

Vecset for the XO

By Mike Hahn

Copyright © 2008 Vecset.net

Date Last Edited: 29-Oct-2008

Table of Contents

Vecset for the XO	1
<u>Introduction</u>	<u>1</u>
<u>Code Editor</u>	<u>1</u>
<u>Board/Level Editors</u>	<u>1</u>
<u>Animated Games</u>	<u>2</u>
<u>Cross-Compiler</u>	<u>2</u>
<u>Python Collaborator</u>	<u>3</u>

Vecset for the XO

Introduction

[[Home](#)]

Vecset for the XO Laptop is written in Python, and uses Pygame. The first version of Vecset for the XO will support the development of board games. The second version will support the development of animated games. Vecset for the XO differs from Vecset for Windows in that its built-in scripting language is Python, not Vecscript, and it does not support playing games over the Internet (inside a web browser). Instead, XO users can play games with other nearby XO users using the XO's wireless networking.

Drag-and-drop board games are constructed out of 5 basic components: 1) Card, such as a playing card or chess piece; 2) CardStack, a stack of Card objects; 3) BoardGrid, such as a chess or Go board; 4) RackGrid, a collection of CardStack objects arranged in a row (or column); and 5) TableGrid, a more flexible version of a BoardGrid object (not just a plain grid), such as a Monopoly board.

Helping the Third World

In order to thank the OLPC community for helping in the development of the XO version of Vecset, vecsworld.com pledges to donate one dollar from the sale of every \$24 annual membership fee to One Laptop Per Child. The XO version of Vecset is of course free for all XO users.

Code Editor

The code editor is used to edit the game designer's Python code. This Python code consists mainly of event handlers connected to board game components. The game designer uses the Board Editor to layout the board, and connect event handlers to the board's components. One of the features of the code editor is syntax highlighting, in which different code elements (keywords, identifiers, operators, etc.) are displayed in different colors.

Board/Level Editors

The board editor is used to layout the board, and connect event handlers to the board's components. The level editor is used to map out each level in an animated game. In general, a level in an animated game consists of various rooms, and each room may contain objects (called "vectors") and/or characters. A character can be either a human player or a non-player character (NPC). The vector editor is used to design vectors, which may be animated or static.

Animated Games

The default user interface is 2-dimensional, consisting of 2 windows side-by-side (split screen). One window is always an overhead view, and the other window is split into 2 windows, one on top of the other: left-side/right-side, or front/rear. All 3 windows are separated by splitters, so it's easy to resize them. Clicking on the point of intersection in the center of the screen resets the sizes of the 3 windows to their default values.

User Commands

The user uses the cursor keys to go forward (up arrow), left, right, or backwards (down arrow). Ctrl+Left Arrow and Ctrl+Right Arrow rotates the overhead view 90 degrees in the given direction. Clicking on an object carries out the default primary user action, and right-clicking on an object brings up a popup menu of choices. Pressing Tab highlights the next command in an onscreen menu. Clicking in the non-overhead view toggles between left/right and front/rear. Right-clicking in the non-overhead view and selecting Backpack (or typing Ctrl+B) displays contents of backpack in the window clicked upon.

Cross-Compiler

A cross-compiler will be developed to translate games written in Vecscript for the Java/Windows version of Vecset into Python, so they can be played on the XO Laptop. This cross-compiler will itself be written in Python for the XO Laptop.

Python Collaborator

VECSworld.com is looking to team up with a Python programmer, who will develop the XO Laptop version of Vecset, collaborating with the Java/Windows programmer, who will develop the Windows version of Vecset. Both programmers will share equally in the revenue generated by Vecset members, who elect to pay \$24/year for the privilege of having access to a private room in VECSworld. Also, Vecset game designers can elect to pay \$48/year for the privilege of locating their games anywhere in VECSworld, and being able to customize their game rooms.

While the Python programmer is developing Vecset, the Windows programmer will be developing the Vecscript compiler (Vecscript is the built-in scripting language included in the Windows version of Vecset). When the compiler is finished, the Windows programmer will translate relevant parts of the Python programmer's code into Java for the Windows version of Vecset.

Remuneration

If Vecset is to be a success financially, the projected user base is let's say 5000 users, at a subscriber rate of say 20 percent, or 1000 members, and let's say an additional 200 game designers who are also members. So the gross income, not including advertising revenue, equals $24 \times 1000 + 48 \times 200 = \text{approx. } \$34,000$ per year. Let's say the average time spent online per user equals 2 hours/week. That's 10,000 user-hours per week. Dividing by $24 \times 7 = 168$ hours in a week = approx. 60 users logged on at any one time.

The Python Collaborator will be paid 50 percent of the net income of VECSworld.com. In the above example, that would be \$17,000 per year less 50 percent of VECSworld.com expenses. That's just an estimate of course, the actual amount may be less (or if Vecset is a runaway success, it could be a lot more).

VECSET for the XO: Block Diagram

