Design Decisions

Author: Jeremy Trendoff – 101160306

Overview

In this milestone, Team 16 was tasked with added in specialty tiles (Go, Jail, Railroads, Utilities etc), support for purchasing houses and hotels on property, and finally support for adding AI Players into the game. The design decisions for each new function are described below.

House and Hotel Support

Author: Frank Dow – 101140402

To allow the player to purchase house and hotels we have added the button to the player menu “purchases houses”. When clicked, this button calls purchaseHouse() that checks whether the current player has a set of properties or not. Unfortunately, it only allows a player to build on one set of properties from the way it was implemented. If the player does own a set of properties, they will be represented with a new GUI Panel of buttons that shows their set of properties as buttons and asks which one they would want to build on. When one of these property buttons are clicked, depending on the property clicked it will either call buyHouse1(), buyHouse2(), or buyHouse3(). These methods all call upon checkHouse() with their specified number to check to see if the player has enough money to build a house. It also checks that there is no hotel already on the property because you can’t build on a property if there is already hotel on it. When five house are purchased on a property this is equivalent to a hotel. The user can pass their turn at any time during the purchase houses menu.

AI Player Support

Author: Jeremy Trendoff – 101160306

The AI Player was designed to perform a set of hard coded decisions to game events. These decisions were to always buy a property if allowed, always pay the fine to get out of jail, and not to buy houses or hotels on a property. These decisions were chosen so the AI Player could easily fit in our model and so that it would make the playtimes faster. The implementation of an AI turn is a screen containing the events the AI player's turn. To do this, we created a child class of Piece, called AIPiece, to implement the AI's turn events. We also added a Boolean variable in Piece called isAIPlayer. This variable will be set to true when an instance of AIPiece is created and will serve the purpose of finding AI Player instances without using the instanceof keyword.

Specialty Tiles Support

Authors: Joshua Gatto - 101150890 and Gilles Myny – 101145477

Railroad and utility act as properties but are separated so that they can be distinguished via type. This makes determining what kind of tile a player is currently on much easier. Jail is a basic tile class that has been given an integer value as an attribute. This allows us to maintain the value of bail in a natural way without hard coding it in. Go is a basic tile class that has been given an integer value as an attribute. This allows us to maintain the value of the reward for passing go in a natural way without hard coding it in. Free parking is a simple tile implementation. The only specialty tiles that fall under the Ownable class structure are Railroad and Utility. All others implement the standard Tile format.