Team Hypnocode
David Chou
Zane Melcho
Bao Nguyen
Tin Tran

CS151 Section 1
Prof. Mak
Assignment #3
Due: 3/09/12 11:59PM

Design Specification

CRC Cards

Game
Description: The underlying class that controls the game
Responsibilities:

Name	Collaborator
Prompt Main Menu	Match
Start Game	
Display Help	
Display Score	

Match
Description: Handles the the rounds of the match
Responsibilities:

Name	Collaborator
Make Throws	Player
Declare Winner	
Tell when match is over	

Player

Description: Represents the actors
Responsibilities:

Name	Collaborator
Increment Score	
Compare throws	
Make throw	

Human
Description: Represents the human (user) player
Responsibilities:

Name	Collaborator
Make a throw	Player

<<ThrowRequestor>>
Description: Requests a throw type
Responsibilities:

Name	Collaborator
Request a throw	ThrowTextRequester

ThrowTextRequest
Description: Make throw of text type
Responsibilities:

Name	Collaborator

Make throw	<<ThrowRequestor>

Computer	
Description: Represents computer actor	
Responsibilities:	

Name	Collaborator
Make a throw	Player

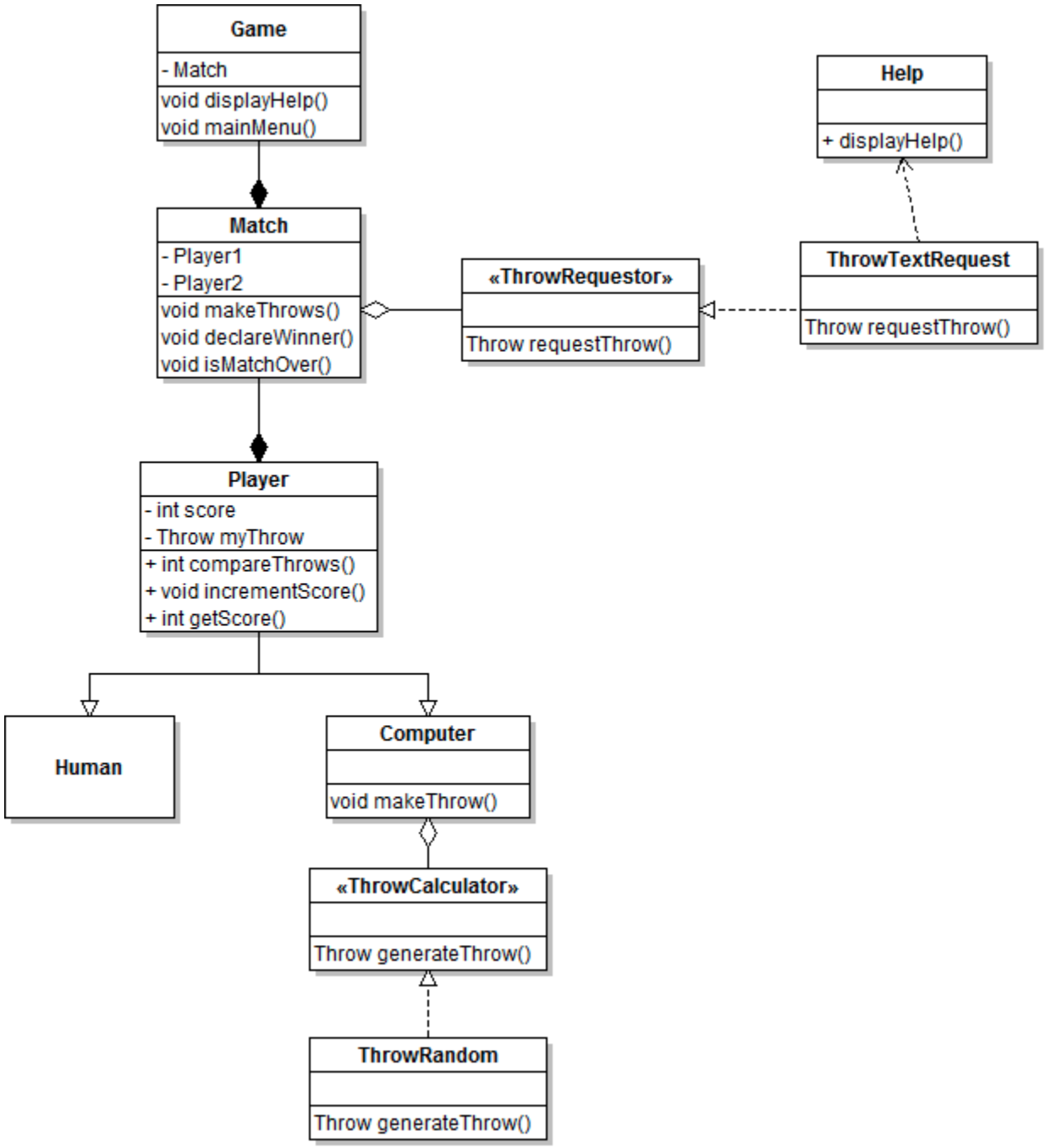
<<ThrowCalculator>>	
Description: Calculate a throw of different types	
Responsibilities:	

Name	Collaborator
Generate a throw by different methods	ThrowRandom

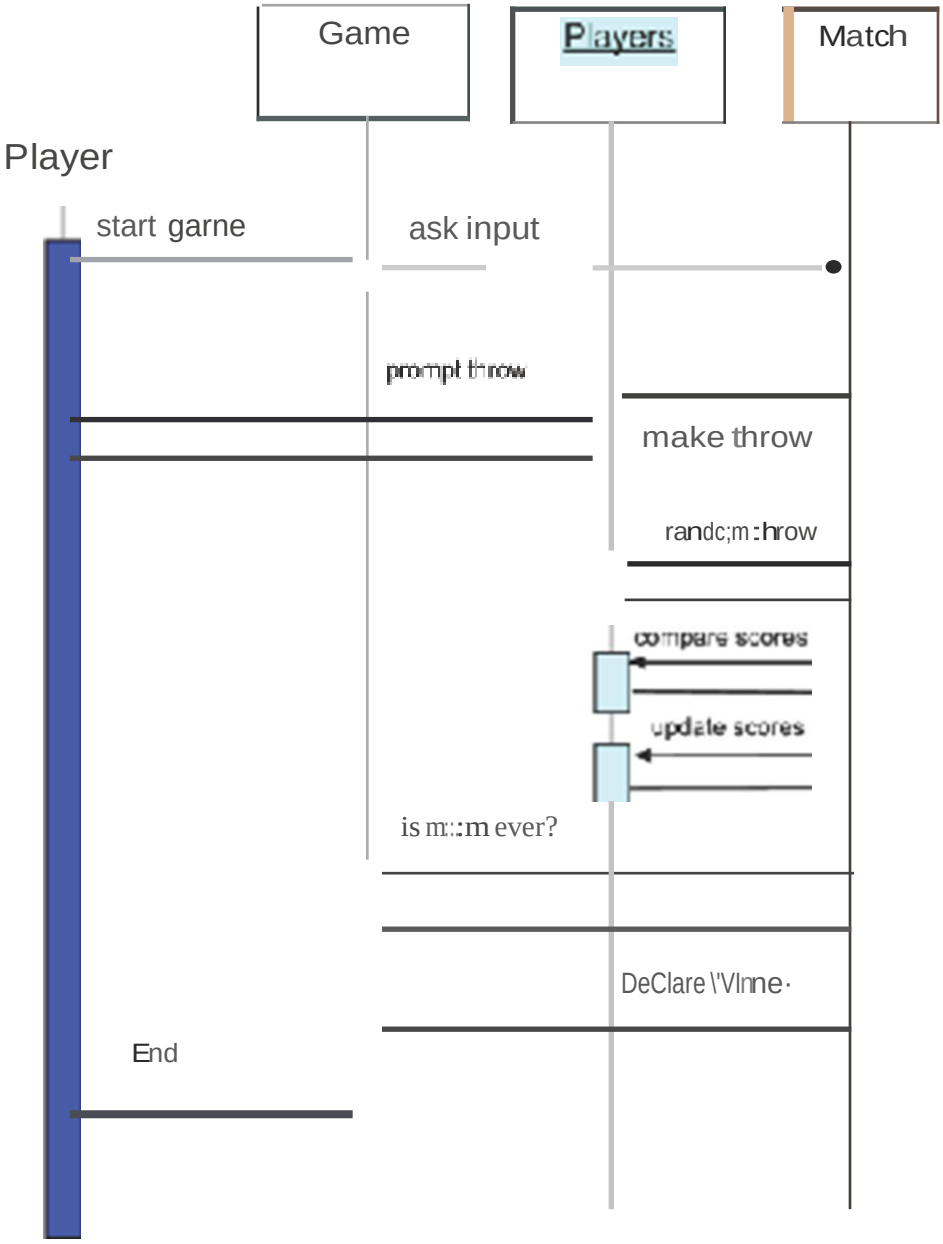
ThrowRandom	
Description: Make a throw based on random output	
Responsibilities:	

Name	Collaborator
Generate random throw	<<ThrowCalculator>>

UML Class Diagrams



UM.. SequenceDiagram



Updated Functional Specification

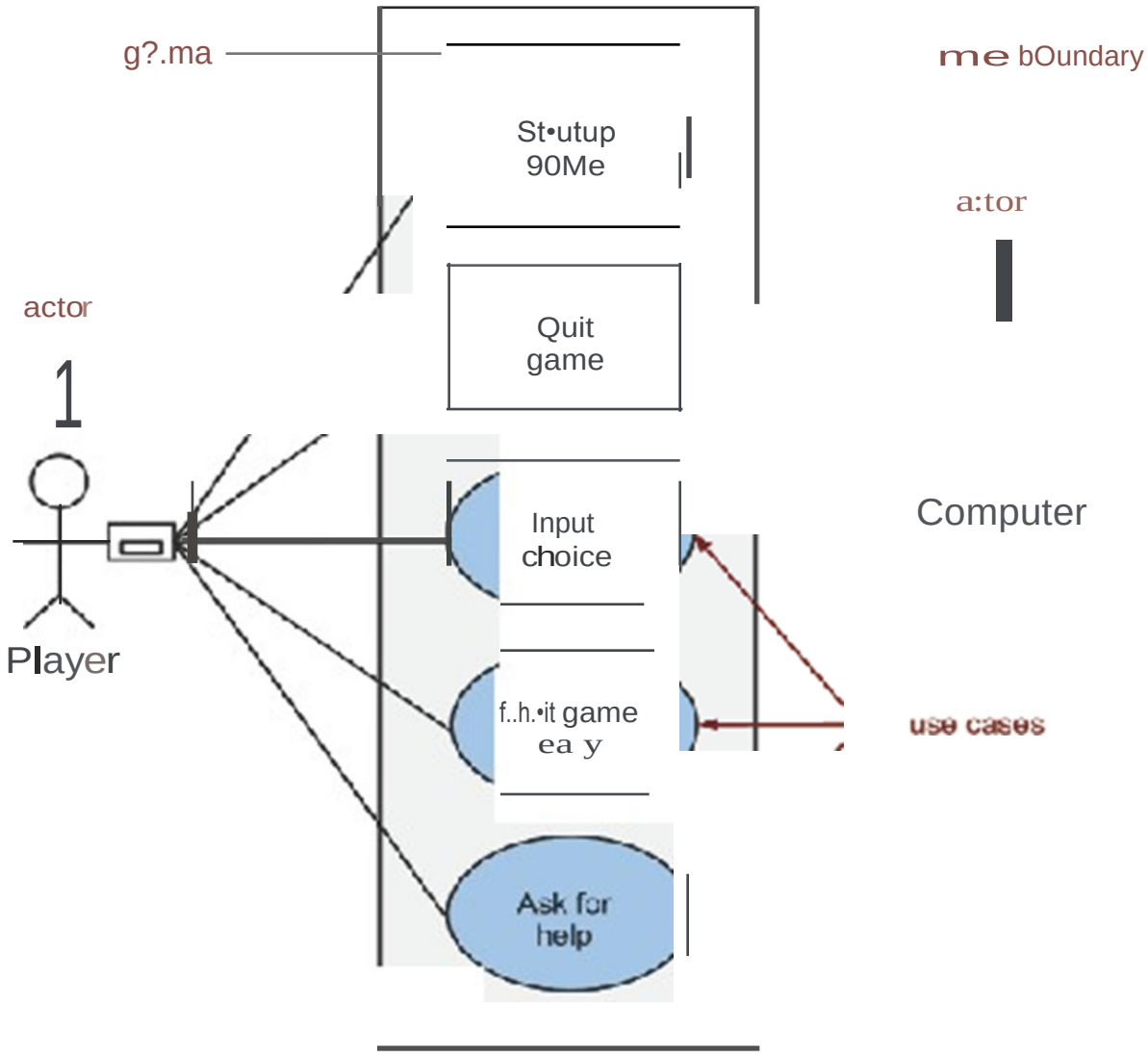
Problem Statement:
To design a program in which one player plays a match, consisting of a variable number of throws against the computer, complete with scoring.

Objectives:
Application provides basic interface of Rock-Paper-Scissors game play.

Functional Requirements:
- Human player must be able to choose number of throws in a match.
- Human player must input their choice (rock, paper, scissors).
- Computer must make selection at random.
- Computer must determine result of throw and update scores.
- User must be able to view current score, current throw, and a help message.
- After each match, it should display the number of wins by player, computer, and ties then return to menu.
- Player must be able to end the match or quit the game at any time.

Nonfunctional Requirements:
- Text based input and display
- Computer's choice must not depend on player's previous choices
- Coded in Java language

Use Case Diagram



Use Case Description 1

Use Case name:	Player makes a throw
Project name:	Rock Paper Scissors
Team:	Hypnocode
Date:	2/15/12

1. Goal

Allow player to make a selection for a throw.

2. Summary

Player chooses rock, paper, or scissors and the computer randomly makes a selection

3. Actors

Actor 1: Player

4. Preconditions

1. New match is started when the game application starts.
2. Throw has not begun.

5. Trigger

Player enters one of three choices (R)ock,(P)aper, or (S)cissors.

6. Primary Sequence

Step	Action
1	Player enters a choice.
2	Computer picks random choice.
3	Window shows throw ASCII art.
4	Application adjusts score depending on outcome.

7. Primary Postconditions

1. Wins, ties, or losses are adjusted.
2. ASCII art is reset.

8. Alternate Sequences	
Alternate Trigger	Player picks an invalid selection.
Step	**Action**
1	Player makes a selection other than (R)ock, (P)aper, or (S) cissors.
2	Computer displays an error message stating what are valid options.
3	Computer prompts player to make a new selection.
Alternate Postconditions	
1.	Player has been told proper selections.
2.	Player can make a new selection.

9. Non-functional Requirements
1. Only one input and output at a time. 2. Computer cannot make a selection based on the player's selection.

10. Glossary
ASCII - The **American Standard Code for Information Interchange.** ASCII codes represent text in computers, communications equipment, and other devices that use text.

Use Case Description 2

Use Case name:	Player quits mid game
Project name:	Rock Paper Scissors
Team:	Hypnocode
Date:	2/15/2012

1. Goal
Player is able to quit the game at any time.

2. Summary
Player can at any point during a game select to go back to the main menu and quit the game.

3. Actors
Actor 1: Player

4. Preconditions
1. The program is launched.

5. Trigger
Player selects return to main menu.

6. Primary Sequence	
Step	**Action**
1	Player selects return to main menu
2	Computer display a message asking if the user really wants to quit
3	Player selects yes they want to quit
4	Computer displays main menu
5	Player selects quit game

7. Primary Postconditions
1. Computer displayed quit warning to player.
2. Program ends

8. Alternate Sequences	
Alternate Trigger	Player selects No they do not wish to quit the game
Step	**Action**
1	Computer display a message asking if the user really wants to quit
2	Player selects No they do not wish to quit
3	Player continues playing
Alternate Postconditions	Player continues as if nothing happened

9. Nonfunctional Requirements
N/A

10. Glossary
N/A

Use Case Description 3

Use Case name:	Player asks for help.
Project name:	Rock Paper Scissors
Team:	Hypnocode
Date:	2/15/2012

1. Goal
Gives player help message.

2. Summary
Player enters help command and game displays help.

3. Actors
Actor 1: Player

4. Preconditions
1. The program is launched.

5. Trigger
Player enters help command.

6. Primary Sequence	
Step	**Action**
1	While playing a match player selects help.
2	Help message is displayed.
3	Computer prompts player to press a button to continue match.

7. Primary Postconditions
User has been shown help message

8. Alternate Sequences	N/A

Alternate Trigger	N/A
Step	**Action**
Alternate Postconditions	N/A

9. Nonfunctional Requirements
N/A

10. Glossary
N/A