***Class: GameOfNim* (Startup)**

This class is the first class to be executed when this program runs.

*Method(s):*

main()

* Creates an instance of NimGame and NimGUI (passes NimGame into NimGUI)
* Instantiates a JFrame and alters the following properties
  + Size
  + Location
  + Default close operation
  + Content pane
  + Visibility

***Class RemovesController* (Controller)**

- This class acts on both the model and view components.

* Controls the flow of information to the model object and updates the view whenever the state is changed.

- This class implements the ActionListener interface.

- This class has 2 instance variables

* NimGame game
* JTextField textField

*Constructor(s):*

RemovesController(NimGame, aGame, JTextField aTextField)

* Initializes instance variables

*Method(s):*

void actionPerformed(ActionEvent e)

* Removes a token(s) from the pile when a valid integer value is entered in the JTextField

***Class NimGame* (Model)**

- This class is responsible for modeling the current problem to the user

* Updates the state of the program

- This class extends the Object Superclass

- This class has 5 instance variables

* NimGUI gameView
* int MAX\_REMOVE
* int pileSize
* String playerTurn
* String winner

*Constructor(s):*

NumGame()

* Invokes the super() method
* Initializes the pileSize and playerTurn variables

*Method(s):*

int getPileSize()

* Returns the number of tokens in the pile

String getPlayerTurn()

* Returns the player who has the turn

String getWinner()

* Returns the winning player

boolean gameOver()

* Determines if the game is over

void removeTokens(int howMany)

* Removes a user specified number of tokens

boolean isLegalMove(int howMany)

* Determines if the move is legal (between 1-3 tokens if there are enough tokens left)

String otherPlayer(String currentPlayer)

* Determines who the next player is

void updateView()

* Updates the game view (updates the state)

***Class NimGUI* (View)**

This class is responsible for showing the program to the user

* Shows relevant information to the user in a user-friendly way

- This class extends the JPanel Superclass

- This class has 6 instance variables

* NimGame game
* JTextField redRemoves
* JLabel redWins
* JTextField blackRemoves
* JLabel blackWins
* JLabel pileSize

*Constructor(s):*

NimGUI(NimGame newGame)

* Invoke the super() method
* Initialize game instance variable
* Invoke internal methods (layoutView, registerControllers, update)

*Method(s):*

void layoutView()

* Draws the initial layout for the game board

void registerController()

* Assigns the controllers to the remove tokens in the JTextFields

void update()

* Redraws the game board according to the current state of the program