Mancala Project

Team : Puma

CS 151 – Section 1

Mike Royal

During the weeks spent working on Mancala project with my group members the experience has been interesting because I’m not use to working with a group it’s just usually me working by myself. So this project was a good experience because I know in the world some jobs involve you being able to work well with others in a group. But overall this was a good experience for me. Some material used from class was the use of frames and buttons. Implementing patterns like strategy and decorator talked about in chapter 6. And the use of arraylist which is part of other collections class that are useful for the mancala board where there will be a large variation in the amount of data that you would put into an array that keeps trace of the player’s turn as the stones move through the pockets. So some new material I learned for the project was the **AlphaComposite** **class**, which implements basic alpha compositing rules for combining source and destination colors to achieve blending and transparency effects with graphics and images. This is done when a line is drawn, a shape is filled, or an image is rendered, that item totally obscures whatever pixels it is drawn on top of. An image is always represented as a rectangular array of pixels. Sometimes, however, we want to use an image to display a nonrectangular graphic. To allow this, some image formats support the notion of a transparent color. When the image is drawn, the background shows through whatever pixels are marked as transparent. Transparency is indicated with a bit mask: for each pixel in the image, the graphics system uses one extra bit of information to specify whether the pixel is transparent or opaque. Another thing I learned was the **LineMetrics** **class** which allows you access to the metrics needed to layout characters along a line and to layout of a set of lines. A LineMetrics object encapsulates the measurement information associated with a run of text.

Alan Chin

In this project, the bulk of my time was spent writing the code and debugging the controller and programming the view and some of the model. I learned to apply decorator patterns to many of the components in the view to make it look more appealing to other people. Also I learned to write classes that implemented an interface in order to get the strategy pattern to work for each different view. I also applied the concept of an Anonymous action listener to the program for smaller one time use buttons so I wouldn’t have to create a whole new action listener class. As far as self-study concepts, I learned a lot about JComponents and how to use them properly as well as how many useful built in methods there are for the various classes. I learned how to use and implement the Observer interface as well as the Observable Class and applied it to the MVC pattern in the project. Working on the project also strengthened my abilities in using Generics to make my programming easier to read for me and others as well and reinforcing my existing knowledge on when to build a class and when to use an interface.

Jordan Jennings

During this project we utilized a number of different ideas from this project. The requirements for weekly documentation along with documentation of the code itself with diagrams and comments was an eye opening experience into the world of programming as a career. Not to mention the use of MVC and patterns along with the java swing library. Along with the strategy pattern those are topics which I had both applied and had to learn more about in self-study to complete the project. The MVC and Strategy patterns are complex topics which I learn more about every time I use them. Java is a language I have yet to be strong in also which I very often have to get reference for some of the data structures included with java, the book has of course been immensely helpful with this and the topics covered in this class.