Kevin Cooper

Maj. Ross

12 Dec 13

CS 359

PEX 4

Team Members:

1. Kevin Cooper – As the lead project designer and programmer, he contributed most, if not all, of the work to complete this project successfully and on time.

User Manual:

1. The networkx library must be installed and in the python path
   1. This was accomplished on my system by using `easy\_install networkx`
   2. Verify that PyDev eclipse adds the new package to the path
2. To run the game, verify that all the files are together and run RiskGameMain.py
   1. The game might pause for a second as the music loads. It is not crashing.
3. Select "Start Game" on the main menu
   1. Player 1 is Red
   2. Player 2 is Blue
4. Controls
   1. Left click an allied cell then an enemy cell to attack
   2. Left click an allied cell then another allied cell to move units
   3. Right click an allied cell to place units there
   4. Press the space bar to end turn and switch players
   5. Press "h" for options
   6. If all else fails, hit "ESC" key for close the program

Release Notes:

* Failed Original Features:
  + Region based gameplay
    - Irregular shapes
    - Circles used instead
  + Only Two Players
  + Cards/Events
    - Implementation of design did not permit for an easy interface with what the cards would do.
* Features Not Included in Original Plan:
  + Sick Menu
  + Awesome Music System
* Not Tested
  + RiskGameMain.py
    - The main file, so does not implement anything but the music, which is in a try/except. Since everything else is in main, there is nothing to test.
  + MenuStuff.py
    - All the feedback from menu stuff comes from events. These events are generated by user interaction on the menu GUI, so not possible to test with pyunit framework.
  + AnimatedSprite.py
    - Since the only thing changing is the image being rendered on the screen. It would not be possible to test using the pyunit test framework
* Known Bugs:
  + None
  + Please report bug to https://github.com/KevinCooper/RiskGame/

Design Pattern:

A custom iterator was built for the GameBoard class. This allows for loops to loop through the regions simply, even though the data structure storing the information is a dictionary. If it has not been implemented, the getRegions() method would need to be called, and the reference of the result would be in the form region[1]. It performs this by converting the dictionary into a list, then returns each of the region objects in the list through the next() call.

Documentation:

1. Animated Sprite Class
   1. <http://shinylittlething.com/2009/07/21/pygame-and-animated-sprites/>
   2. The following tutorial was going over how sprites could be animated by switching through the images as a certain fps. I use a slightly modified version of his Animated Sprite class in order to simplify the explosions used in my project. This class determines whether enough time has passed based on the last time it was updated. If it has been enough time, it increments the internal counter of which picture should be rendered.
2. Simplified OOP Menu Class
   1. <https://github.com/redw0lf/scriptersCornerGaming/tree/master/examples/GameMenu/src>
   2. This class works by instantiating the desired options as its own menu item object. The main menu then determines where they should be located based on how many there are. When an event occurs, it determines which menu item object was clicked returns both a index item and the text of the item that was clicked.
3. NetworkX Library
   1. <http://networkx.github.io/>
   2. This is an extensive package that allows for the creation os network using several different data structures. The data structure used in this project was the Graph. Each element on the board was added to the graph as a node, and each connection added as an edge
4. PEP8.py, included in Eclipse
   1. A python style checker that was used to ensure no line went over 80 character
5. Doxygen + GraphViz
   1. Used to convert the comments into an HTML document with class diagrams
6. FreeMusicArchive.org
   1. Being able to get music that I can actually turn in with my project