# 1. Short Description of Project

1. Title: **Mastermind Game**
2. Name of team members: Ivanne Sanchez, Omar Ahmad Mohssen, Maywadee Soytong
3. Text with description-outline of the characteristics of the software

Mastermind is a puzzle game, in which one player tries to guess the code his opponent comes up with.

**How to play mastermind :**

The computer has a secret combination of colors (usually randomly chosen).

The player has to guess it. For that the player proposes a combination of colors and the computers tells him how many colors are correct (right color on the right place) and how many are just misplaced (right color but in the wrong place). Usually, A maximum number of proposals possible for the player in this game is 10.

The collection of colors for this game are Green Yellow Red White Black Pink.

**For example**, the pattern that the game randomly generating is “Yellow, White, Red, Black”.

If the player guess like “White, Red, Pink, Green”.

Then the game will give player the clue, “2 matching colors 0 matching place”.

**The goal of our mastermind :**

The idea is to implement a program that choose a color combination randomly and then, for each proposal of the player, gives the number of well-placed colors and the number of right colors, but misplaced.

1. Brief on main functional and non-functional requirements

**Functional requirements:**

1. Console system

a). initial the game

b). start the game

c). show collection of the colors

d). call game logic

e). call score service

2. System of game logic

a). random the secret pattern

b). check well place

c). check matching colors

d). scoring

e). ranking

f). call score management system

3. Score management system

a). read old top ten

b). write new top ten

**Non-functional requirements:**

**1**. Reliability – 7 days/week, 24 hours/day

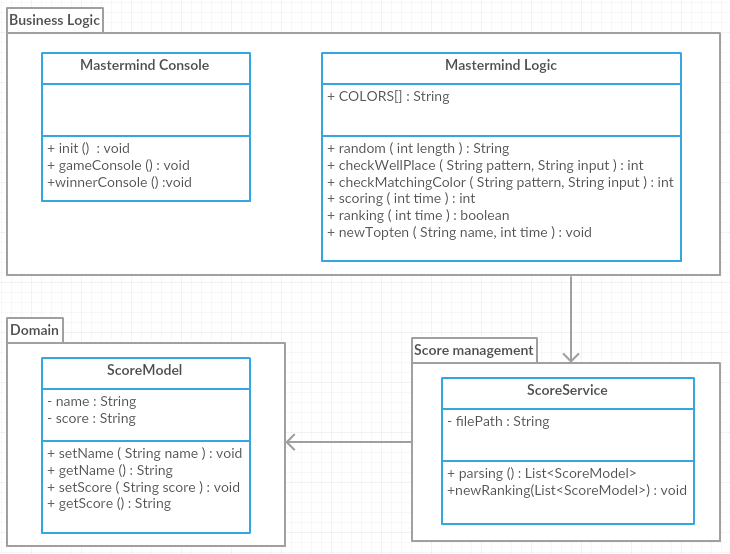
2. Usability – the system can be easily used by player

3. Capacity – the system store only score and name of the top ten

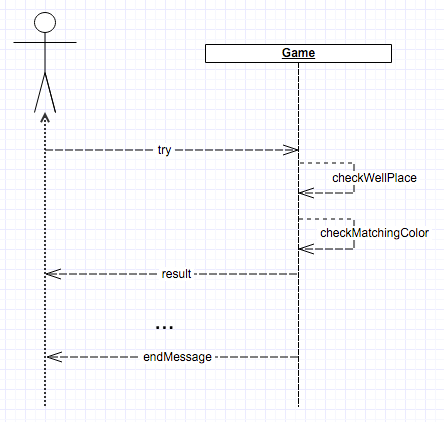
1. Short description of the main components using some UML notations.

We are planning on using 2 components:

* a game console component
* a score management component
* a domain component

Below is a diagram of the game layers. A first layer contains all game elements and associated to it is a business logic layer. Another layer, The score management file is also in charge with the retrieval and data storage the top ten score with the domain information.

# 2. Formal verification of component(s) design

**1). Play the game**

**Promela code:**

#define iSize 100

mtype = {

try,

checkWellPlace,

checkMatchingColor,

result,

endMessage

}

chan PlayerInterface = [iSize] of { mtype }

proctype Player() {

int stage;

Start: atomic {

printf("\n Player start \n");

goto Next1;

}

Next1: atomic {

PlayerInterface!try;

goto Next2

}

Next2: atomic{

byte nr; /\* pick random value \*/

do

:: nr++ /\* randomly increment \*/

:: nr-- /\* or decrement \*/

:: break /\* or stop \*/

od;

printf("-end game-");

goto Start;

}

}

proctype Interface () {

int n = 0;

int endGame = 0;

Start: atomic {

printf("\n Interface start \n");

PlayerInterface?try;

if

:: (n < 5) ->

PlayerInterface!checkWellPlace;

PlayerInterface!checkMatchingColor;

PlayerInterface!result;

n = n + 1;

PlayerInterface?try;

:: (n == 5) ->

PlayerInterface!endMessage;

goto Start;

fi

}

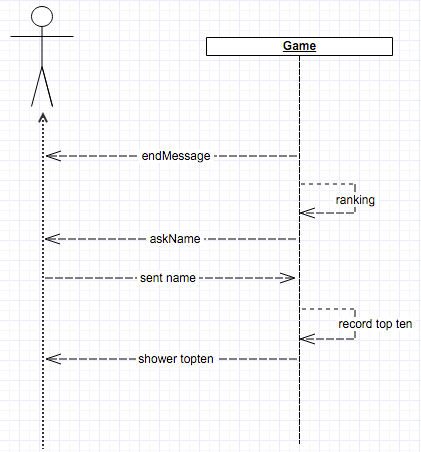
}

init {

run Player();

run Interface();

}

**2). Win the game and hit the top ten score**

**Promela code:**

#define iSize 100

2

3 mtype = {

4 endMessage,

5 ranking,

6 askName,

7 recordTopten,

8 showTopten,

9 }

10

11 chan PlayerInterface = [iSize] of { mtype }

12

13 proctype Player()

14 {

15 Start: atomic {

16 printf("\n End Game \n");

17 PlayerInterface!ranking;

18 goto Next1

19 }

20

21 Next1: atomic {

22 printf("\n Player Next2 \n");

23 PlayerInterface!askName;

24 }

25

26 Next2: atomic {

27 printf("\n Player Next3 \n");

28 goto Start;

29 }

30 }

31

32 proctype Interface ()

33 {

34 int n = 0;

35 Start: atomic {

36 printf("\n Interface start \n");

37 PlayerInterface?askName->

38 PlayerInterface!recordTopten;

39 goto Next1;

40 }

41

42 Next1: atomic {

43 printf("\n Record Top Ten \n");

44 goto Next2;

45 }

46 Next2: atomic{

47 printf("\n show Topten \n");

48 }

49

50 }

51

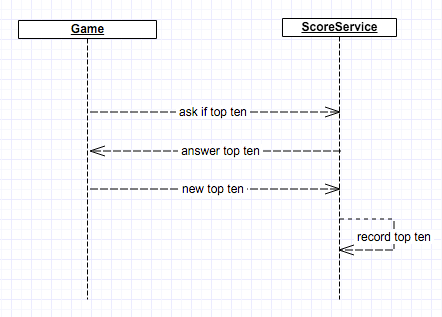
52 init {

53 run Player();

54 run Interface();

55 }

56

**3). Write new score**

**Promela code:**

#define iSize 100

2

3 mtype = {

4 askIfTopten,

5 answerTopten,

6 newTopTen,

7 recordTopten,

8

9 }

10

11 chan PlayerInterface = [iSize] of { mtype }

12 chan InterfaceScoreFile = [iSize] of { mtype }

13

14

15

16 proctype Interface ()

17 {

18 int n = 0;

19 Start: atomic {

20 printf("\n Interface start \n");

23 }

24

25 Next1: atomic {

26 printf("\n Interface Next1 \n");

27 InterfaceScoreFile!askIfTopten;

28 goto Next2;

29 }

30

40 }

41

42 proctype ScoreFile ()

43 {

44 Start: atomic {

45 printf("\n ScoreFile start \n");

46 InterfaceScoreFile?askIfTopten;

47 InterfaceScoreFile!askIfTopten;

48 goto Next1;

49 }

50

51 Next1: atomic {

52 printf("\n ScoreFile Next1 \n");

53 InterfaceScoreFile?recordTopten;

54 printf("\n store score \n");

55 /\* goto Start; \*/

56 }

57 }

58

59 init {

60 run Interface();

61 run ScoreFile();

62 }