

Monopoly CPT User Interface Document

Sheridan, Vivian and Aevery

Home Page

JButton
Variable Name:
PlayButt

- evt.getSource() = playButt
- This will set play page one to visible

JButton

Variable Name:
InstrucButt

- evt.getSource() = InstrucButt
- This will set the instructions page to visible



JButton

Variable Name:
ExitButt

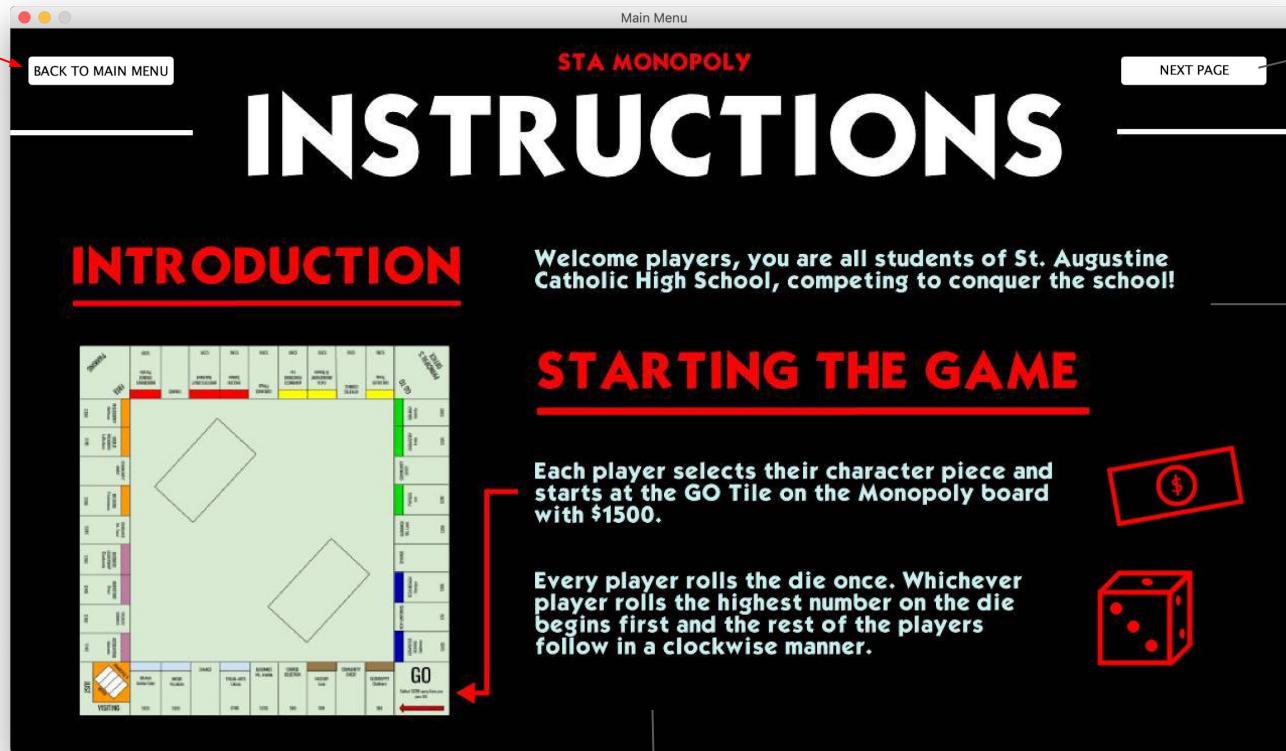
- evt.getSource() = ExitButt
- This will exit the program using the following code: System.exit(0);

JPanel : monopoly instructions

JButton

Variable Name:
backtomain

- evt.getSource()
- go back to main menu screen



JButton

Variable
Name:
thepanelinstruction
s.topage2

- evt.getSource()
- sets the blnPage1 = false and blnPage2 = true

JPanel

Panel Name:
Monopoly
Instructions

Pictures are loaded into the panel. If blnPage1 = true on the monopoly instructions panel. It displays the following image

JPanel: public monopolyinstructions thepanelinstructions;

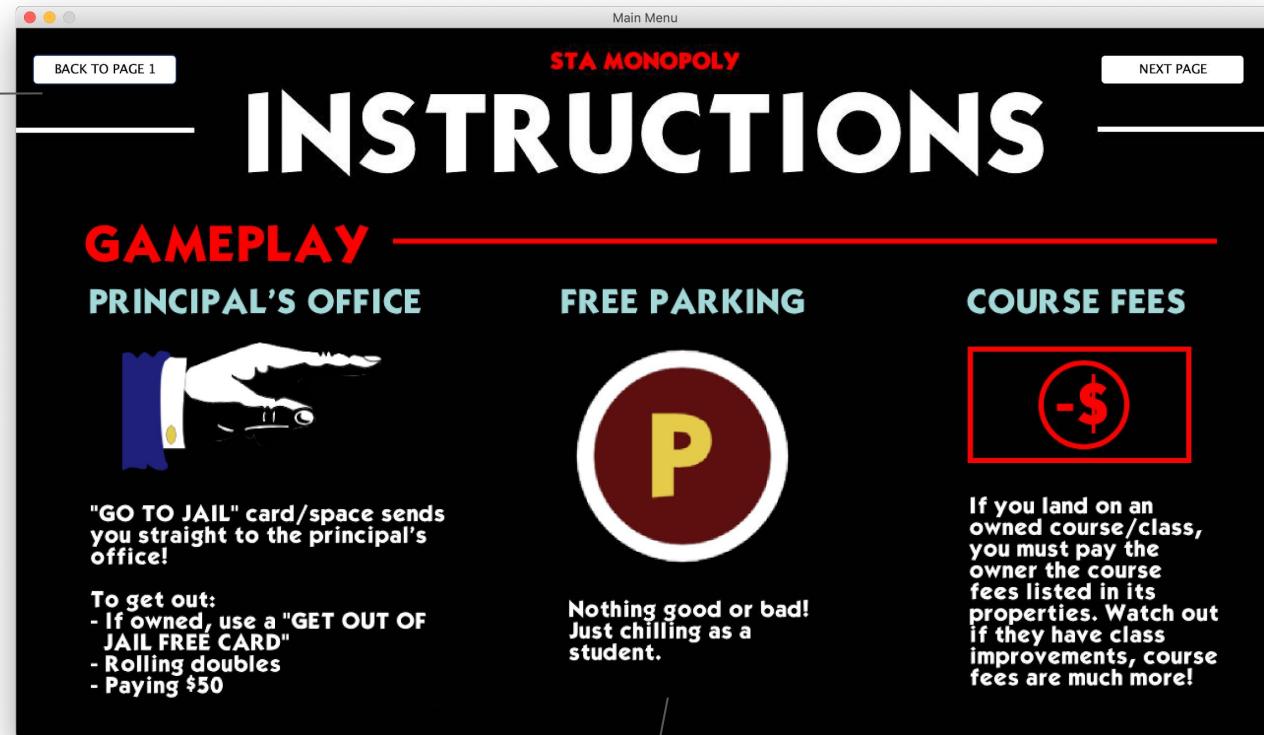
This is the JPanel that becomes visible when you press the instructions button (on the main page). BlnPage1 is set to true therefore the following image is displayed.

JPanel : monopoly instructions

JButton

Variable Name:
backtopage1

- evt.getSource()
- sets blnPage 1 = true it displays the previous slide.



JPanel: monopoly instructions

The variable blnPage2 is set to true displaying the following image

JButton

Variable
Name:
thepanelinstruction
s.topage3

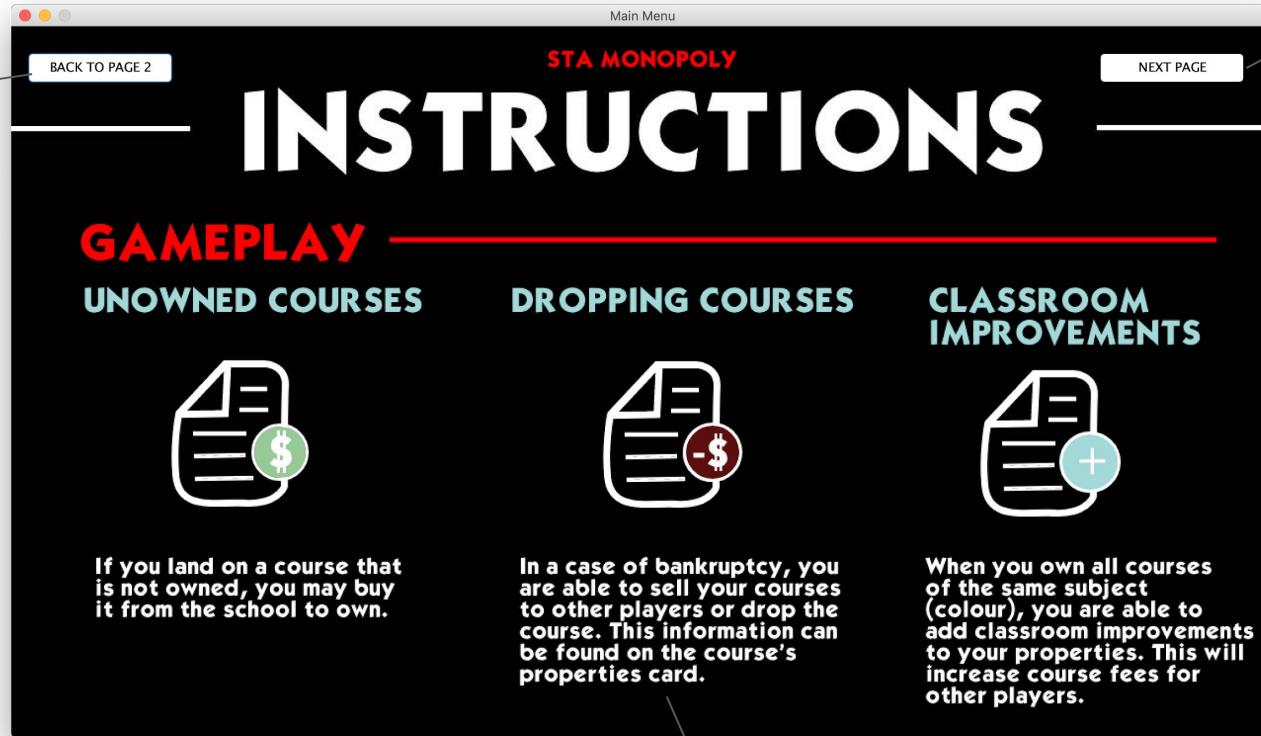
- evt.getSource()
- sets blnPage3 = true and blnPage2= false. This helps to display the next slide and load in the proper image

JPanel : monopoly instructions

JButton

Variable Name:
backtopage2

- evt.getSource()
- sets blnPage 2 = true this displays the previous slide.



JPanel: monopoly instructions

The variable blnPage3 is set to true, displaying the following image, the image is loaded into the panel (monopoly instructions)

JButton

Variable
Name:
thepanelinstruction
s.topage4

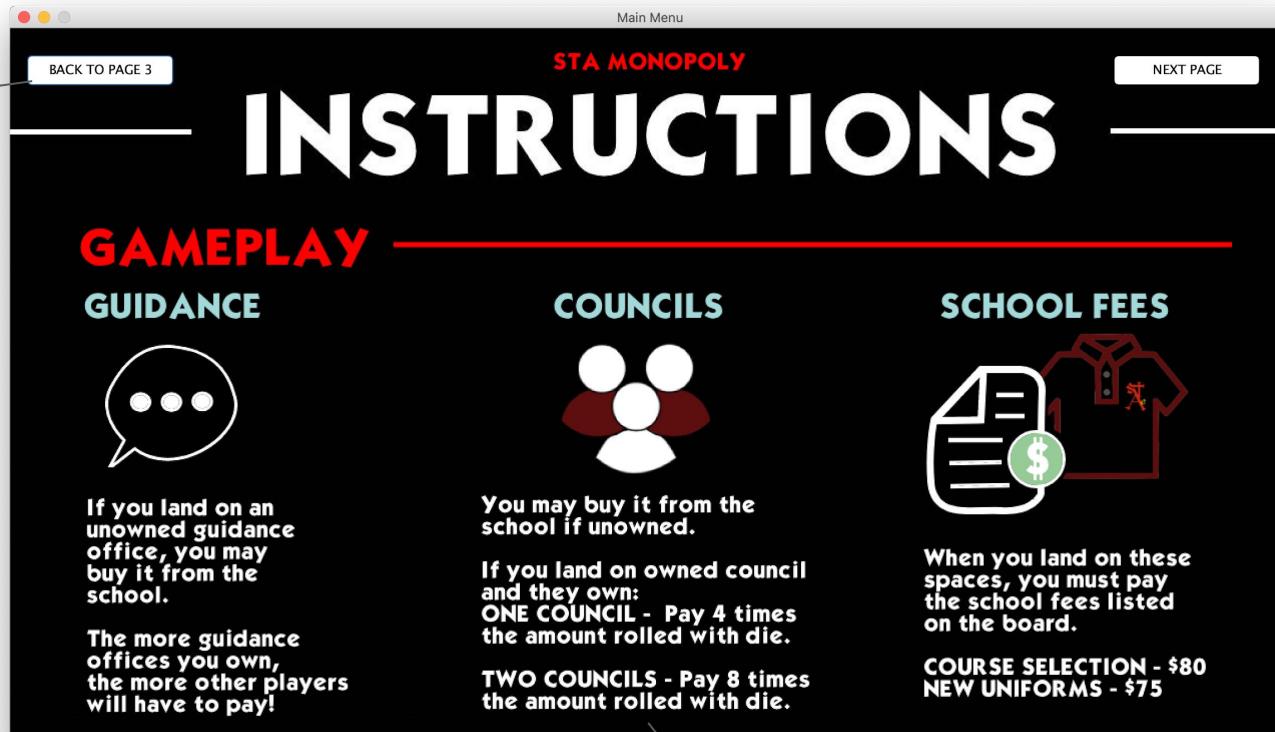
- evt.getSource()
- sets blnPage3= true and blnPage2 = false. This helps to display the next slide and load in the proper image

JPanel : monopoly instructions

JButton

Variable Name:
backtopage2

- evt.getSource()
- sets blnPage 3 = true this helps to display the previous slide.



JButton

Variable Name:
thepanelinstruction
s.topage5

- evt.getSource()
- sets blnPage 5 = true this helps to displays the next slide

JPanel: monopoly instructions

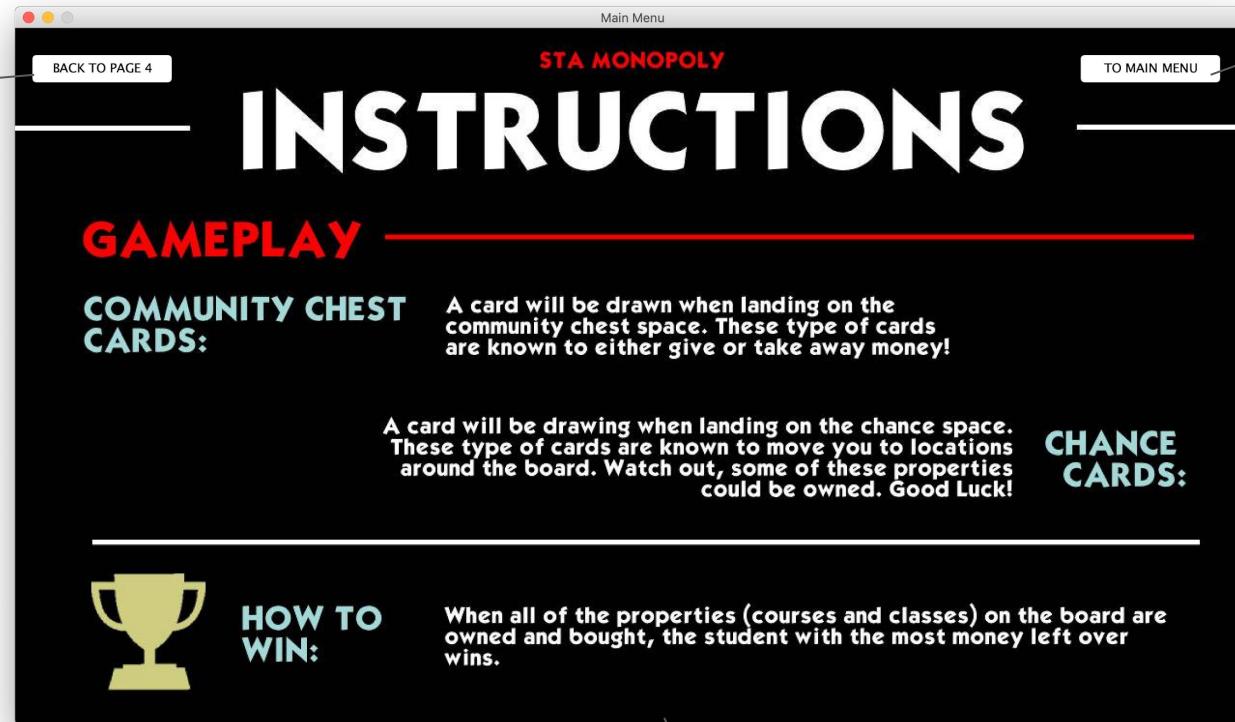
The variable blnPage4 is set to true displaying the following image, the image is loaded into the panel (monopoly instructions)

JPanel : monopoly instructions

JButton

Variable Name:
backtopage3

- evt.getSource()
- sets blnPage4 = true this helps to display the previous slide.



JButton

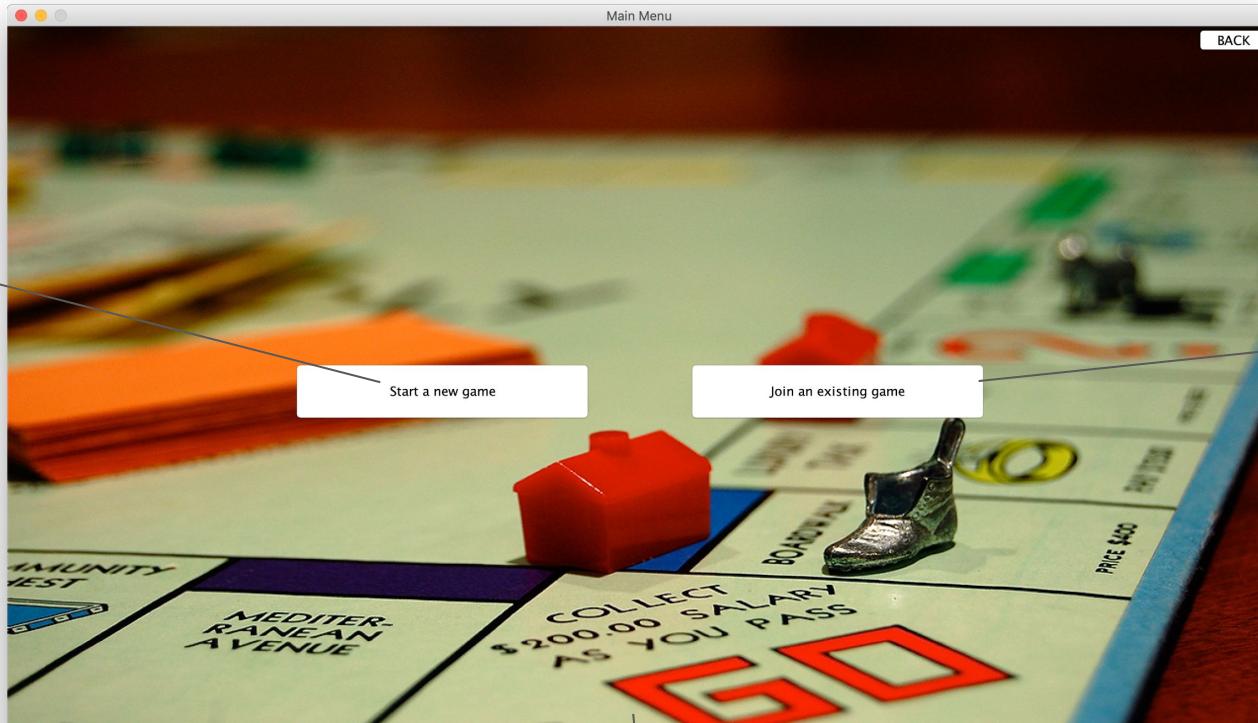
Variable Name:
backtomain5

- evt.getSource()
- sets the main page back to visible

JPanel: monopoly instructions

The variable blnPage5 is set to true displaying the following image, the image is loaded into the panel (monopoly instructions)

JPanel : playpage1



JButton

Variable Name:
playpage1.start

- evt.getSource()
- this will set:
theframe.setContentPane(start
page) and that page to visible
- this will set up a server:
ssm = new
SuperSocketMaster(1969,
this);

JButton

Variable Name:
playpage1.back

- evt.getSource()
- go back to main
menu screen and
set that to visible

JButton

Variable Name:
playpage1.existing

- evt.getSource()
- this will set:
theframe.setContentPane(joinpage)
and that page to
visible

JPanel: public serverpanel playpage1;

This is the JPanel the becomes visible when you press the play button (on the main page). This picture is loaded into the JPanel and two buttons that give you options to start a game or join an existing game.

JPanel : StartPage

JButton

Variable Name:
.startpage.strAddress =
ssm.getMyAddress()

- evt.getSource()
- It prints out your IP address

JTextField

Variable Name:
startpage.TFname

- evt.getSource()
- It notifies the animation panel of your name
- Now when you send a message you send it with your name that you entered



JButton

Variable Name:
startpage.back()

- evt.getSource()
- go back to main menu screen and sets that to visible

JButton

Variable Name:
startpage.gameplay()

- evt.getSource()
- It sets the character panel to true.

JPanel: public startgame startpage;

This is the JPanel the becomes visible when you press the start a game button (on the play page). This picture is loaded into the JPanel, and allows you to start a game.

JPanel : JoinPage

JTextField

Variable Name:
joinpage.TF

- evt.getSource()
- It notifies the animation panel and turns inputted data into joinpage.strCode
- The joinpage.strCode is then used to connect you into the server.



JButton

Variable Name:
joinpage.back

- evt.getSource()
- go back to main menu screen and sets that to visible

JButton

Variable Name:
joinpage.gameplay()

- evt.getSource()
- It sets the character panel to true.
- When you press the start game button it takes the IP address string and connects you to the server
- ssm = new SuperSocketMaster(joinpage.strCode,1969, this);

JPanel: public joingame joinpage;

This is the JPanel the becomes visible when you press "join an existing game" button (on the play page). This picture is loaded into the JPanel, and allows you to join a game.

JPanel: public stamopolycharacters characterspanel;

This is the JPanel the becomes visible when you press the start game button on either the join page or start page.

String

Variable Name: characterspanel.intPlayerN
-prints out the number of players connected to the server

Number of players: 0

Buttons and Server

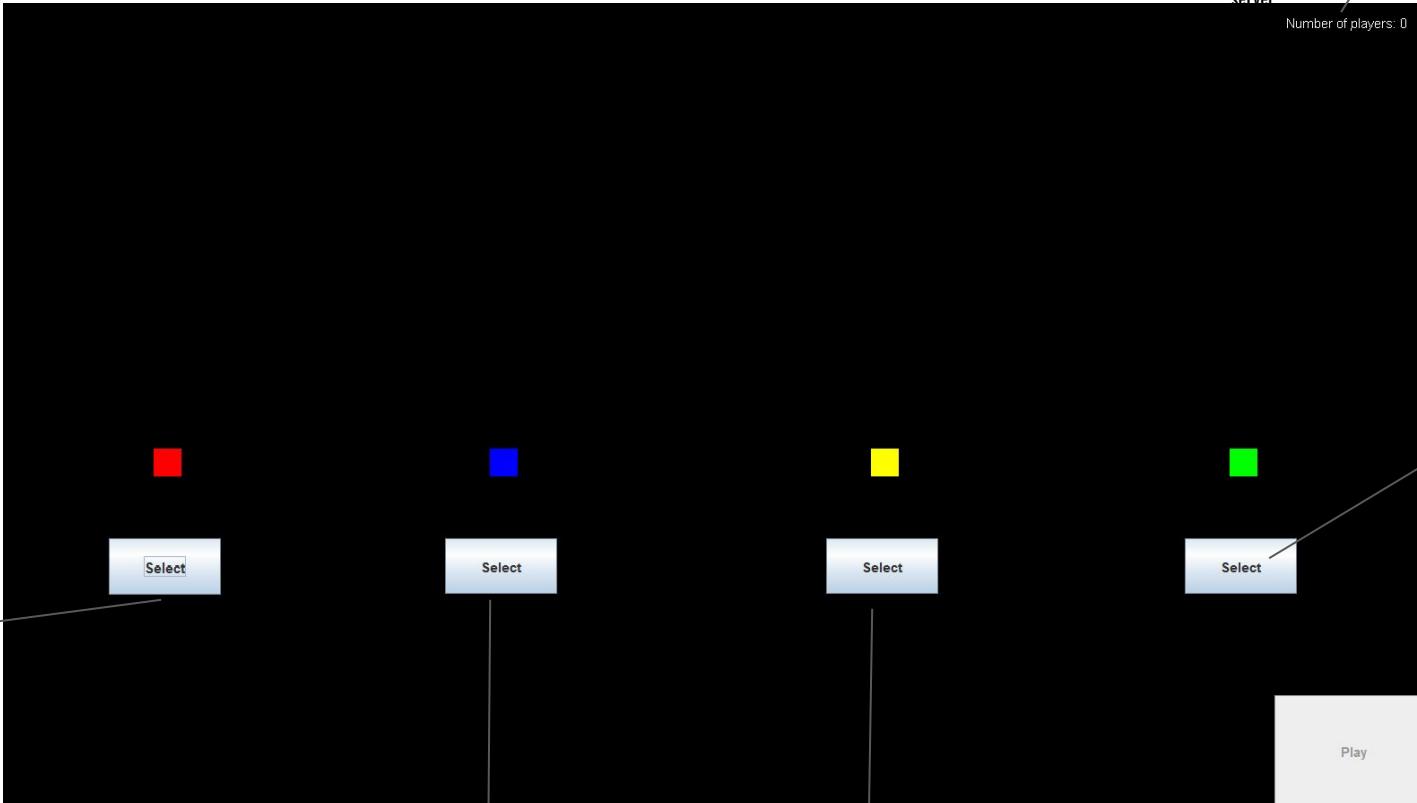
- Once a button is clicked you are no longer able to click it.
- a message is sent over the network telling other users which colour you picked this way they can not be the same colour as you

```
ssm.sendText(" select4")  
If they receive a text saying  
"select4" the button is  
disabled on their screen.
```

JButton

Variable Name:
characterspanel.select1

- evt.getSource()
- sets the colour of your icon into the colour selected monopolypanel.strColour which in this case would be red



JButton

Variable Name: characterspanel.select2

- evt.getSource()
- sets the colour of your icon into the colour selected monopolypanel.strColour which in this case would be blue

JButton

Variable Name: characterspanel.select3

- evt.getSource()
- sets the colour of your icon into the colour selected monopolypanel.strColour which in this case would be yellow

JButton

Variable Name: characterspanel.gameplay

- evt.getSource()
- if one of the buttons is pressed then you can click the play button to start the game.

JPanel : AnimationMonopolyPanel

Principal's Office:

If you land here you are sent to jail. It does a check by checking if strPropertyN = Principal's Office

Status Bar

This prints out where you landed and the total of what you rolled. This is done by storing the information in a string named strDiceSum and strPropertyN



Icon Display

-according to the colour you previously chose the box will show the appropriate colour. The data was stored in string called strColor.
-if statements were used to set the correct colour

Chance:

When landing on this you are either affected by an increase or decrease in money, or a change in position. This card is randomly generated.

Icon:

Your icon is displayed here. The location of the icon is kept as an integer value, known as intYOUx and intYOUy. Your location after rolling is sent over the network this allows other players to see where you landed.

JTextField: textField

You can type in here and send a message over the server. Your text is then shown in the above box of what you sent.

JPanel: AnimationMonopolyPanel

JButton

Variable Name: monopolypanel.buy
- evt.getSource()

- if the property is unowned you can buy it if you have enough money. The button is set to true
- A message is sent over the server to notify players that player x has bought the property

Properties Display

When you purchase a property it lists out the properties of the property.

Including the name, rent, rent with one improvement, rent with two improvements, rent with three improvements and rent with STEM upgrade.

This information is read from a csv file and was loaded into an array that was accessed called strProperties[][]

JButton

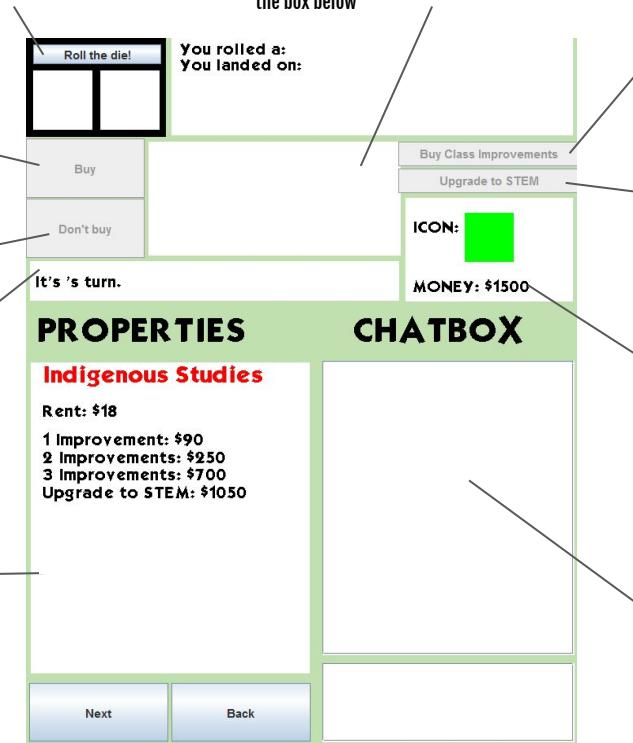
Variable Name: monopolypanel.rolldie
- evt.getSource()
- if it is your turn you can roll the dice and the button is set to true
- turn number is decided by if the turn number = player number

JButton

Variable Name: monopolypanel.dontbuy
- evt.getSource()
- Click don't buy if you wish to not purchase the property.

Display

-prints out who's turn it is by checking turn number



Display Cards

Prints out information in regards to chance and community chest cards

- A number is randomly generated if you land on a chance or community chest card
- chance and community chest cards are loaded into an array from csv files and displayed in the box below

JButton

Variable Name: monopolypanel.house
- evt.getSource()
- if you own all the properties in the set you can buy classroom improvements this helps to increase rent when other players land on your property
- We check to see if the count of how many properties is equal to the number of properties in the set, if so button is set to true

JButton

Variable Name: monopolypanel.hotel
- evt.getSource()
- if you have purchased three houses for a set you can now upgrade to a fully loaded STEM classroom and the button is set to true
- There is a check to see if the number of upgrades is equal to three. If so you can buy a STEM upgrade.

String and Integer

Variable Name: stMoney and intMoney

- a string with the amount is printed to the screen
- intMoney is kept to see how much money a user has. Every user starts with \$1500.

JTextArea and JScrollPane

Variable Name:textfield and scroll

- if incoming messages are coming into the client or server they are printed here. This allows users to chat with each other.

JPanel : AnimationMonopolyPanel

Pictures for winner and loser screen

Variable Name: winner and loser

- if you have no more money the loser picture will show up on the screen
- If you are the last surviving player the winner picture will show up on the screen

