

March 15, 2019

ECSE-223-001

Model-Based Programming

Group P22

Guevorkian, Andrei - 260 737 896

Kim, Younggug - 260 531 176

Kinman, Garrett - 260 763 260

Mircic, Michael - 260 587 925

Nuviadenu, Edem Koshi - 260 779 440

Smith, Sean - 260 787 775

Iteration 4 Report

## Table of Contents

<b>1-Start/Pause/Resume Game - Younggue Kim</b>	<b>2</b>
<b>2-Move Ball - Michael</b>	<b>4</b>
<b>3-Ball Hit Paddle or Wall - Nuviadenu Edem Koshi</b>	<b>5</b>
<b>4-Ball Hits Block - Garrett Kinman</b>	<b>8</b>
<b>5-Ball is Out of Bounds - Andrei Guevorkian</b>	<b>10</b>
<b>6-View Hall of Fame - Sean Smith</b>	<b>11</b>
<b>7-Test Game - Michael</b>	<b>12</b>
<b>9-Move Paddle - Team</b>	<b>14</b>

## 1-Start/Pause/Resume Game - Youngguk Kim

\*The outer for loop is the game controller loop

\*I used if and break statement to pause: break out of the game loop

\*in the sequence diagram, setWaitTime() is missing.

public static void startGame(String name) throws InvalidInputException

public Game getPlayableGame()

public String moveBall()

public Level getPlayableLevel()

public Block getPlayalbeBlock()

public Paddle movePaddleLeft()

public paddle movePaddleRight()

public void leftKeyPressed()

public void rightkeyPressed()

public void spaceBarPressed()

public Game pausedGame()

public int getPlayerLives()

public boolean getFinishedLevel()

public boolean getPausedStatus()

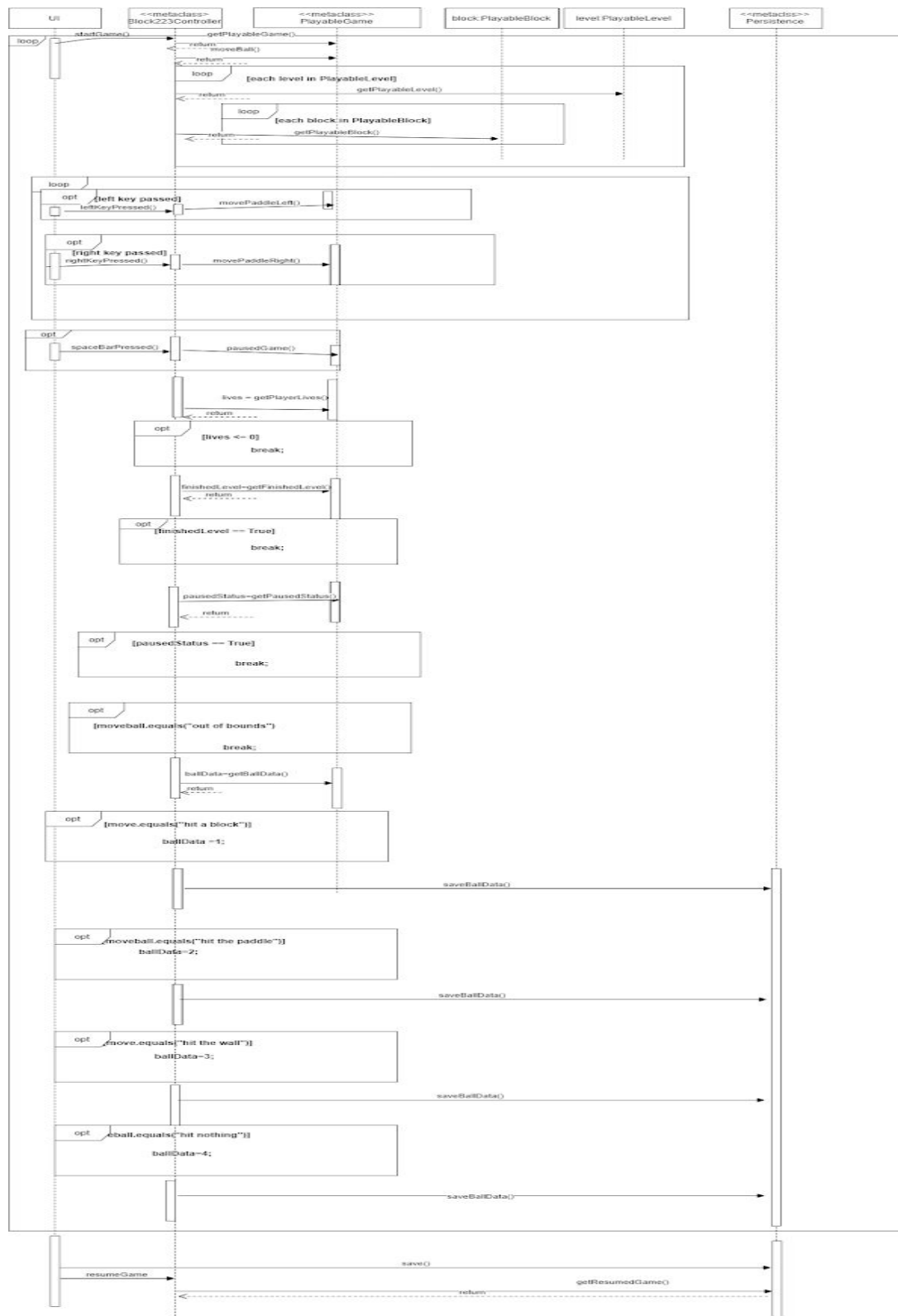
public int getBallData()

public void saveBallData()

public void save()

public Game getResumedGame(String name) throws InvalidInputException

public Time setWaitTime()



## 2-Move Ball - Michael

Ball movement will be a part of continuous execution once a game begins.

The first method is:

**public void moveBall()**

This method will tell the Ball object to update its X and Y positions based on its current movement values. Queries for this method are:

From Game:

**public Ball getBall()**

**public getNewBallPos()**

From Ball:

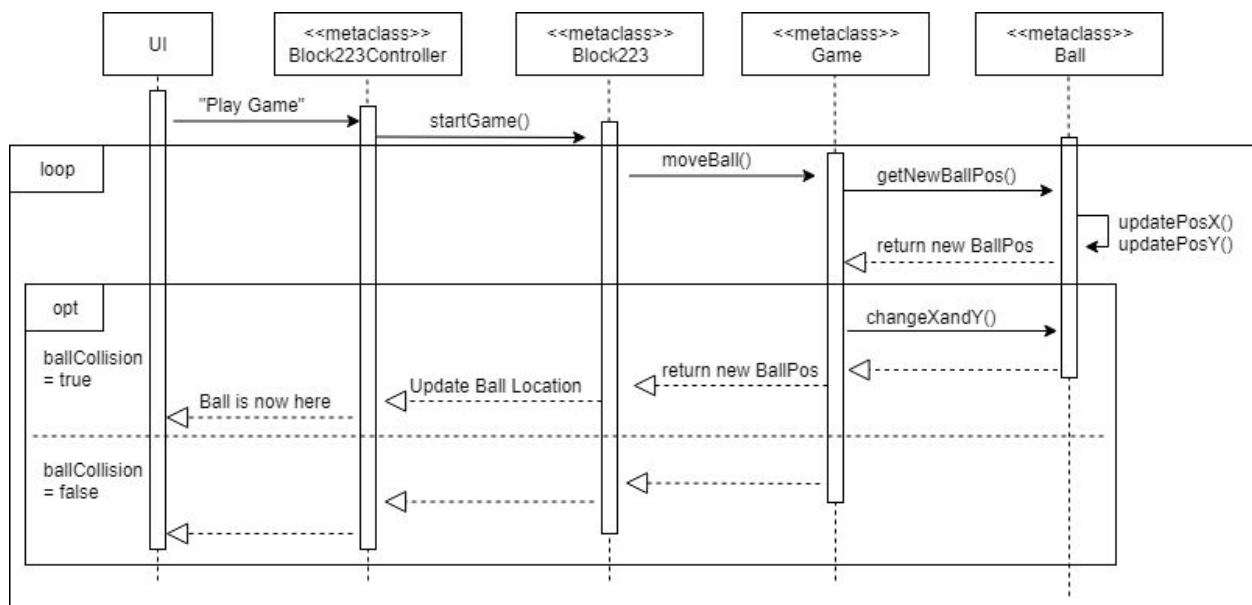
**public updateDirectionX(double x)**

**public updateDirectionY(double y)**

A second method may also be required if the ball collides with another solid object:

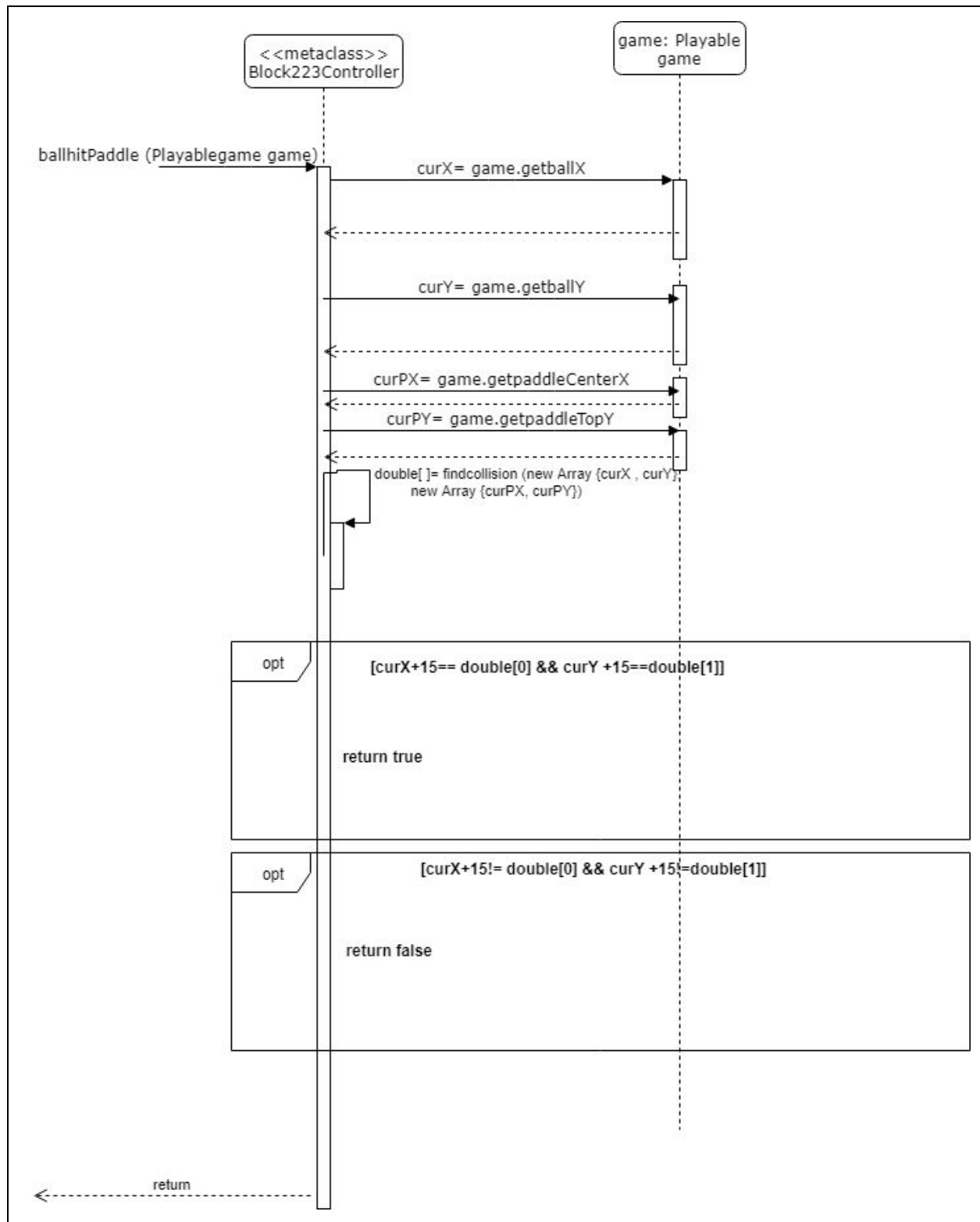
**public void changeXandY(double x, double y)**

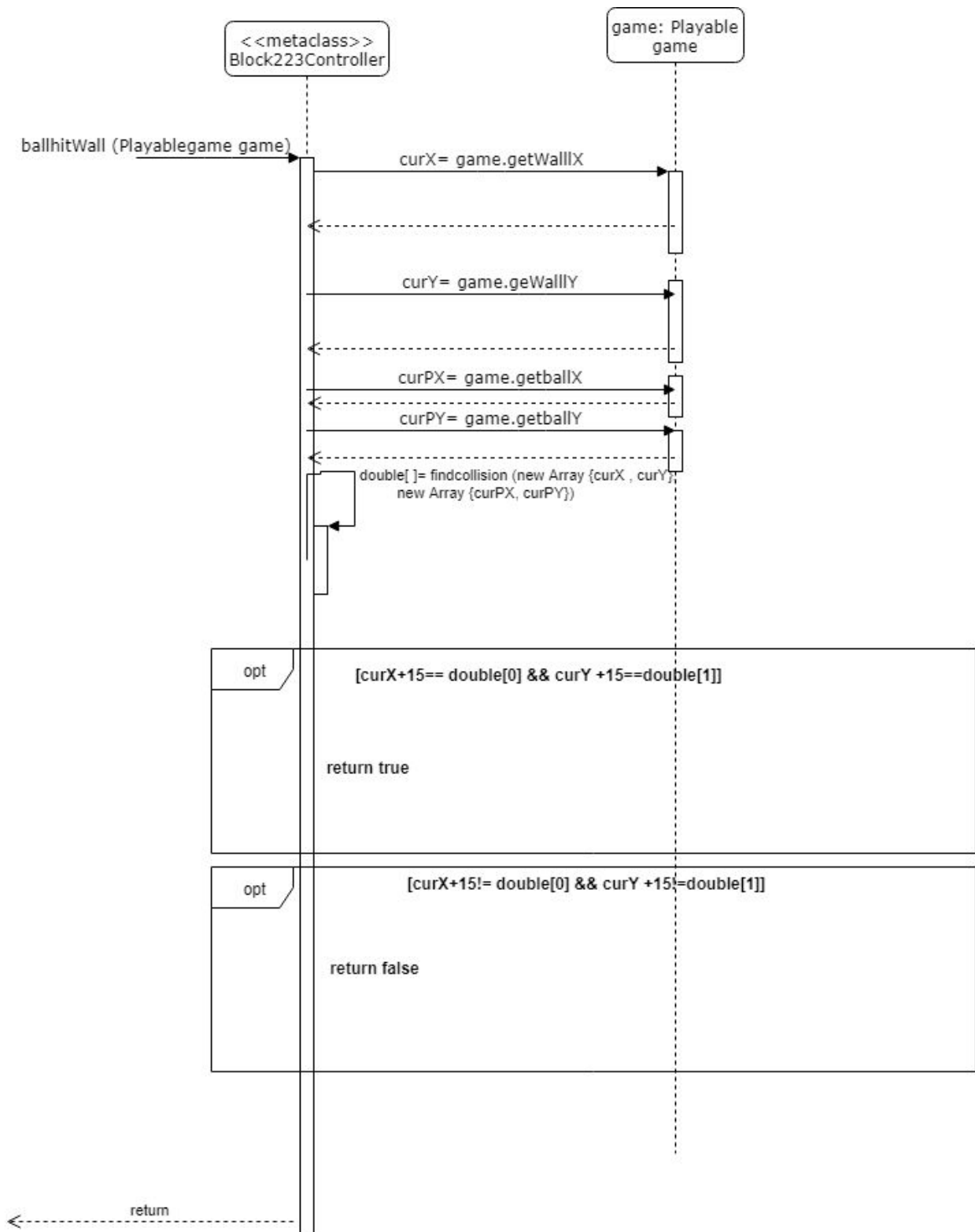
Move Ball Sequence Diagram



### 3-Ball Hit Paddle or Wall - Nuviadenu Edem Koshi

Block223Controller Interface
Public static boolean ballhitPaddle( Playerablegame game)
public static double[ ] findCollisionPoint(int[2] p1, int[2] p2, int[2] p3, int[2] p4)



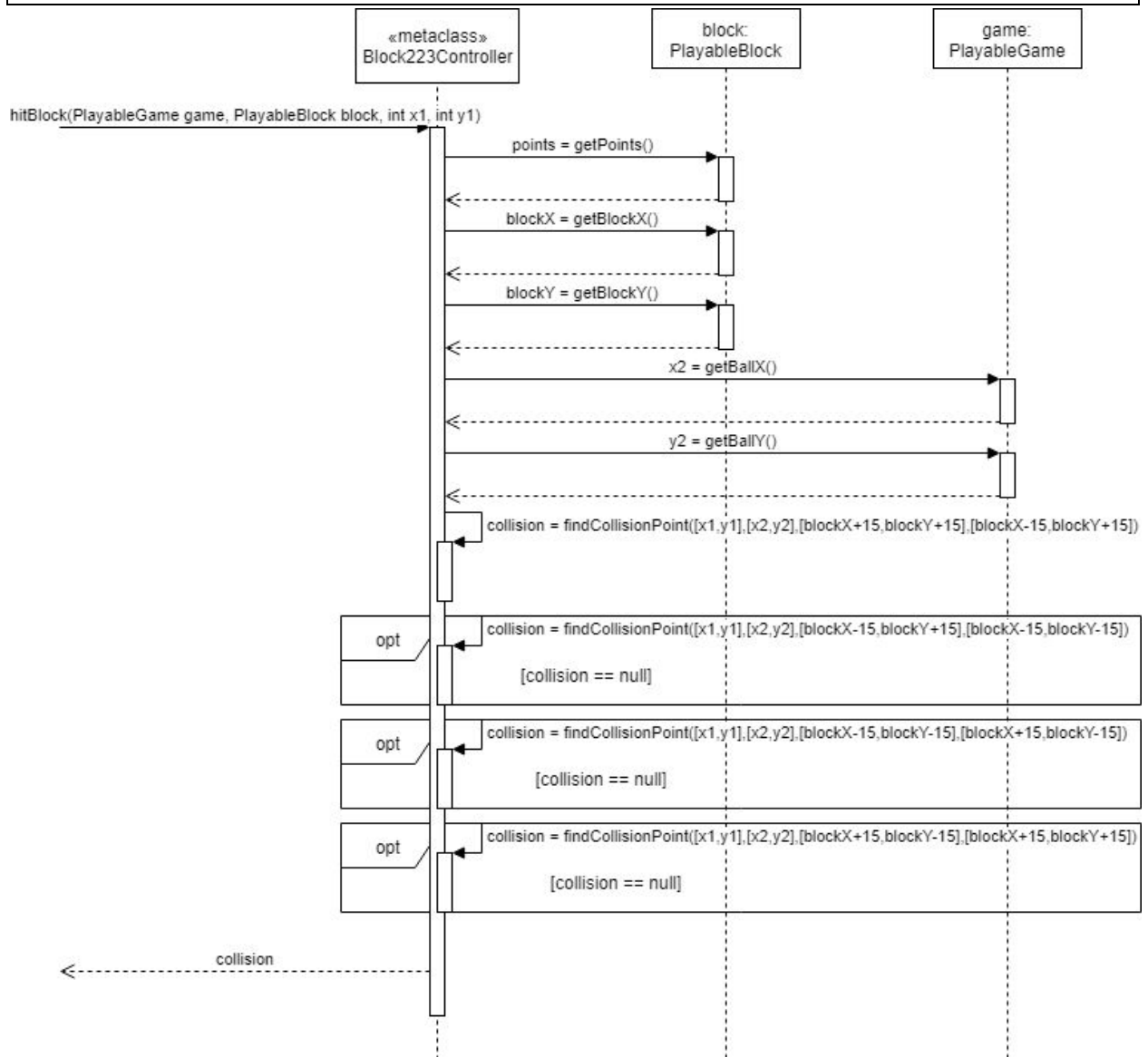


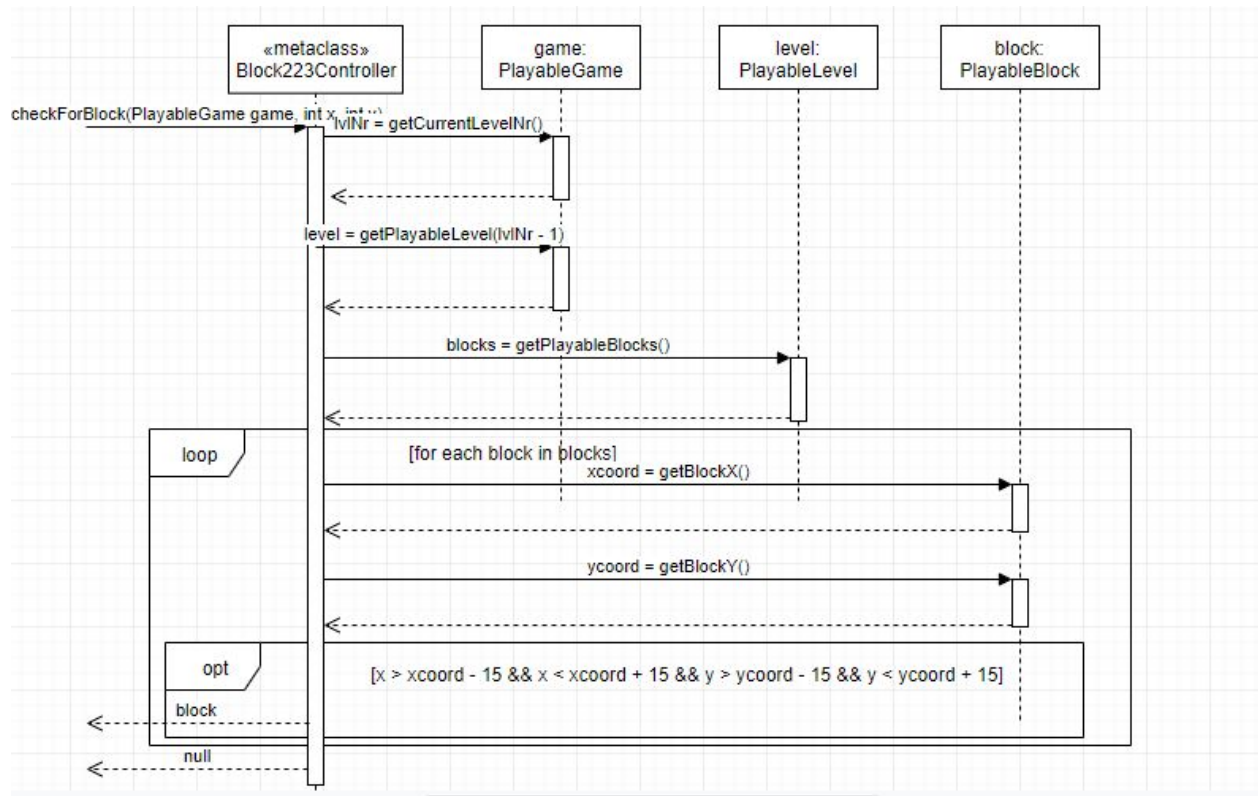


## 4-Ball Hits Block - Garrett Kinman

```
public static PlayableBlock checkForBlock(PlayableGame game, int x, int y)
```

```
public static void hitBlock(PlayableGame game, PlayableBlock block, int x,  
int y)
```





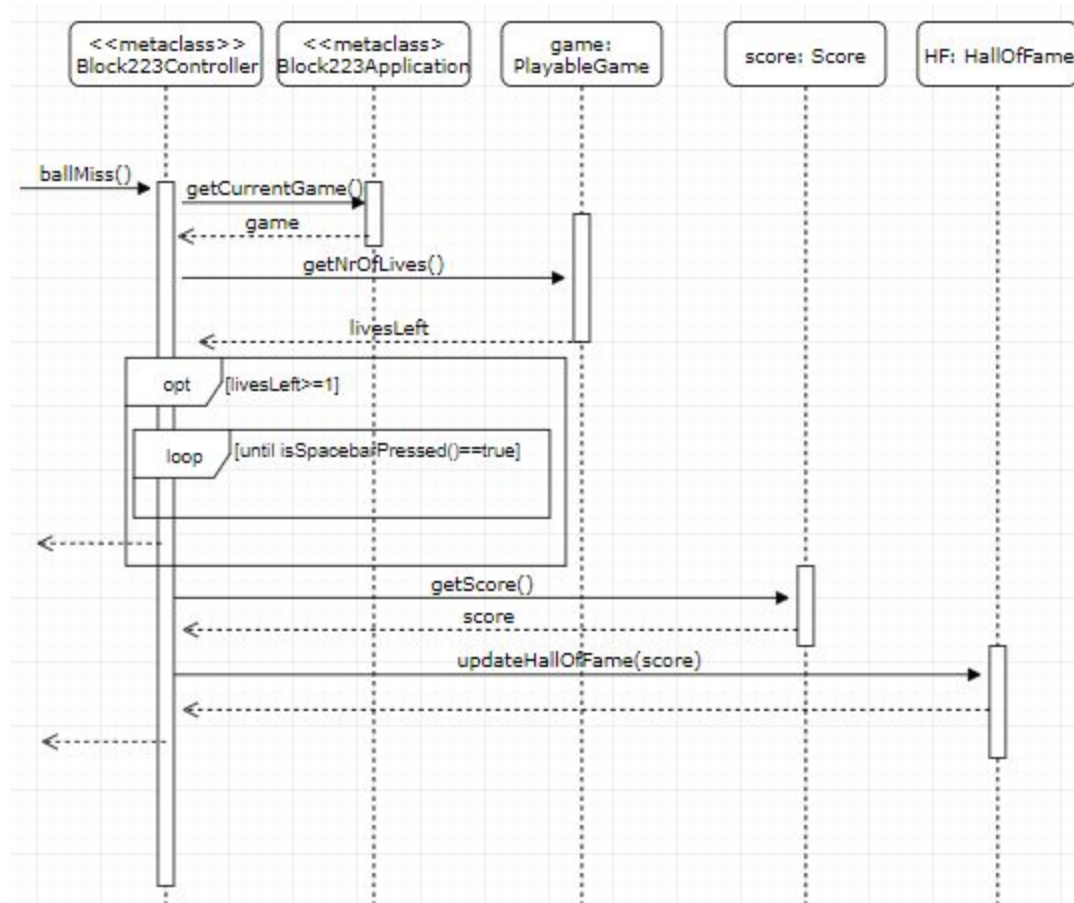
## 5-Ball is Out of Bounds - Andrei Guevorkian

Use public static int getBallY(PlayableGame game) to see that ball is out of bounds.

Then, game is paused (view pauseGame()).

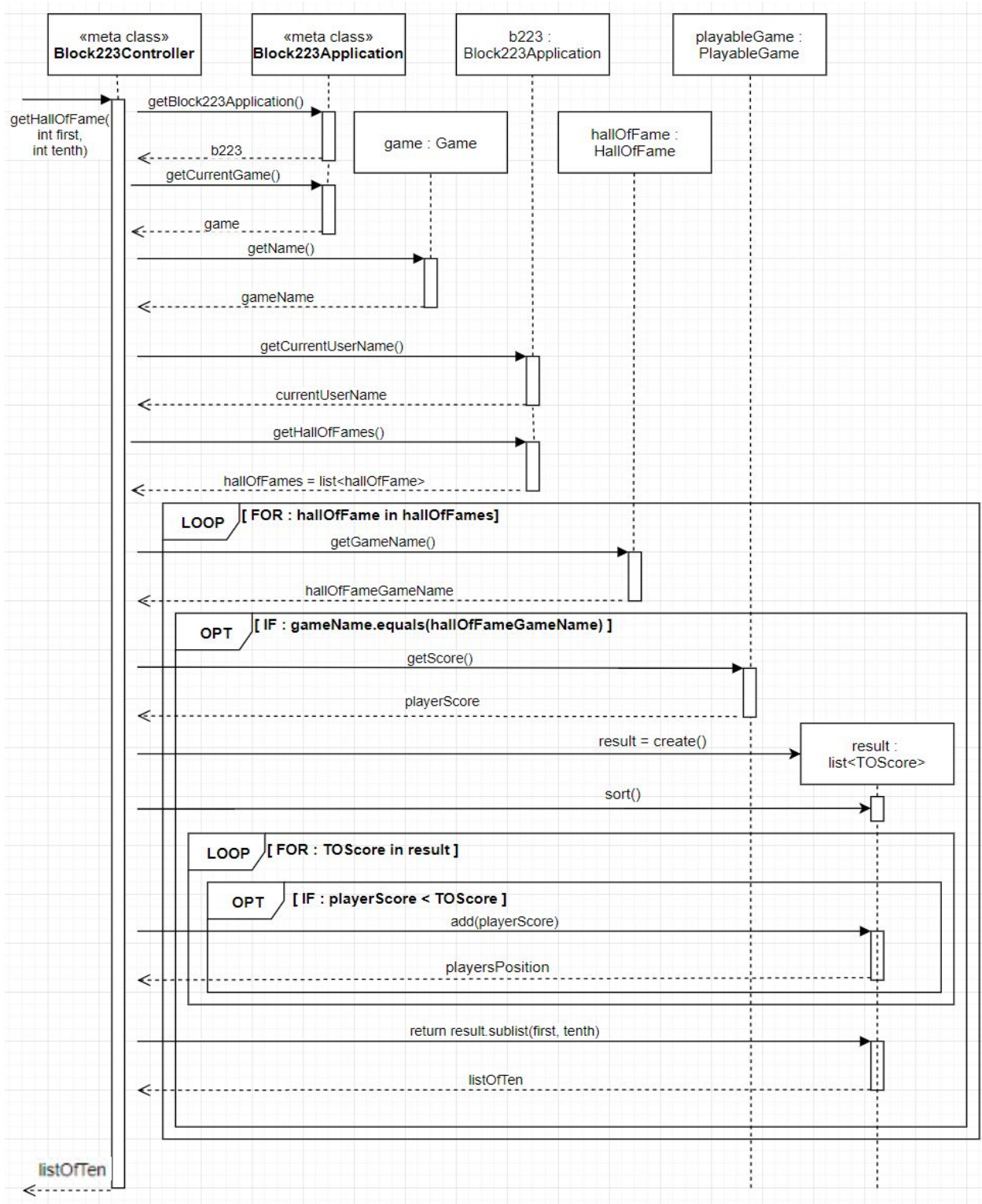
Then, wait for user to press on spacebar key, after which resume game (see resumeGame()).

Public static void ballMiss()



Throw Invalid Input Exception	Condition	Error Message
<code>getNrOfLives()</code> (catch <code>indexOutOfBoundsException</code> and rethrow)	<code>livesLeft</code> is above 3 or below 1.	"You didn't have a legal amount of lives left"

## 6-View Hall of Fame - Sean Smith



View Hall of Fame will be called in the Block223 view in order to display a list of 10 scores ("listOfTen") which are returned and added to a component of the JFrame. Also in the view class, the getUsername() method will be called on the associated scores according to the values passed to viewHallOfFame() method.

NOTE : in viewHallOfFame(int first, int tenth){...} method, the int variable "tenth" will be replaced with "last" to avoid confusion for when there are less than 10 players in the hall of fame.

Throw Invalid Input Exception	Condition	Error Message
Upon calling method	First and tenth are invalid entries	" An error occurred when viewing hall of fame. "

## 7-Test Game - Michael

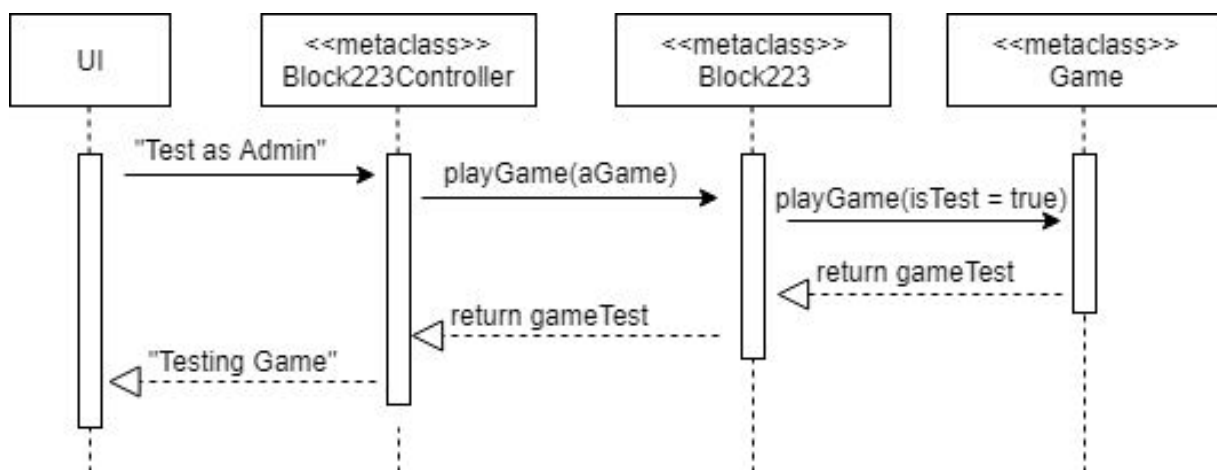
Testing a game will be accessible from the admin menu.

It uses the same method to play a game, namely:

**public void playGame(aGame)**

However, a flag will also be set to the game marking it as a test. As such, it will be missing some key features of a regular game.

Test Game Sequence Diagram



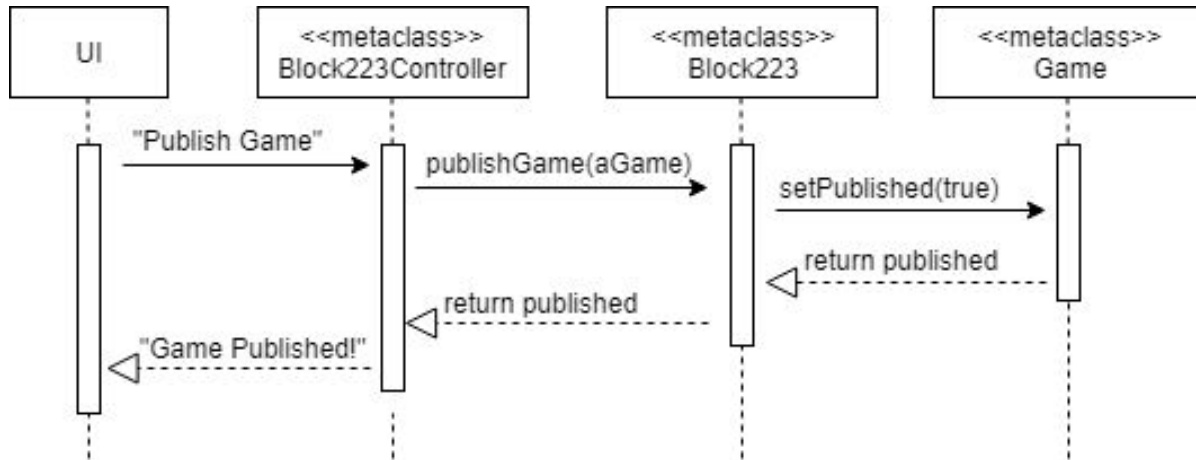
## 8-Publish Game (Michael)

A game can be published from the admin menu using the method:

**public boolean publishGame(aGame)**

This method will also trigger a flag variable in the game which will set it as published, not allowing any further modifications.

Publish Game Sequence Diagram



## 9-Move Paddle - Team

Left and right arrow keys converted to chars 'a' and 'b' (respectively) which are added to a queue.  
 Moves left while queue has a and stops when reaches left wall.  
 Moves right while queue has b and stops when reaches right wall.  
 When player changes direction before it hits the wall, dequeue until empty and add new char to queue.

