Redneck Spies of Catan – User Guide

# How to Play

You are a redneck from Louisiana trapped on an island with three other rednecks from Texas, Georgia and Alabama. You have played the game Settlers of Catan before, and your fellow rednecks have agreed to attempt to survive based on the game itself. However the island is deficient in several aspects, and it is up to you to find what is possible and what is not. You know that the game can be won if you reach 10 victory points, but you don’t know how. Once you have reached 10 victory points, you will have obtained dominance over all other rednecks on the island and obtain the right to ride the lone pickup truck off the island.

Victory points are obtained by building cities and settlements: 1 for a settlement, and 2 for a city. You need resources to build cities and settlements. Each player starts off with two bricks, two woods, zero stone, two grains, and two sheep. This is enough to make two settlements. These settlements can be placed in locations using the Build option. Once this is done, you can roll the dice using ‘Roll’. You will gain resources if the dice number is equal to the number associated to a hex adjacent to one of your settlements. You can also trade 4 of a resource for another resource with the bank, using the ‘Trade’ option.

# Features

## Roll

The Roll method uses a random number generator to simulate dice rolls for gameplay. Players can gain resources depending on the outcome of the dice.

## Build

The Build method is used to create settlements and cities. It checks whether the player has enough resources to build the specified construction, and subtracts the necessary resources. Next, it displays the newly-built construction on the UI, and finally adds the necessary victory points to the player’s collection.

## Trade

You are able to trade resources at a 4:1 ratio with the bank through the trading panel. It checks if sufficient resources are available and trades them according to the users wishes. Once the trade is completed, the UI is updated to display the new resource counts.

# Limitations

There are a few limitations to our program. First of all, only settlements and cities can be built. In the real game, roads can be built as well. In addition, there are known glitches to the graphics. An artificial intelligence engine is still needed. This will give the option of facing the computer. Finally, the locations of buildings can be out of sync.

Redneck Spies of Catan – Developer’s Guide

# Overview

The game Redneck Spies of Catan requires five packages, the default package, Card package, Building package, Map package and the Player package. The default package is responsible for tying the entire game together. The card package allows for ResourceCards and DevelopmentCards to be created and added to the game. The Building package displays and creates buildings such as settlement, cities and roads and their corresponding contribution to victory. The Map package stores the map, its components and contains classes to display the map itself on a JPanel, it is also responsible for modifying the map as needed. Lastly, the Player packages contains all possessions of a player and their various attributes, it is also responsible for providing an AI along with trading and dice rolling. These packages will be described in further detail below.

# Packages

## Default package

The Default package is where everything is tied together. It consists of the Game class, which initializes the entire game and has various methods to ensure full functionality of the game. The game can be easily modified by changing several variables in the game class. The initialization methods will scramble the map, construct all objects and build the GUI. It will also harvest resources by getting dice rolls and adjacent tiles. The Listener class implements ActionListener to provide a way of launching different windows based on actions present in the Game class. The buildListener class launches the build window to allow users to build buildings while the tradeListener class does the same for trading. All of this is modular enough to allow for easy modification. For example, the paintComponent method is overloaded for simple modification. JFrames, JPanels, JLabels, JButtons and JTextFields were used in the making of this GUI.

## Card Package

The Card package is used to define resource cards used in the game. The Card abstract class is used to define the structure of a card. Each card has a name, a description, a constructor, a String output method, and a draw method. This structure is used as the framework behind Resource Card objects. In addition to the inherited methods from the Card class, the Resource Card class has a method that returns the type of resource stored. There are five types of resources: brick, wood, stone, wheat, and sheep, defined by the enumerator type ‘Resource’. The constructor of the Resource Card will initialize the resource card based on the value of Resource type, which is received through the getType method. For example, if the type is set to ‘sheep’, then a Sheep Resource Card will be created. The draw method can then be used to display the resource card in the user interface.

## Building Package

The Building package contains the objects needed to define the three different constructions a player can create: settlements, roads, and cities. The Building abstract class gives the overall structure for such a construction. The type of construction is stored in an enumerated type called ‘buildingType’. Each Building object will have a type, a location, an array of resources required to build the construction, and the victory points gained by building such a construction. In addition, the Building abstract class has a constructor to define these values, getters for all variables, and a draw method. The Building abstract class is used as the framework for the City, Road, and Settlement objects. In addition to the inherited methods from the abstract class, a method is included to set the location of the construction. The Settlement and City classes also have a method to find adjacent tiles. This is used to gain resources from adjacent hexes.

## Map Package

The Map package contains every element of the board along with the information about the types, the locations, and adjacent locations of every component. Transparent PNG image files of the board and tile types are enclosed in one folder. The classes that manage the information manipulated of the map’s objects is stored in other folders.The Location class has location objects to represent everything and gets adjacent locations with the help of arraylists. The Tile class stores and creates tile objects defined by an x-coordinate, a y-coordinate, and a type. To differentiate between the types of tiles, objects, and characteristics of the hexagonal-shaped tiles, the enums were also included in the package. Finally, the location of the map, along with its objects are stored in the Map class, drawn, and displayed in the MapPanel class, which implements JPanel.

## Player Package

The Player package consists of classes that handle the controls for the AI, dice rolling with values being passed, storing the game data for each player, and trading resources amongst other players. In the player class, each player has values initialized for the resources they have, the roads, settlements, and cities built, and the victory points earned. In the same class, the victory points are calculated, the possessions on the map are drawn, getter methods are used for elements such as the color of the settlements for respective players, roads built, trade transactions, numbers on the die rolled, and all of the other components that were initialized in the class’s constructor. There are also adder methods for specific elements that include resource cards, roads, cities, and settlements. The value of the dice rolls are stored and used in the Dice class of the Player package. The Trade class analyzes a ratio of resources of the players that want to trade items and executes the trade, if possible. Finally, the AI class, which extends the main Player class, performs its next moves automatically through the getter method.

# Suggestions for Improvement

There are many possible improvements for our game. The addition of artificial intelligence would mean that a single player can play against the computer, and does not have to bring another human player. In addition, the inclusion of roads and development cards will allow our game to more closely resemble the original game. Trading between players encourages interaction between players. In the same way, the inclusion of robbers lets players take revenge on other players, making the game much more interesting. The UI can be improved as well. Right now, most notifications are displayed in the System.out screen. These notifications can be moved to the actual UI, as a JPanel. The user interface can be made more user-friendly and aesthetically appealing as well.