Term Project:

In the end I created a class-based, Tkinter-based program that uses EventBasedAnimationClass to create a simplified, local version of the classic game Settlers of Catan that uses mouse clicks to operate.

First I created a game class that stores all the important information:

To start off, there are drawing functions that display a start screen and a few help screens that get the game moving, that are changed through mouse clicks. Then there is a player selection screen, which is manipulated by multiple functions and onMousePressed to select 2 - 4 players and their colors. There is also a very simple game over screen at the end, which allows you to restart from the beginning by pressing r.

Three different boards: a resource board that references the hexagonal tiles, with a true/false statement to determine whether or not the robber is on the tile, a value for the corresponding dice roll, the resource it contains, and the possible settlements that could be placed on it; then a settlement board whose indices correspond to where it is on the board, which stores whether or not a player owns that space and whether or not the owned space is a city or a settlement; and finally, a road board which contains the indices of the settlements it connects and which player owns it, if it is owned.

In relation to these I have several drawing functions that calculate how to draw hexagons for the tiles, circles for the probability tokens, circles for the settlements, stars for the cities, and thick outlines for the roads.

In order to manipulate the set up, I need to keep track of turns and have a slightly more complicated nextTurn function that will go backwards through the list of players for the second round of settlement and road placement.

For placing roads, settlements, and cities, I have true/false statements that determine if a press on the board means creating a new road or settlement, which connects to another function that checks if the placement is legal by reading the point and checking the corresponding indices in the road drawing and road board functions. These tru/false statements are manipulated by the first few rounds of next turn and if such things are bought from the bank.

The bank is one of the other things drawn in the window; in addition to info text in the middle of the screen that tells you what to do in certain cases, like in the set up, or when you play a development card or build a road; player tabs that display the player's whose turn it is and their resources, development cards, victory points, and remaining roads/cities/settlements (the idea is to pass the computer after your turn is over to maintain some privacy); and a set of dice that display dice rolls.

If it is your turn, you can click on the buttons on the bank that display what you're buying and how much they cost, you can click on one of the resources at the bottom to trade in four of one resource for another, or you can end your turn.

In set up, there is a special function that distributes the connected resources to the settlement that is placed, and after that there is a different function that distributes resources based on the roll and whose settlements are on the corresponding tiles.

If a seven is rolled, or a knight (a type of development card) is placed, if any player has more than 7 cards, the latter half of them are removed from their hand. Then you are asked to move the robber, which is denoted by the state mentioned above in the board and by a black circle on the board. You can move the robber to any tile with an opponent's settlement or city on it (even if it is connected to your own). Then you are asked to steal from a player. you do this by click their tab, and a random resource is popped from their list of resources (explained below in player class).

Knights are special cards called development cards, or devCards as I describe them in the game, that can be bought at random from the bank (there is a list in initAnimation that contains all the possible devCards), and played by clicking on them in the devCard section of the player tab.

On top of these, there is a check winner function which calculates if a player has 10 victory points (victory points are given by settlements, cities, and devCards), in which case the game is over and that player has won.

In correspondence to these things, I have a player class that is called when players are added to the list of players in the game class. This stores player data like color, roads/settlements/cities left, which cards to get rid of when a seven is played, the resource cards and development cards the player has, the number of knights played, and how many victory points the player has