

Slot Machine Game

By Raveen Savinda Rathnayake

Table of Contents

List of Figures.....	3
Introduction.....	4
Functional Requirements.....	6
Non-Functional requirements	6
Domain model	7
Class Diagram	8
Use Case Diagram	9
Use Case Descriptions	10
Screen shots.....	14

LIST OF FIGURES

Figure 1	Domain Model
Figure 2	Class Diagram
Figure 3	Use case Diagram
Figure 4	Screenshot 1: Initial Screen
Figure 5	Screenshot 2: After pressing "Add Coin"
Figure 6	Screenshot 3: After pressing "Bet One"
Figure 7	Screenshot 4: After Pressing "Bet Max"
Figure 8	Screenshot 5: Player Lose game. No reels matched.
Figure 9	Screenshot 6: Try again screen. Two reels matched.
Figure 10	Screenshot 7: Statistics window
Figure 11	Screenshot 8: Exit confirmation box

Introduction

This is a GUI application which represents a Slot machine game. This slot machine has three reels with six different types of symbols. The user gets the chance to spin these three reels and after the spin if all the reels got the same symbol, the game is won. To play a game there must be credits and the player must redeem the credits as bets to play the game. If the player is won, the bet amount and winning credits are added to the remaining credits. Winning credits are calculated according to the assigned value of the symbol. Reels are implemented using threads in java. I have used jFreeCharts to implement and customize the Pie chart which displays the statistics.

The GUI has the following graphical component:

- An "Add coin" button: this let the user to add a coin to the credit.
- A "Credit Area": this informs the user of the number of credits that he/she has left to play with. Note that you can't continue to play with no credits. When you begin the program, you are given 10 free credits.
- A "Bet One" button: This allows the user to bet a single credit.
- A "Bet Max" button: This allows the user to bet a maximum number of credits. The maximum number of credits is permanently set to 3. This button is just a convenience, as you may add more credits if you like by clicking the "Bet One" button.
- A "Reset" button: This allows the user to return the amount that he/she has bet to the credits that he/she still has.
- A "Spin" button: This causes the three reels to spin.
- A "Bet area": This informs the user of the number of credits that he/she is betting on the current game.

- Three reels: When the Spin button is clicked, all three reels will begin to spin. To stop the reels from spinning, simply click on one of them.
- “Statistics” button: This allows the user to check his/her number of wins, losses, and the average number of credits that she/he has netted per game. If the average is positive, then the player has been, on average, winning more credits than losing. If it is negative, then the player has been, on average, losing more credits than winning.

A game is over after all three reels have been clicked and have stopped spinning. The results of the game will be displayed in the information area.

NOTE: A player cannot play if his credit is zero.

Functional Requirements

- 1) Adding a coin
If the user needs to add more coins other than the initially given 10 coins, there must be a component (graphical) to do this action.
- 2) Credit Area
The credit area should display the available coins at a moment. To play a game there must be credits. A game cannot be continued to play with no credits.
- 3) Adding one coin for bet
The system must provide a function to user to add a one coin as the bet for a game. When the bet one function is called the bet amount should have gone up by one.
- 4) Bet Area
The Bet Area should display the bet at a moment. To play a game there must be a bet. Game cannot be played without a bet.
- 5) Reset function
If the user wants to reset the bet and return the bet amount to credits area prior to a particular game play, there must be a function for that.
- 6) Spinning three reels
There must be a function to spin three wheels. When the user calls for this function it should spin the reels and when the user clicks on one of the reels, all three reels must have stopped. Each time the user clicks on spin the reels, each reel must display different sequence of symbols. A reel must contain 6 symbols which are seven, bell, watermelon, plum, lemon and cherry.
- 7) Statistics
Statistics function allows the user to check his/her number of wins, losses, and the average number of credits that she/he has netted per game. This must be in a new window and on that window there must be a button to save statistics. When this button clicks number of wins, loses and average credits netted per game must be saved in a text file with the current date and time as the file name.
- 8) When stopping three reels if symbols in all three reels are the same, then the game is won. If two symbols in reels are same, the user gets another chance to spin. If no symbols in reels are matching, the game is lost. If the player is won or lost appropriate messages should be shown and if wins the appropriate coins must be added to the credit area.

Non-Functional Requirements

- 1) Usability
- 2) Reliability
- 3) Quality

Domain Model

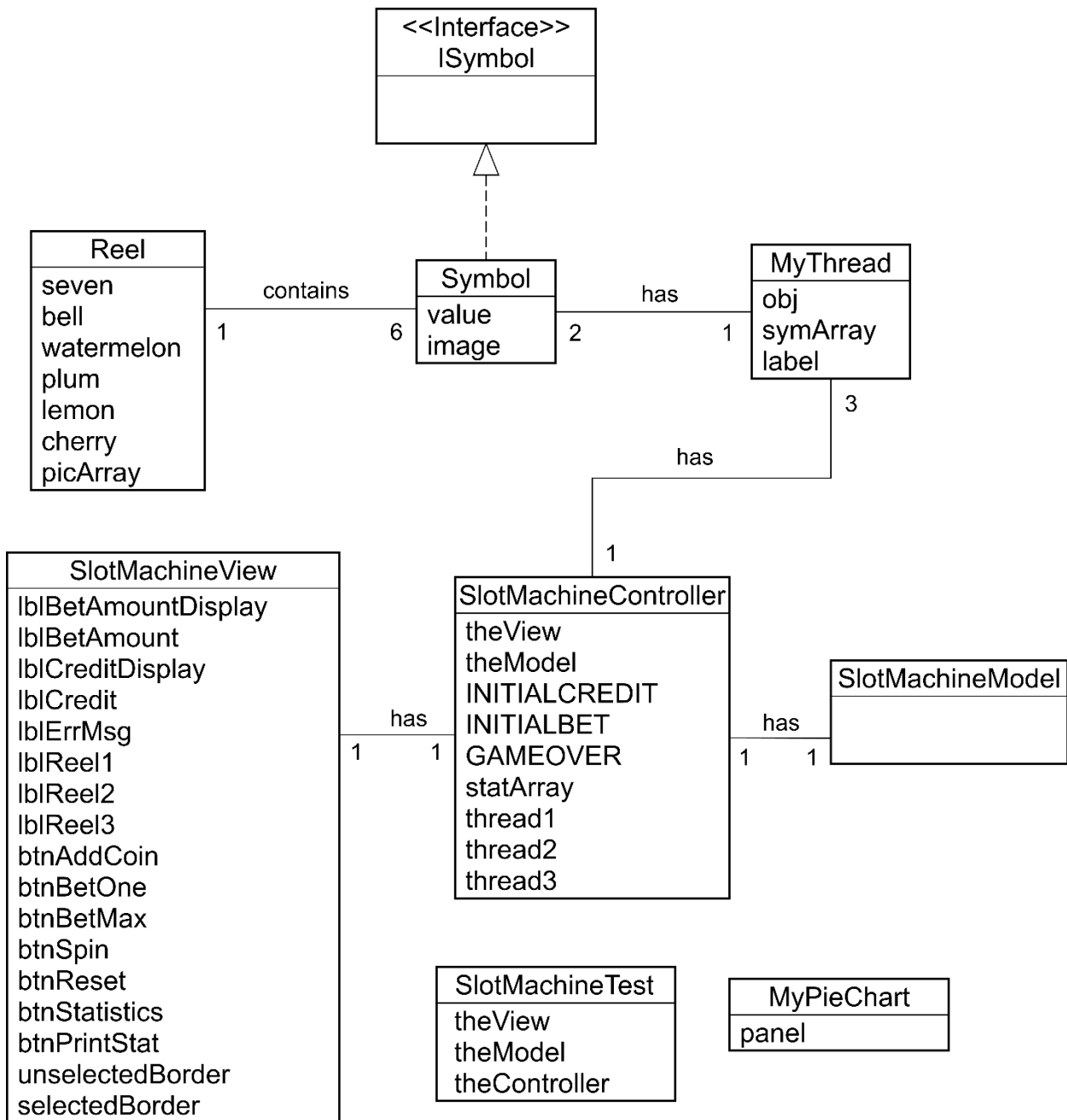


Figure 1

Class Diagram

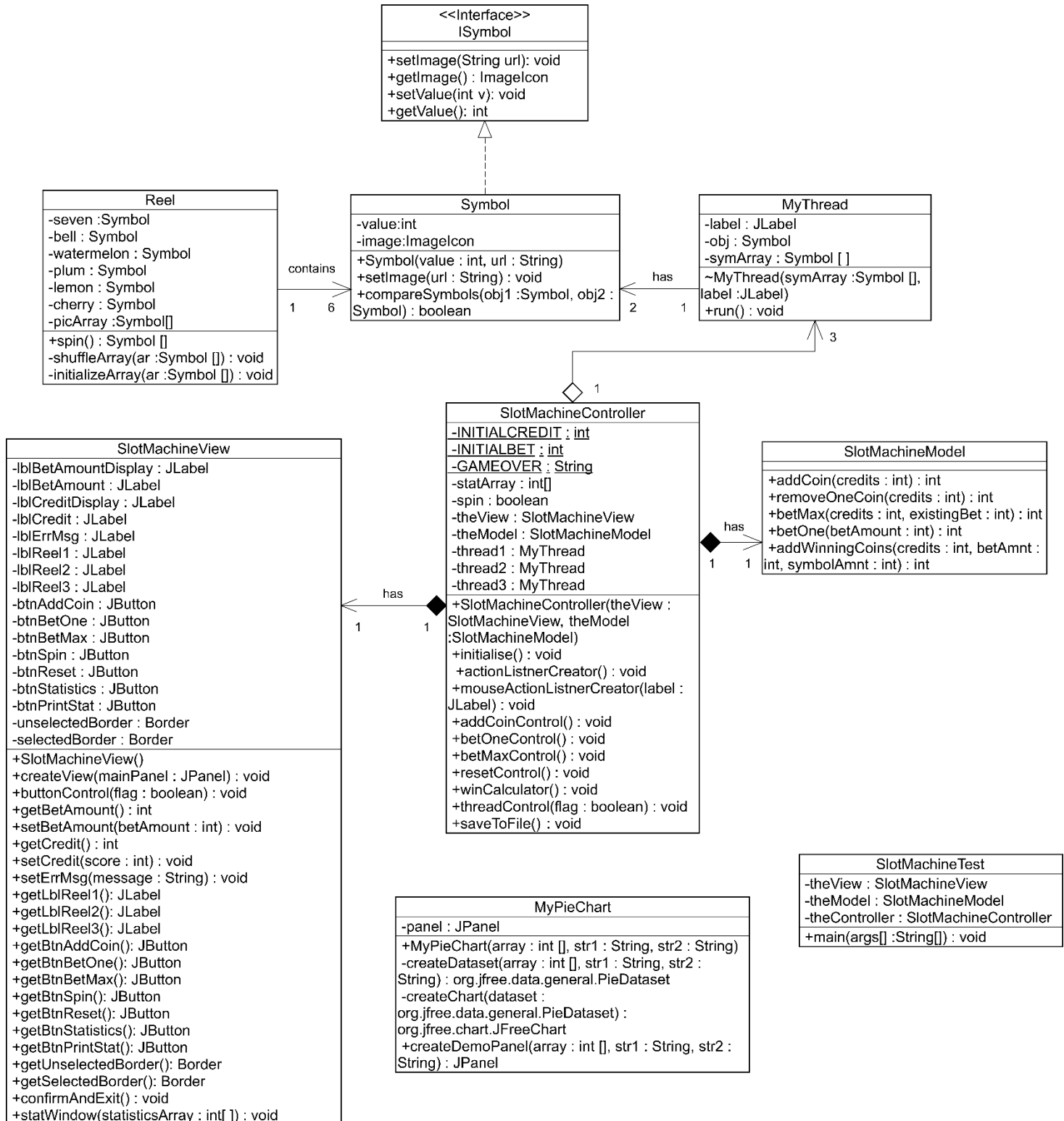


Figure 2

Use Case Diagram

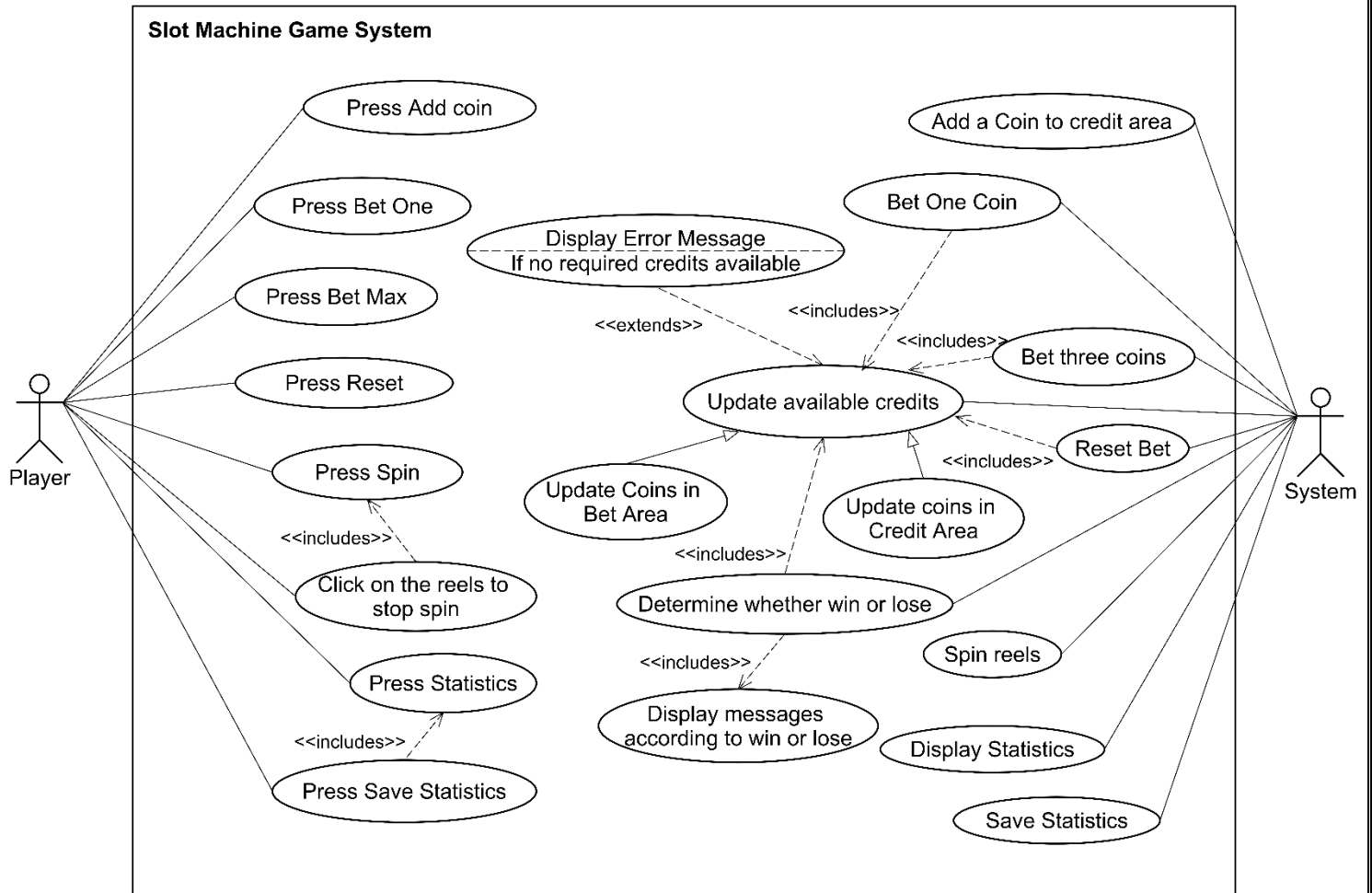


Figure 3

Use Case Descriptions

Use Case Name:	Add a coin to credit area	
Use Case No:	01	
Priority:	High	
Participating Actors:	Player, System	
Pre Conditions:	-	
Trigger Event:	Pressing "Add Coin" Button	
	Player	System
Main flow of events:	1) Press "Add Coin"	2) Get current credits 3) Add a coin to current credits 4) Display current credits in "Credit Area"
Alternate Flow:	-	
Exceptional Flow:	-	
Post-Conditions:	-	
Inclusions	-	
Extensions	-	

Use Case Name:	Bet One Coin	
Use Case No:	02	
Priority:	High	
Participating Actors:	Player, System	
Pre Conditions:	currentCredits>0	
Trigger Event:	Pressing "Bet One" button	
	Player	System
Main flow of events:	1) Press "Bet One" button	2) Get current credits 3) <<include>> Update available credits
Alternate Flow 1:	3) Press "Add coin" 4) Press "Bet One"	2) Display "You have no credits" 5) <<include>> Update available credits
Exceptional Flow:	-	
Post-Conditions:	-	
Inclusions	Update available credits	
Extensions	-	

Use Case Name:	Bet three coins	
Use Case No:	03	
Priority:	High	
Participating Actors:	Player, System	
Pre Conditions:	currentCredits>2	
Trigger Event:	Pressing the “Bet Max” button	
	Player	System
Main flow of events:	1) Press “Bet Max” button	2) Get current credits 3) <<include>> Update available credits
Alternate Flow 1:	3) Press “Add coin” until currentCredits become 3 or more 4) Press “Bet Max”	2) Display “You have less than 3 credits” 5) <<include>> Update available credits
Exceptional Flow:	-	
Post-Conditions:	-	
Inclusions	Update available credits	
Extensions	-	

Use Case Name:	Update available credits	
Use Case No:	04	
Priority:	High	
Participating Actors:	System	
Pre Conditions:	-	
Trigger Event:	Pressing “Bet One” or “Bet Max” or “Reset” buttons	
	User	System
Main flow of events:		1) Get current credits 2) Get current bet 3) Update current credits 4) Update current bet 5) Display current credits in “Credit Area” 6) Display current bet in “Bet Area”
Alternate Flow:	-	
Exceptional Flow:	-	
Post-Conditions:	-	
Inclusions	-	
Extensions	Display Error Message	

Use Case Name:	Reset Bet	
Use Case No:	05	
Priority:	High	
Participating Actors:	Player, System	
Pre Conditions:	-	
Trigger Event:	Pressing "Reset" button	
	Player	System
Main flow of events:	1) Press "Reset" button	2) Get current credits 3) <<include>> Update available credits
Alternate Flow:	-	
Exceptional Flow:	-	
Post-Conditions:	-	
Inclusions	Update available credits	
Extensions	-	

Use Case Name:	Display Statistics	
Use Case No:	06	
Priority:	High	
Participating Actors:	Player, System	
Pre Conditions:	-	
Trigger Event:	Pressing "Statistics" button	
	Player	System
Main flow of events:	1) Press "Statistics" button	2) Get wins, losses and average credits netted per game 3) Display wins and losses in a pie chart 4) Display average credits netted per game
Alternate flow:	-	
Exceptional Flow:	-	
Post-Conditions:	-	
Inclusions	-	
Extensions	-	

Use Case Name:	Determine whether win or lose	
Use Case No:	07	
Priority:	High	
Participating Actors:	Player, System	
Pre Conditions:	-	
Trigger Event:	Clicking on a spinning reel	
	Player	System
Main flow of events:	1) Clicking on one of the three spinning reels	2) Check all three reels got the same symbol at the end 3) If all three symbols are same <<include>> Update available credits, <<include>> Display messages according to win or lose 4) If two reels got the same symbol, another chance to spin the reels given and display "Try again" 5) If three reels got different symbols <<include>> Update available credits, <<include>> Display messages according to win or lose
Alternate flow:	-	
Exceptional Flow:	-	
Post-Conditions:	-	
Inclusions	Update available credits Display messages according to win or lose	
Extensions	-	

Screen Shots



Figure 4: Initial Screen

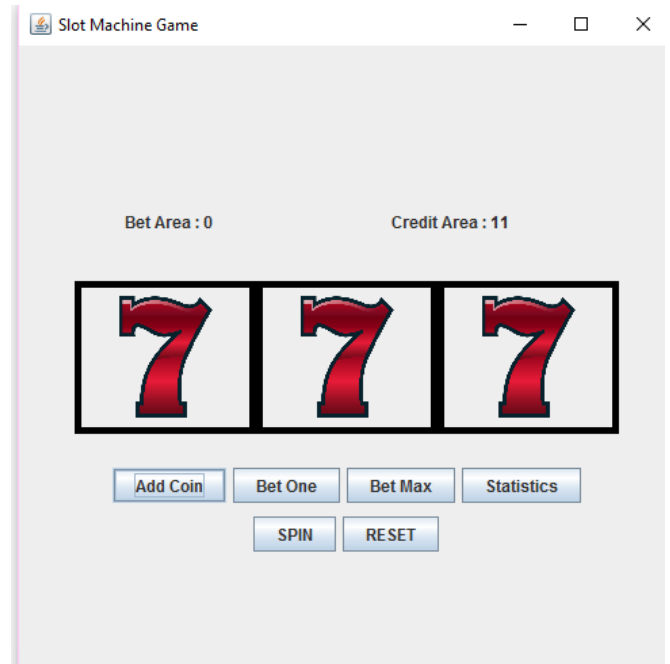


Figure 5: After pressing
"Add Coin" button



Figure 6: After pressing
"Bet One" button



Figure 7: After pressing
"Bet Max" button

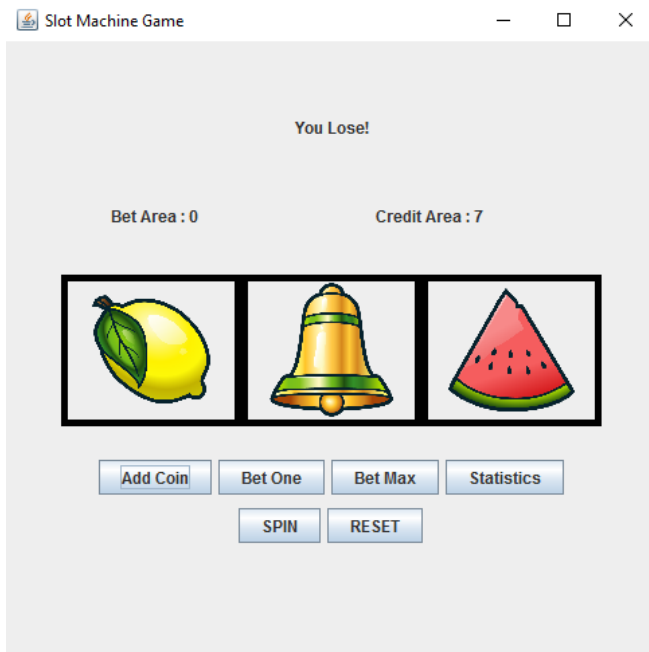


Figure 8: Player Lose game. No reels matched

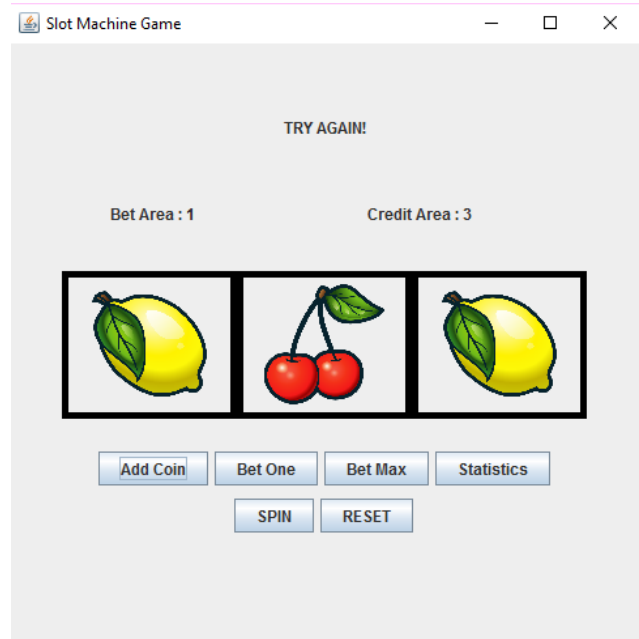


Figure 9: Try again screen. Two reels matched

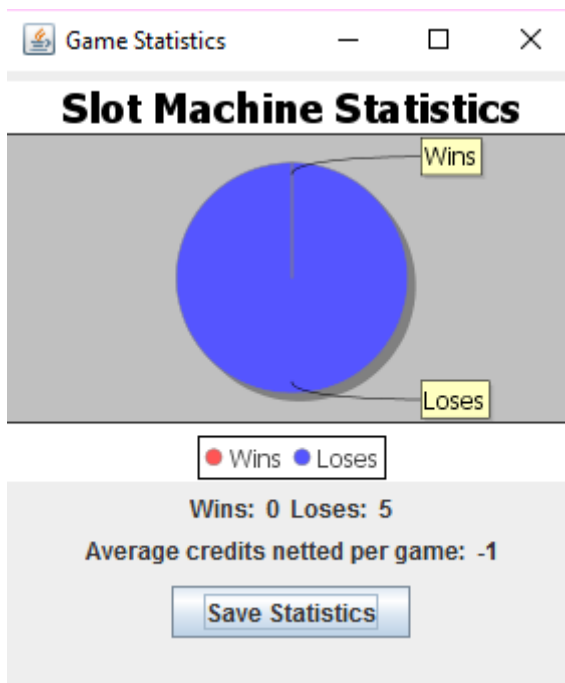


Figure 10: Statistics Window

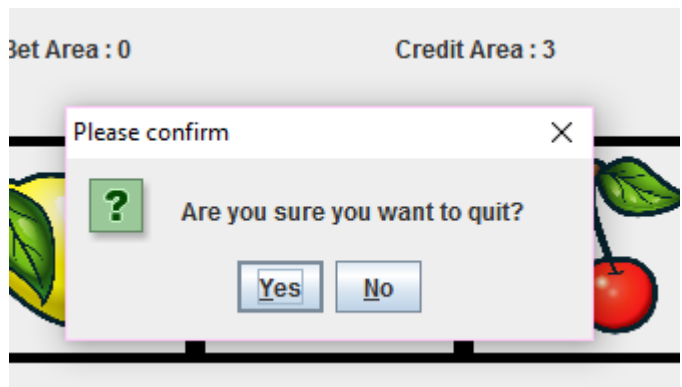


Figure 11: Exit confirmation box