Capabilities

4.2.1 Prototype Deliverable Features

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		



4.2.1.1 teaches polymorphism

4.2.1.2 teaches abstraction

4.2.1.3 teaches encapsulation

4.2.1.4 teaches inheritance

4.2.1.5 Single language taught

4.3.1.6 Single Player

4.2.2 Fully Functional Components

4.2.2.1 Game Assets

4.2.2.1.1 Developed Story

4.2.2.2 Portable Compiler

4.2.2.3 Scenes that teach polymorphism

4.2.2.4 Scenes that teach abstraction

4.2.2.5 Scenes that teach encapsulation

4.2.2.6 Scenes that teach inheritance

4.2.2.7 Tutorial section

4.2.3 Partially/Maybe Functional Components

4.2.3.1 Sandbox Level

4.2.4 Eliminated Capabilities

4.2.4.1 Multiplayer

4.2.4.1.1 Why? Security.

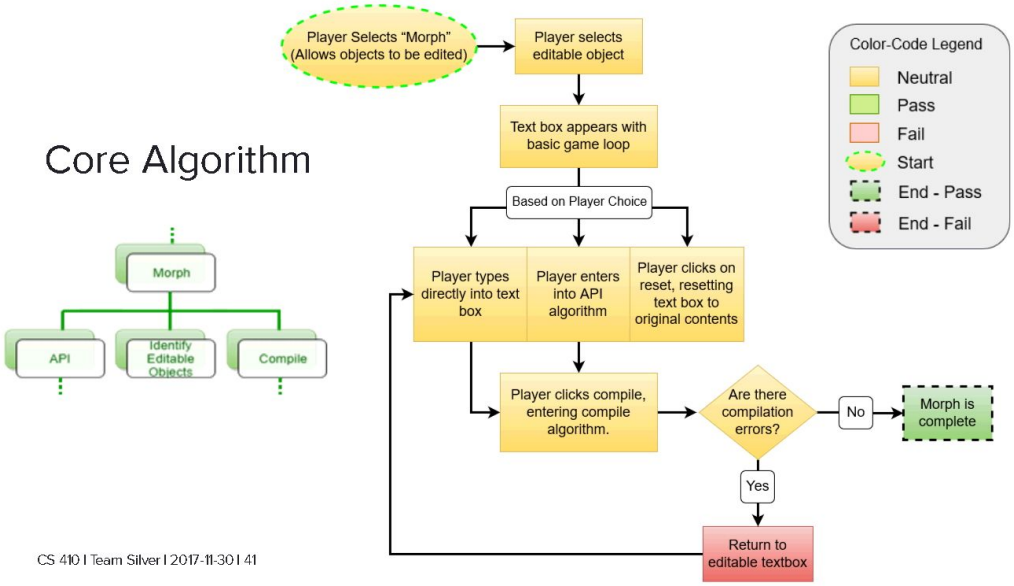
4.2.4.2 Web Application

4.2.4.2.1 Server maintenance

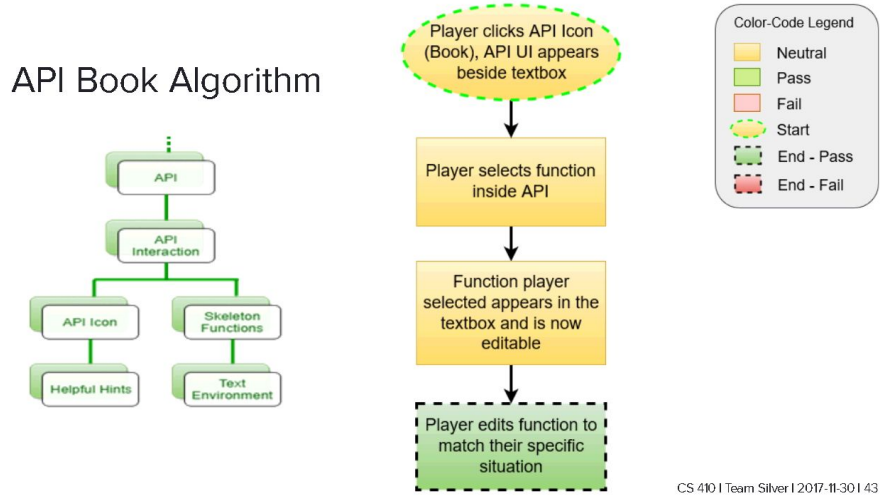
4.2.4.2.2 Portable Compiler wouldn't work

4.2.5 Algorithms

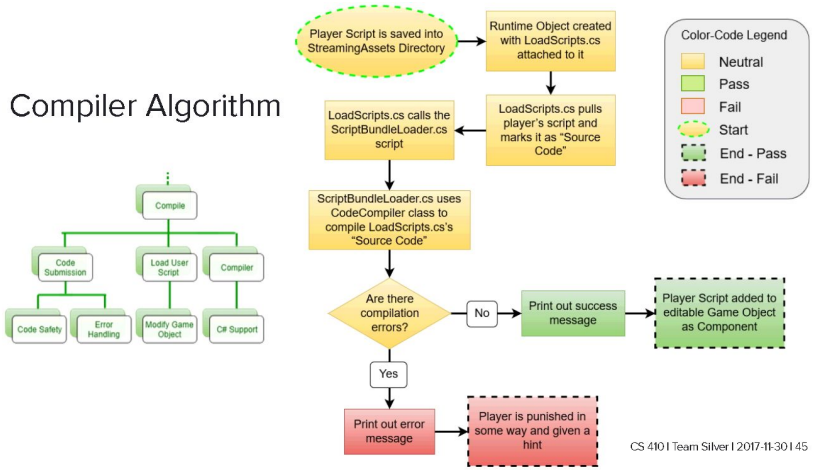
4.2.5.1 Core Algorithms



4.2.5.2 API Algorithms



4.2.5.3 Compiler Algorithms



4.3 Prototype Development Challenges

4.3.1 Design continuity

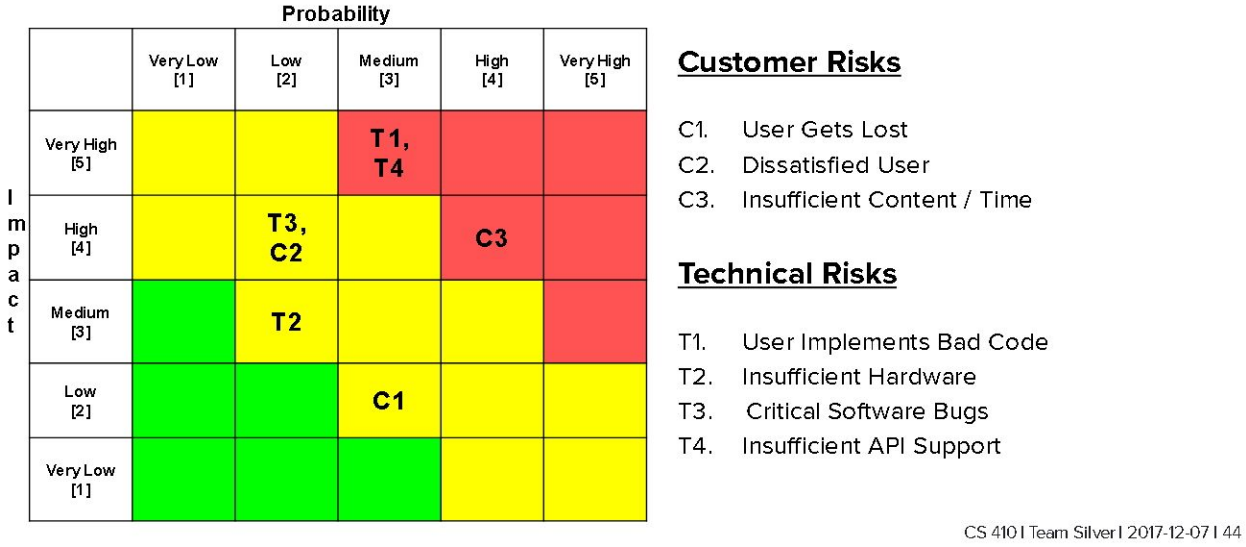
4.3.2 Difficulty debugging

4.3.3 Playtesting

4.3.4 Maintaining player engagement

4..3.5 Teaching enough

4.4 Risk Mitigation/Risk Matrix



4.4.1 Technical risks with mitigation

4.4.2 Consumer risks with mitigation

5 Development Pipeline

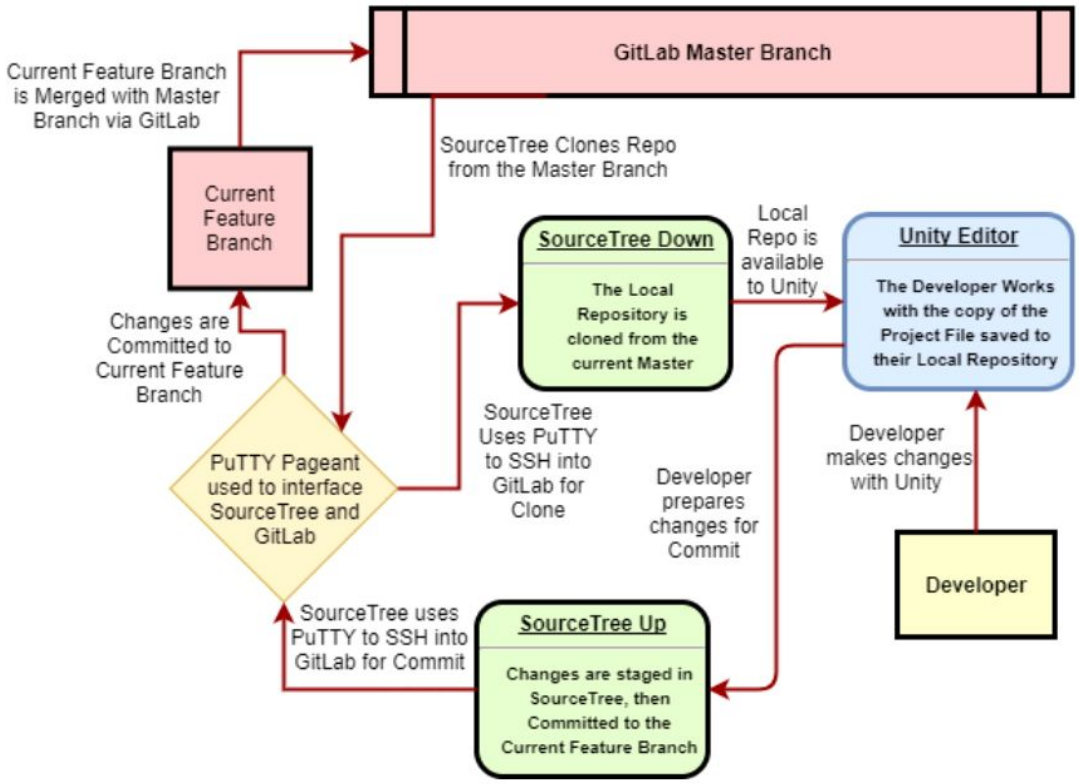
5.1 Unity Software

5.1.1 C#

5.1.2 Monodevelop

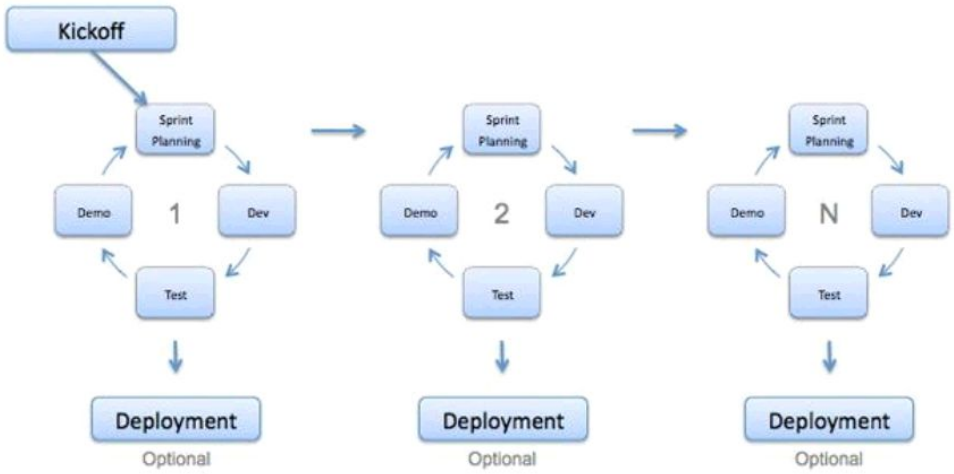
5.2 Source Tree Software

5.3 Version Control



5.3.1 GitLab

5.4 Agile Development



5.5 Work Management

5.5.1 Groups

5.5.2 Topics

6 Glossary

Streaming Assets Directory: File directory where all scripts accessible to the player via the in-game Coding Interface are stored and organized according to level and programming concept relevance. It is unique in that it is one of the few source file directories that are accessible to the player in the Unity Engine under any condition.

Code Compiler Directory: File directory holding the portable compiler, and the associated companion files, which are used to manipulate the scripts in the Streaming Assets Directory.

LoadScripts.cs: Manages files accessed by the entire portable compilation system by identifying which script is currently in focus as the “source” script for any selected object in game, pulling this file from the Streaming Assets Directory and passing it off to the Script Bundle Loader for compilation.

ScriptBundleLoader.cs: Takes scripts passed off from the Load Script file and marks them for compilation, setting up the Assembler and Compiler and running the selected script through them. In the event of any compilation errors it will send out an error report through the Unity Error and Log files, otherwise it will attach the script to whichever game object was selected.

Coding Interface: An in-game GUI accessible to the player that pulls specified scripts from the Streaming Assets Directory for them to edit and compile using the portable compiler from the Code Compiler Directory.

7 References

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

"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

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

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/

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