Baskin Engineering
UC SANTA CRUZ

Unity Optics

July 18, 2019

Team Unity: Jason He, Minsu Jang, Shealtiel Mulder, Matthew Rhea (Product Owner), Baoqing Xie

Introduction

- Increasingly, game developers rely on different forms of digital monetization to provide games as a service (subscriptions, in-game stores, advertisements, etc).
- Of these, advertisements have been integral to funding these games.
- We wanted to experiment with the telemetry that can be derived from in-game advertisements.
- Unity Optics sought to build a plugin for in-game advertisements that would record the actions taken by players when encountered by these advertisements while playing.

High Level Goals

- Goals We Set
  - A Unity plugin that provides 3D-based advertisement tools for in-game monetization:
    - Records players' interactions with advertisements.
    - Provide easy access to a database of telemetry in order to perform visual analysis of this data.
  - A sample webpage that visualizes the recorded telemetry in real-time as a player plays the game:
    - Update data for specific advertisements in real-time
    - Static and interactive visualizations update alongside the data itself

High Level Goals

- Goals we achieved
  - A Unity plugin that records player interactions with sample advertisement objects in-game, and stores this on an online database.
  - Static and interactive visualizations of the data that was collected
  - A simple webpage that displays an exported Unity game and presents the data as you play.

Biggest Challenges and Accomplishments

Challenges

- Resolving conflicts in member's daily class and work schedules
- Integrating testing into the development process
- Overcoming the learning curve of new technologies
- Short sprints led to rushed development
- Continuous Integration with Unity Engine felt impossible

Accomplishments

- A mostly realized product based on the original specification
-

System Demo

Baskin Engineering
UC SANTA CRUZ

System Overview

Baskin Engineering
UC SANTA CRUZ

Technologies

Game Development IDE: Unity Engine (C#, Firebase for Unity)

Database: Google Firebase

Automated Testing: Travis-CI

Web: Firebase for Web (HTML, Javascript)

SCRUM: Trello for scrum board, Google Sheets for burnup charts

Project Management Techniques

Agile Scrum Practices

- Daily Standup
- SCRUM Board
- Three 1-week Sprints
- User Stories
- Sprint Reviews and Retrospectives
- Burn Up Charts

Other Management Techniques

- Management Tools: GitHub, Trello, Google Sheets, Slack

Things We Enjoyed

- Working with Unity
- Creating a game demo for the plugin
- The organization of Agile Development made software development feel more like an engineering practice
- Learning how other technology integrates into the Unity Engine

Things We Did Not Enjoy

- Time constraint of 5-week session made the project feel rushed and incomplete
- Many user stories and tasks had to be changed or re-adjusted to fit technology or time limitations
- Difficult to set up a time for daily meetings that works for everyone, so we often did not have the full team present

Lessons Learned

What Worked

- Frequent meetings
- Working with a group
- Having a larger project so everyone works on different parts instead of one part together
-

What Did Not Work

- Remote meetings
- Working alone
- Communicating exactly what each team member is doing

Lessons Learned

What We Wished We Had Done

- More efficient division of work
- A more detailed SCRUM board
- Better documentation from the beginning of the project
- Figure out how to make proper Burn Up Charts earlier

Baskin Engineering
UC SANTA CRUZ

Questions?