**Raymond Smith**

**Final Project**

**Smith Game Tracker Classes**

**Class: Player**

The class “Player” represents a player of any competition. This is the parent class of the “BasketballPlayer” class. The class stores the player’s first and last name and jersey number. The methods implemented are:

* public Player(String first, String last, int num) – This is the constructor for the class. It receives the first and last names and the jersey number
* public String getFirstName() – This returns the players first name
* public String getLastName() – This returns the players last name
* public int getNumber() – This returns the players number
* public void setFirstName(String first) – This receives and sets the player’s first name
* public void setLastName(String last) – This receives and sets the player’s last name
* public void setNumber(int num) – This receives and sets the player’s number

**Class: BasketballPlayer**

The class “BasketballPlayer” is a child class of “Player”. This represents and stores data for a basketball player during a game. It stores data for points, shots taken, shots made, 3 point shots taken, 3 point shots made, free throws taken, free throws made, blocks, steals, assists, fouls and rebounds.

* public BasketballPlayer(String first, String last, int num) – This is the constructor for the class. It receives the first and last names and the jersey number. Sets the instance variables for stats to 0
* public void setPlayerAlert(PlayerAlert p) – Receives an instance of the PlayerAlert interface in order to set the Alert for when players get into foul trouble or foul out
* public String toString() – This sets what will be displayed when the players are put into a combo box
* public int getShotsTaken() – This returns the amount of shots taken excluding free throws
* public int getMadeShots() – This returns the amount shots made excluding free throws
* public int getThreesTaken() – This returns the amount of 3 point shots taken
* public int getThreesMade() – This returns the amount of 3 point shots made
* public int getFreesTaken() – This returns the amount of free throws taken
* public int getMadeFrees() – This returns the amount of free throws made
* public int getPoints() – This returns the amount of point scored
* public int getFouls() – This returns the amount of fouls committed
* public int getSteals() – This returns the amount of steals
* public int getBlocks() – This returns the amount of shots the player blocked
* public int getAssists() – This records the amount of assists the player has
* public int getRebounds() – This returns the amount of rebounds the player has
* public void made2() – This records the stats for a player making a 2 point attempt
* public void missed2() – This records the stats for a player missing a 2 point attempt
* public void made3() – This records the stats for a player making a 3 point attempt
* public void missed3() – This records the stats for a player missing a 3 point attempt
* public void freeMade() – This records the stats for a player making a free throw
* public void missedFree() – This records the stats for a player missing a free throw
* public void commitFoul() – This records the stats for a player committing a foul
* public void assist() – This records the stats for a player making an assist
* public void block() – This records the stats for a player making a block
* public void steal() – This records the stats for a player making a steal
* public void rebound() – This records the stats for a player getting a rebound
* public String getFieldGoalPerc() – The returns the percentage of shots the player has made excluding free throws
* public String getThreePointPerc() – This returns the percentage of shots the player has made for behind the 3 point line
* public String getFreePerc() – This returns the percentage of shots the player has made from the free throw line

**Class: GameTrackerUI**

The GameTrackerUI class is the user interface to load players and track the game data. Accepts the users input to load players names and numbers. When players are all entered, it then switches windows and allows for selection of rostered players and for selecting of events happening during the basketball game. It warns the user when players are close to fouling out and lets them know when they have fouled out. When a player fouls out they are no longer available to record game events. When the game is over the end game button is selected and sends the program to the summary class.

* public GameTrackerUI() – Constructor that puts the window together and sets fields.
* public void alert(String title, String msg) – Alerts players with appropriate messages when players reach certain foul situations
* public void actionPerformed(ActionEvent e) – Tells what to do when certain buttons are pressed
* public void ItemStateChanged(ItemEvent i) – Update stats as a player is selected from the comboBox
* public void loadPlayer() – Loads player into the ArrayList
* public void updateStats() – Updates the stats as new events happen.

Class: SummaryUI

The SummaryUI class is the user interface to view the summaries for each player after the game. If the user desires, they may export the data to a .csv file.

* public SummaryUI(ArrayList<BasketballPlayer> pList) – Contstructor to build window
* public void actionPerformed(ActionEvent event) – Exports data to .csv file
* public void itemStateChanged(ItemEvent i) – Updates the stats as players are selected