Ludo Game

Kwabena Manu

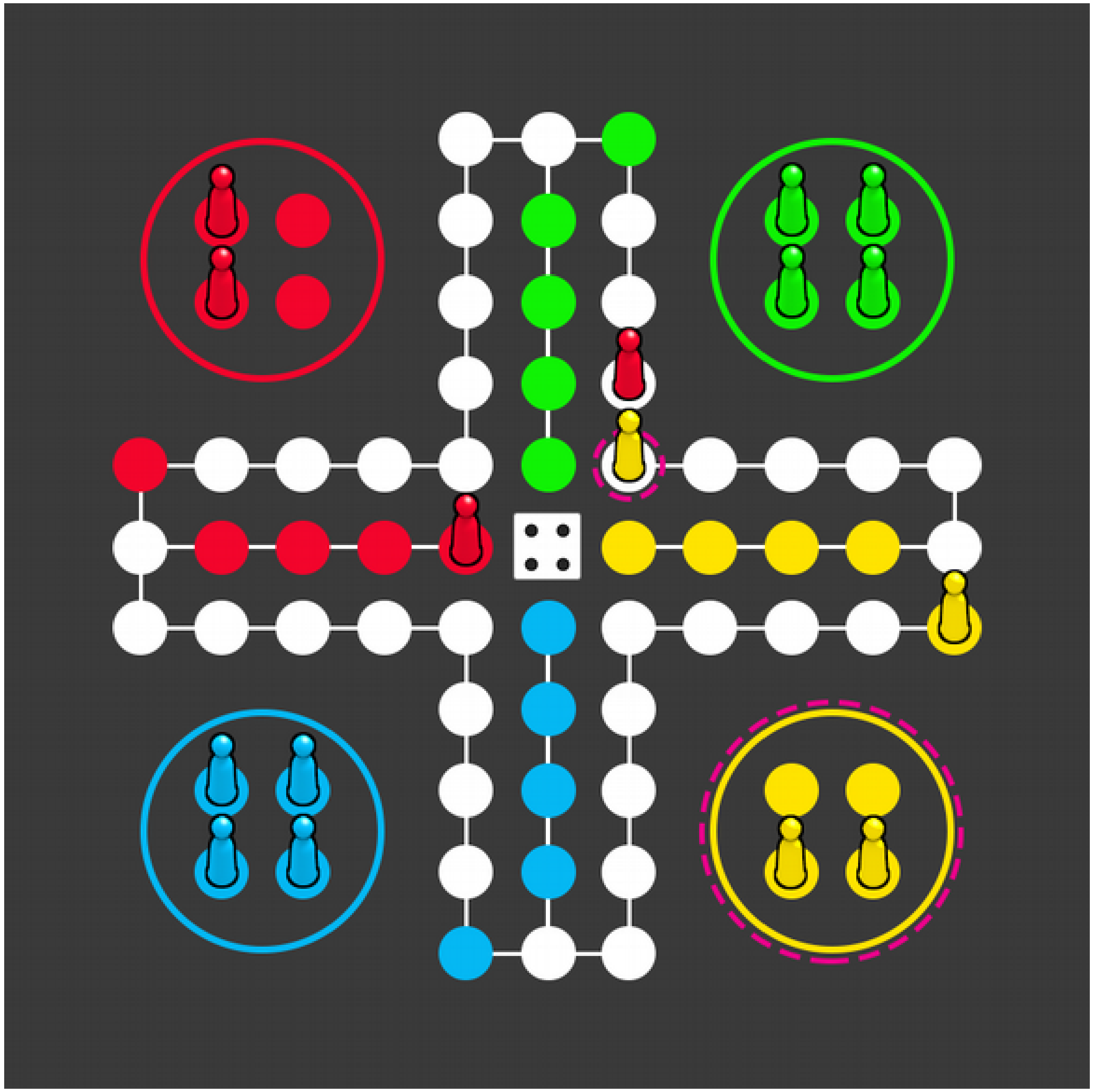
Thomas Kojo Collins.

Abstract

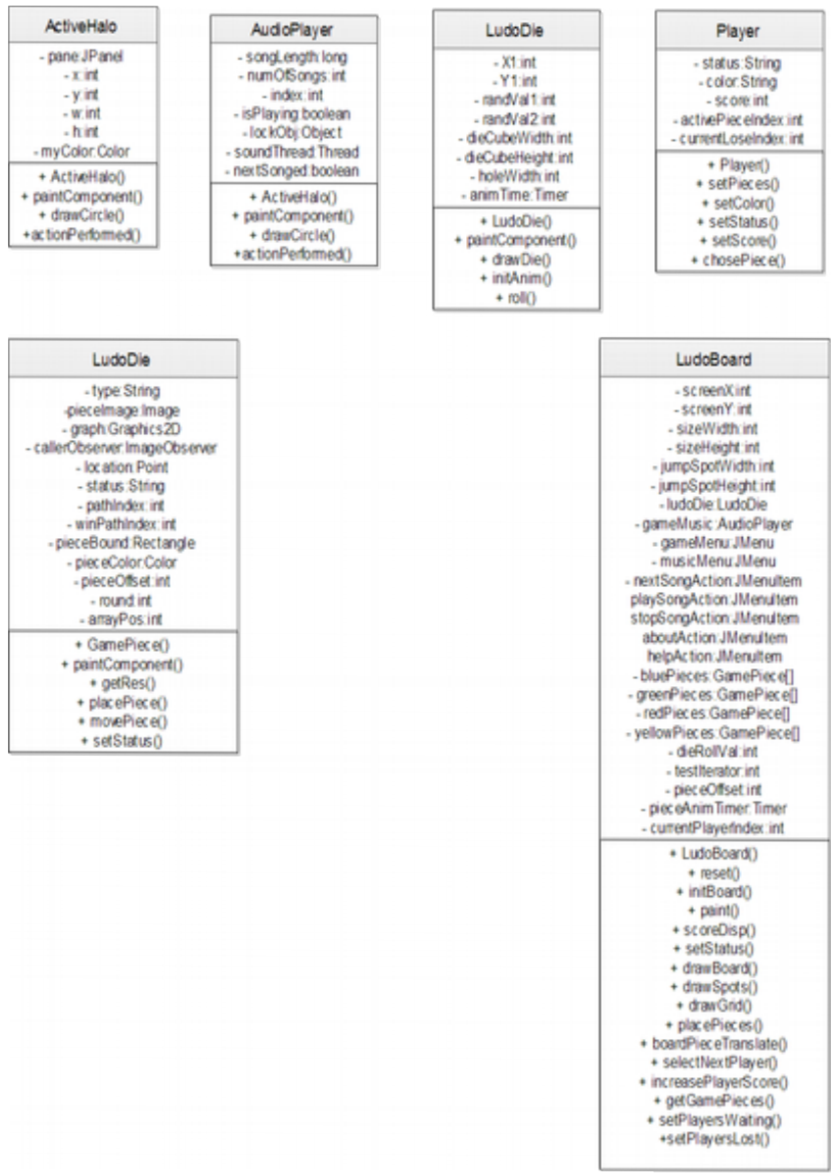
- The aim of the project was to design and construct, using the classes and tools contained within the JDK and the Eclipse IDE, a functional Ludo game.

“Using java swing design the graphics and logic of a Ludo game.The application should have the following features:Implement the game play between a user and the CPU.Four colors (Positions) from which a player can choose.Four playing pieces for each color.Implement a cell and a home for each color and link each cell to home with a path tocomplete the environment.Implement a die using random number generator. The die should have six faces eachrepresenting numbers from 1~6.Play audio tunes in the back ground.For the basic rule, a user should be able to toss the die for the number 6 before hecan have a chance to move a single playing piece into the environment, after whichever toss value of the die is used to count the piece through a path around theenvironment till it reaches home. This is repeated on every toss of 6 for each playing piece until all pieces reaches home.For every toss of die a user has to select which piece should move and in whichdirection assuming there are more than one piece on the playing environment.A user and CPU play the game in turns. Who ever moves all four pieces from cell tohome is the Winner for that game.”

Conceptual Design



UML Class Diagram



Resource creation Procedure

- The game pieces are images which were first modeled as 3D objects in Blender, a 3D development software. They were then sized appropriately in Photoshop, before being imported into the java application.
- The music used in the game are all license free songs, gotten off Youtube.

Implementation Process

- The Ludo game was implemented using the Eclipse IDE.
- The code was worked on simultaneously, and version control was handled via GitHub.

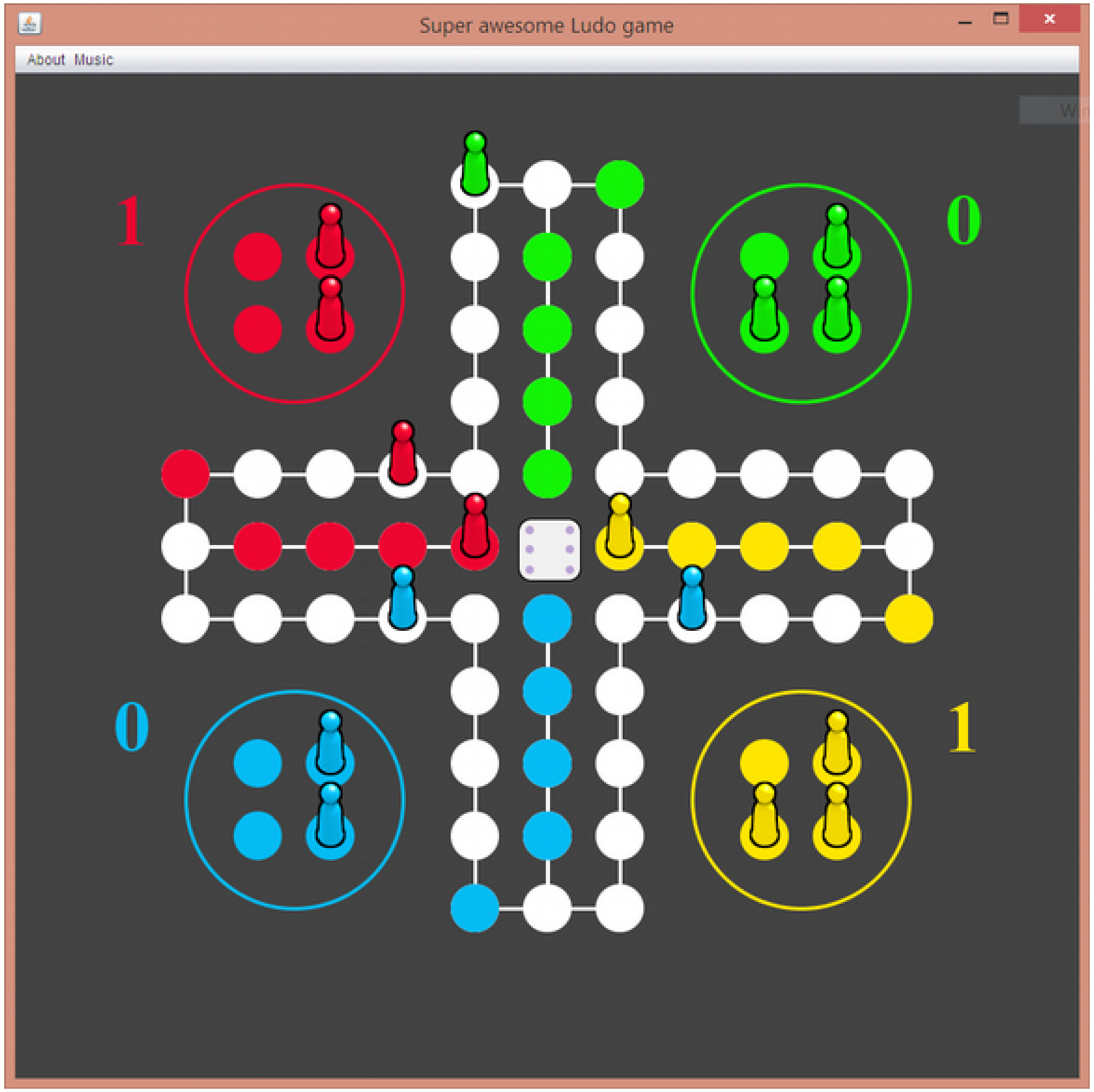
Implementation: Game Flow

- The game starts out with 4 players, each with a different colour, and each having 4 game pieces.
- There is a game path, consisting of 38 round 'jump spots' on which game pieces traverse around the board. There are also 4 win spots for each player.

Implementation: Game Flow

- The game starts by pressing the space bar to roll the die. Upon rolling, the first player, the RED player, has it's first piece moved out onto the board, and is moved the number of steps the die value is. After this piece has moved, the green player's game piece then moves, followed by the yellow, then blue. The first player to line up all 4 pieces on the win spots wins the game. Upon winning, the players are notified and the game is reset.

Implementation:



Discussion

- Due to the heavy use of the swing paint function, and given that the swing classes are generally not thread safe, there was a problem of the repaint rate not catching up to the rate at which changes were being made. This resulted in the UI appearing to be flashing as moves were made.
- The audio files used had to be converted from mp3 format, which is a compressed file format, to wav, which has relatively no compression. This greatly expanded the file sizes of the music, bloating the project size.