Lowering Development Barriers in Educational Game Design

Austin Bart, Robert Deaton, Eric McGinnis
Virginia Tech, University of Delaware

Introduction

[Sentence on video games as a useful tool in both education and in computer science education] [Introduce CISC374 as a course which aims to combine both of these goals] [Sentence on the emphasis on educational game development, working with CCCS] [Elaborate on CCCS, talk about their income level, the donation of XOs] [Pivot back to the use as an educational tool in software engineering for college juniors/seniors] To help this course achieve its goals, we have developed a set of tools to help address some difficulties and limitations in the tools previously available.

Development Platform

The primary development platform is the OLPC XO, as they are the most readily available platform at the target middle school. To allow for rapid development in one semester, and because it is the favored language on the OLPC XO, Python was chosen as the language to be used in the class. Lastly, because it is preinstalled and because the lack of hardware support in the operating system eliminates most other options, Pygame was chosen as the graphics library to be used in the course.

Spyral

Spyral is a 2D sprite-based engine [built on top of pygame?] designed to allow the rapid development of games, particularly those targetted at low performance platforms. It started as a library to optimize drawing during an early semester of the course, and grew to include many more features that previously had to be written by each team individually. [Perhaps a note on why we have to use pygame?] Spyral now includes modules to handle:

- Scenes, for game organization
- Images, for loading and drawing graphics in a consistent way
- Sprites
- Fonts
- Event Handling
- Vectors
- Rectangles
- Animations
- Clock

In addition to providing tools for making game development easier, spyral also works to teach and encourage better software engineering practices to the users, targetting places where poor decisions were routinely made in the past. In particular, the use of a scene system encourages the separation of content into disjoint pieces where possible, and the event system is designed to mimick and be used like modern event driven systems for games, net-

Conspyre

Conspyre is a cloud-based networking system built to work in conjunction with Spyral. The system is designed with two parts: (a) a client library for XO games that talks to (b) a web framework that can store and retrieve data and provides a portal for teachers. Using this sytem, XO developers can persist data between students' play sessions and enable communication between teachers and students. As it is used primarily by novice developers with limited experience, Conspyre is written in Python and built on a scaffolding paradigm by which students can quickly develop functional applications with a minimal knowledge of web development.

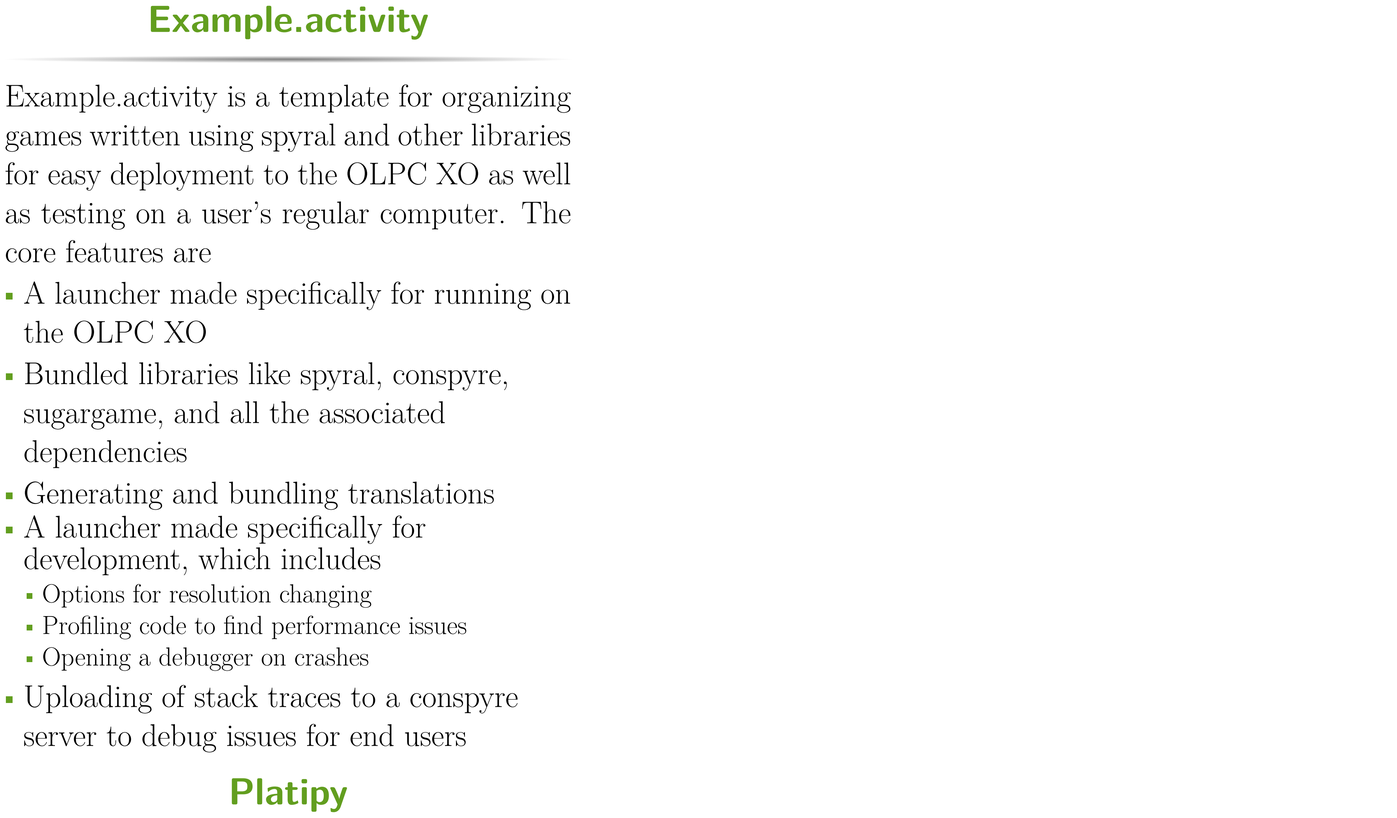
Example.activity

Example.activity is a template for organizing games written using spyral and other libraries for easy deployment to the OLPC XO as well as testing on a user's regular computer. The core features are

- A launcher made specifically for running on the OLPC XO
- Bundled libraries like spyral, conspyre, sugargame, and all the associated dependencies
- Generating and bundling translations
- A launcher made specifically for development, which includes
  - Options for resolution changing
  - Profiling code to find performance issues
  - Opening a debugger on crashes
- Uploading of stack traces to a conspyre server to debug issues for end users

Platipy

Platipy is the misspelled plural of platipus.



Experiment

This is the experiment.