

Table of Contents

Project Description

Process Followed by Team

Requirements & Specifications

Architecture & Design

Future Plans

Project Description

Our project was to create an action RPG video game. The goal of the game is to explore the world searching for objects, fighting various enemies during the process. Once all of the objects are collected, a final boss will appear, which must be defeated to complete the game.

The game was created using Python 2.7.6 and the Pygame library. This allowed us to make the game cross-platform.

Process Followed by Team

Issues of Iterative Development

The biggest issue we experienced with iterative development was that we assigned multiple tasks that depended on each other to the same milestone. This meant that if one of those tasks took longer than expected to complete, then all of the other tasks would also get behind schedule. This typically resulted in all of the dependent tasks getting pushed to later milestones.

A second issue we faced was that we initially had no idea how long certain tasks would take to accomplish. This meant that many of our earlier time estimates were inaccurate, and it caused us to shuffle around the tasks assigned to certain milestones. This issue resolved itself as we became more familiar with Pygame.

Refactoring

Testing

Testing was a continuous process throughout the process. One place testing took place was during the code review process. When a developer reviews a branch, they test it on their local machine to ensure that it works properly and that there are no noticeable regressions. We also had a set of automated unit tests that helped us test the functionality of our code in a consistent and automated fashion. In addition to this, we had a set of Lettuce tests for testing the behavior of the code. These tests allowed us to use a plain English description of the behavior of a section of code, and then have the tests run and produce a nice clear output. Finally, we had a script providing code coverage metrics to help guide us in writing our tests and verify that we were testing all of our code.

Collaborative Development

In Extreme Programming, it is typical to pair program with another developer as much as possible. As students, this proved very difficult in CS 427 due to all of us having very different work schedules. As a result, we decided to drop pair programming in favor of code review. This means that all development work is performed in separate git branches. When the changes are functioning properly and ready to be merged into master, a pull request is opened. At this point, one or more other developers review the code changes and test out the branch. This provides us with the benefit of having multiple sets of eyes reviewing all code, but does not require us all to have the same work schedule.

Requirements & Specifications

We successfully implemented all of our user stories for this project. At this point, we have a functioning game where the player can move around the world collecting items and killing enemies before battling the boss to win the game.

Architecture & Design

Future Plans

We currently have no plans to continue development of this project after the class ends. This project was meant to be a learning experience for us, and it allowed us to experiment with Pygame. We will make the source code, documentation, and all associated resources available to the public under a XXX license, which will allow any interested parties to continue development. We will also probably use the skills and knowledge gained from this project to help us develop additional games in the future.

Nathan Handler

This was my first time developing a game. I quickly learned that it has several challenges not present in traditional application development. First, the user interface plays a very important role. When developing a tool, functionality tends to be more important than a pretty interface. With a game, an ugly or unusable interface makes the game useless.

Kyle Nusbaum

Jonathan Albrecht

Andrew Roth

Ian Canaday