FINAL YEAR PROJECT

The Breakout

Team Members

Shirza Firtas (14-0100)

Nimra Imran (14-0161)

Sajid Khan (14-1650)

Supervised by

Dr. Hasan Mujtaba

Department of Computer Science,

National University of Computer and Emerging Sciences,

Islamabad, Pakistan

2017

Intellectual Property Right Declaration

This is to declare that the work under the Title **"The BreakOut"** carried out in partial fulfilment of the requirements of **BS FYP** is the sole property of the **National University of Computer and Emerging Science**, and is protected under the intellectual property right laws and conventions. It can only be considered/used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University

This above statement applies to all students and faculty.

Date: _____

Member 1
Name: **Shirza Firtas(14-0100)**

Signature:_____

Member 2
Name: **Nimra Imran(14-0161)**

Signature:_____

Member 3
Name: **Sajid Khan(14-1650)**

Signature:_____

Supervisor (Faculty)
Name: **Dr. Hasan Mujtaba**

Signature:_____

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: The BreakOut is the sole contribution of the author(s) and no part hereof has been reproduced on as it is basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: _____

Member 1
Name: **Shirza Firtas(014-0161)**

Signature:_____

Member 2
Name: **Nimra Imran(014-0161)**

Signature:_____

Member 3
Name: **Sajid Khan(014-1650)**

Signature:_____

Table of Contents

INTRODUCTION

Gaming business, also known as Interactive Entertainment industry, has been developing, advancing, and innovating with time. This, in turn has created opportunities and employments for aspiring developers, professional players and casual gamers not restricted by age, culture or gender.

This project delves into 2D android game development using Unity3D as a development platform. Applications of Artificial Intelligence (AI) in this area is an exciting new prospect for many developers. It has created a new realm of interaction between players and the environment. Using C/C#, our aim is to implement a form of intelligence that will increase the difficulty of Non-Player Characters by the end of the project and by introducing the concept of virtual reality in our game, we are enhancing the interactive experience for player.

Competitive as it is, three major platforms; namely the PlayStation series, the XBox iterations and Desktop Computers; have largely lead the way for new games. Recently though, iOS, Android and Windows (phone) have also gained popularity amongst developers and gamers alike.

Real time 2D recreations have existed for almost a decade. However, up until recently, the expense of authorizing one of the chief gaming engine has run from a few hundred thousand to a few million dollars for each title, consigning the fantasy of making one's own 2D amusement games to an unattainable dream. Unity provides a vigorously highlighted free form of their platform, which radically changes the estimating models for top of the line games or Computer Graphics (CG). Unity3D provides a uniform and concise way to produce animations and maps. These animations and maps can be made interactive via the engine and are supported by Web, PC, Mac, iOS, Flash, Android, Xbox360, PS3 and Wii. We aim to translate this utility into a functional First-Person Shooting game.

Improvements in Hardware (i.e. graphic cards, sound cards and processors) have contributed a great deal in new horizons of gaming industry to new levels of success. Modern games in computers, consoles and even mobiles use high power and require high graphics.

PROJECT VISION

1. PROBLEM STATEMENT

The saturation in the gaming industry has increased subsequently, that results in the demand of more advanced and modernise games. Since gaming industry is increasing quickly, locally and internationally. The main focus while developing a game is on the graphics which may provide an exciting user experience but compromising the replay-ability factor resulting in a collaborated and less interactive user experience.

2. BUSINESS OPPORTUNITY

In commercial games, such as action games, role-playing games, and strategy games, the behaviour of the NPC non player character is usually implemented as a variation of simple rule-based systems. With a few exceptions, machine-learning techniques are hardly ever applied to state-of-the-art computer games. Hence, by implementing AI techniques we may enable the NPCs with the capability to improve their performance by learning from mistakes and successes, to automatically adapt to the strengths and weaknesses of a player, or to learn from their opponents by imitating their tactics, thus increasing replay-ability factor and user engagement.

3. AIM & OBJECTIVES

We are developing a cross platform Unity 2D based game that integrates Artificial Intelligence for improving and enhancing "*replay-ability*" and the involvement of human factors. The main objective of the game is for the player to solve the puzzle, kill the bots existing on the level by utilizing the weapons that will be available for the help.

Moreover we look forward to fulfil the following requirements:

- Implement Artificial Intelligence
- User friendly and effective interface
- Make AI to learn from the patterns and work accordingly
- A "*replay-ability*" game
- Properly allocated time and resources

4. PROJECT SCOPE

The project will involve designing and development of a 2D unity survival game for providing an interactive and friendly user experience by enhancing the "*replay-ability*" factor. By using Artificial Intelligence, the patterns for the movement will be studied that will further help in knowing the weak and strong points resulting in the formation of diverse traps.

SOFTWARE REQUIREMENT SPECIFICATION

1. LIST OF FEATURES

1. **Game character:**
   Player plays the game through a character's viewpoint.

2. **Player quantifier (Score):**
   The score for current and past players is saved.

3. "***Replay-ability***" **factor:**
   Different puzzles are assigned randomly to the player. Player solves a new puzzle every time, increasing its replay-ability.

4. **Movement:**
   Player can move freely in the 2D environment.

5. **Different levels of difficulty:**
   The difficulty increases with each level. The number of bots will be increased with each level.

2. FUNCTIONAL REQUIREMENTS

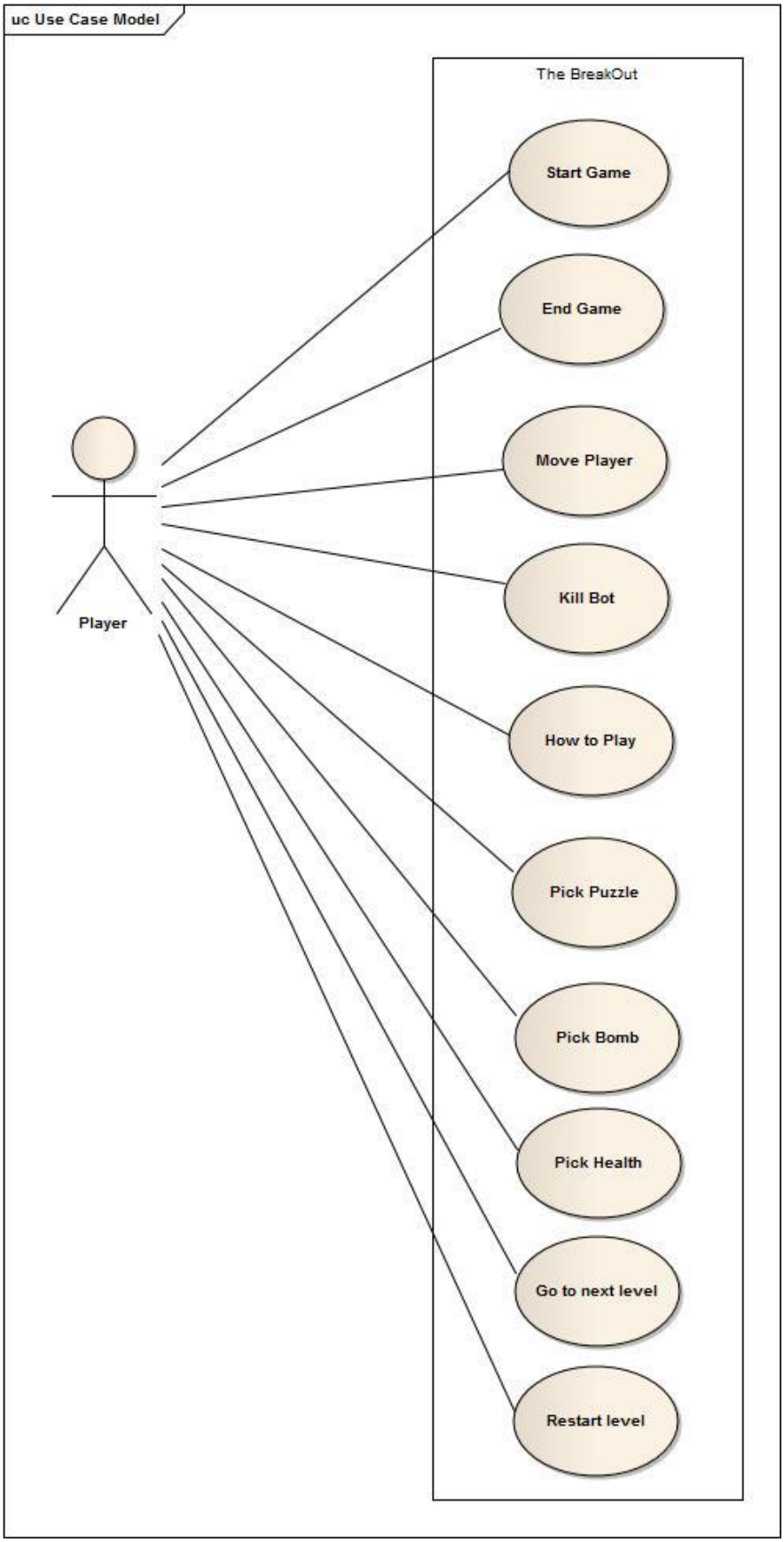
- When the game starts, the system must assign the puzzle.
- The system ends the game once bots kill the player.
- The system should outline the levels and create the environment according to that.
- The system shall assign the puzzle randomly to the player.
- The system should response to the movements made by the player within seconds.
- The system should response to the game controls properly.

HIGH LEVEL USE CASES

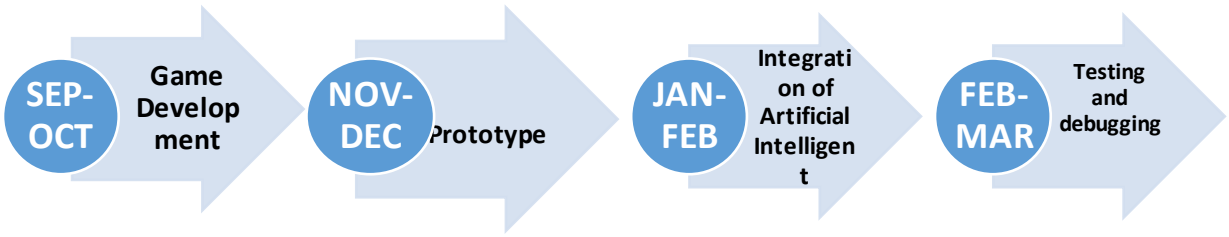
S.no	Use case Name	Actors	Type	Description
1	Start Game	Player	Primary	The system starts the game along with releasing puzzles, maps & environment variables
2	Exit Game	Player	Primary	The system lets the user to quit the game.
3	Kill bot	Player	Primary	Whenever a bot comes into player's way, player can kill it with the assigned weapon.
4	Move Player	Player	Primary	Player can move in the environment freely.

5	How to Play	Player	Primary	Player can select the how to play option to take help.
6	Pick Puzzle	Player	Primary	Player picks the puzzle to move to the next level.
7	Pick Bomb	Player	Primary	Player picks up the bomb to kill multiple bot with in an explosion radius.
8	Pick Health	Player	Primary	Player picks up the health to increase its strength.
9	Go to Next Level	Player	Primary	Player selects the next level option to go to the next level.
10	Restart Level	Player	Primary	Player selects the restart option to start the current level again.

USE CASE DIAGRAM



ITERATION PLAN



Our first step is the development where all the documentation along with a model for game will be made according to our requirements. With the base of that we will make a prototype for our project. After that, our main test will come that is the implementation of artificial intelligence for judging the environment, assign different puzzle to the player and killing of bots. At the end, with a complete project in hand, we will test it for any inconveniences and further improve them.

ITERATION PLAN

ITERATION 1

1. EXPANDED USECASES

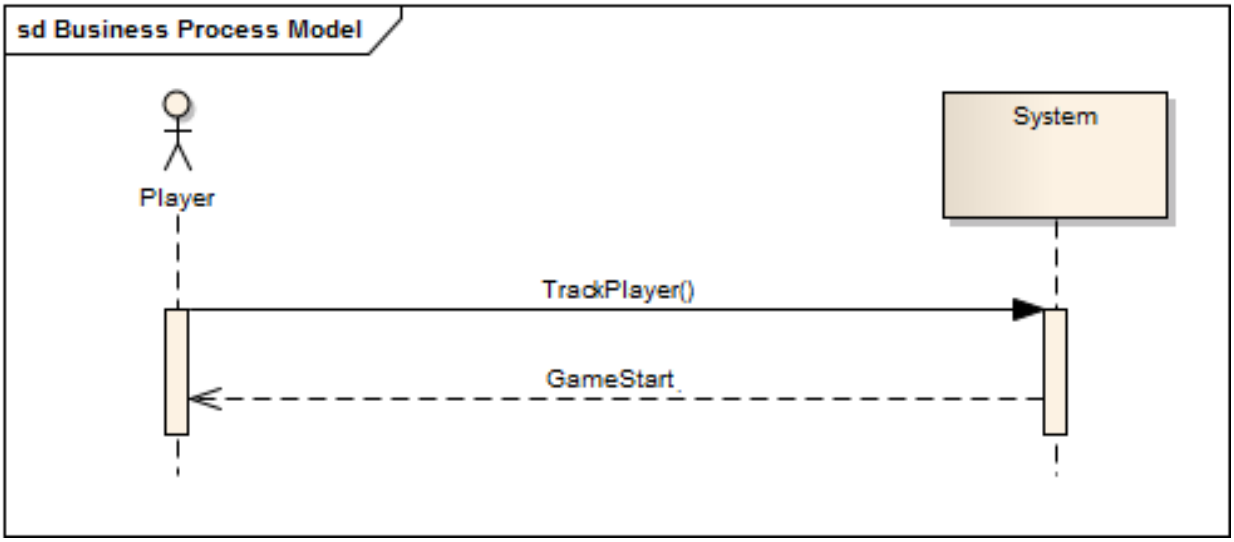
USE CASE NAME	**EXIT GAME**
Scope	The BreakOut
Level	User goal
Primary actors	Player
Stakeholders & Interest	Player wants to exit the game
Pre-Conditions	User is playing the game
Post Conditions	The game will end.
Main Success Scenario **Actions** 1. Player requests to exit the game.	**System Response** 2. Systems quits every operation. 3. System saves the score and exits the game.
Extensions	Nil
Special Requirements	Game executes properly in the environment
Technology and Data Variation List	Nil
Frequency of occurrence	Nil

USE CASE NAME	START GAME
Scope	The Breakout
Level	User goal
Primary Actors	Player
Stakeholders & Interest	Player wants to start a game
Pre-Conditions	Game is not in the play environment
Post Conditions	Game environment is initialized Puzzle is assigned to the player
Main Success Scenario **Actions** 1. Player requests to start the game	System Response 2. System constructs map and puzzle 3. Player character is generated
Extensions	Nil
Special Requirements	System is functioning and running
Technology and Data Variation List	Nil
Frequency of Occurring	Nil

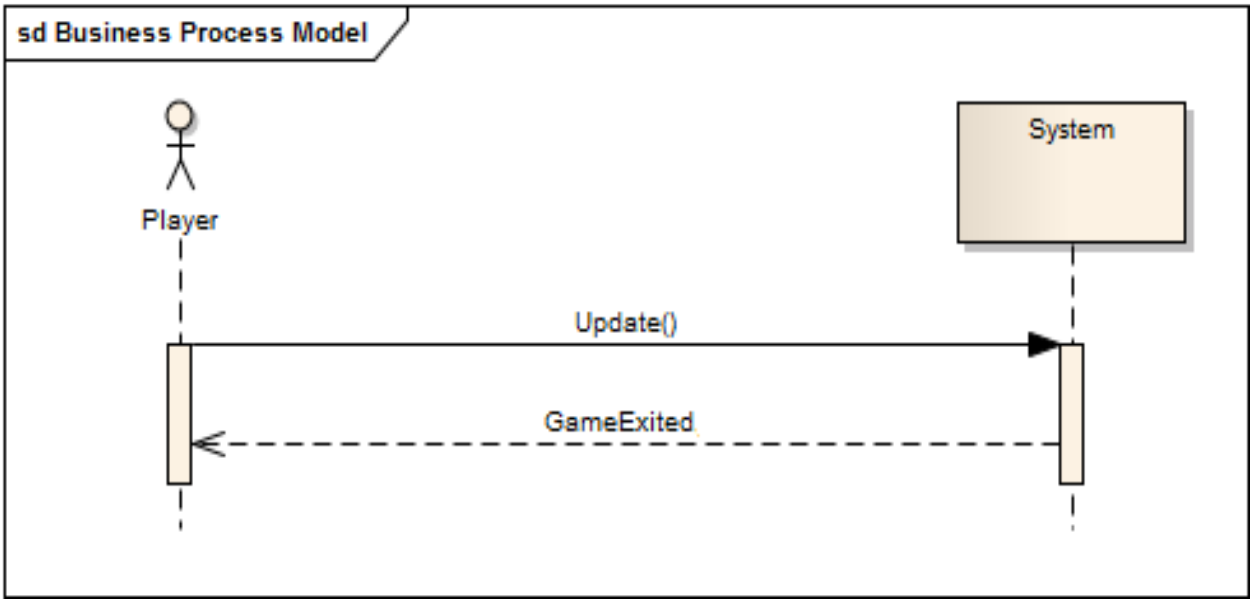
USE CASE NAME	MOVE PLAYER
Scope	The Breakout
Level	User Goal
Primary Actors	Player
Stakeholders & Interest	Player gives move instruction to character
Pre-Conditions	Player in Game Character in Map Move instruction given
Post Conditions	Character moves according to instruction
Main Success Scenario **Actions** **1.** Player requests movement through an input key.	**System response** 2. System performs the required movement.
Extensions	nil
Special Requirements	Game is in execution mode.
Technology and Data Variation List	Nil
Frequency of Occurring	As per user's demand

3. SYSTEM SEQUENCE DIAGRAM

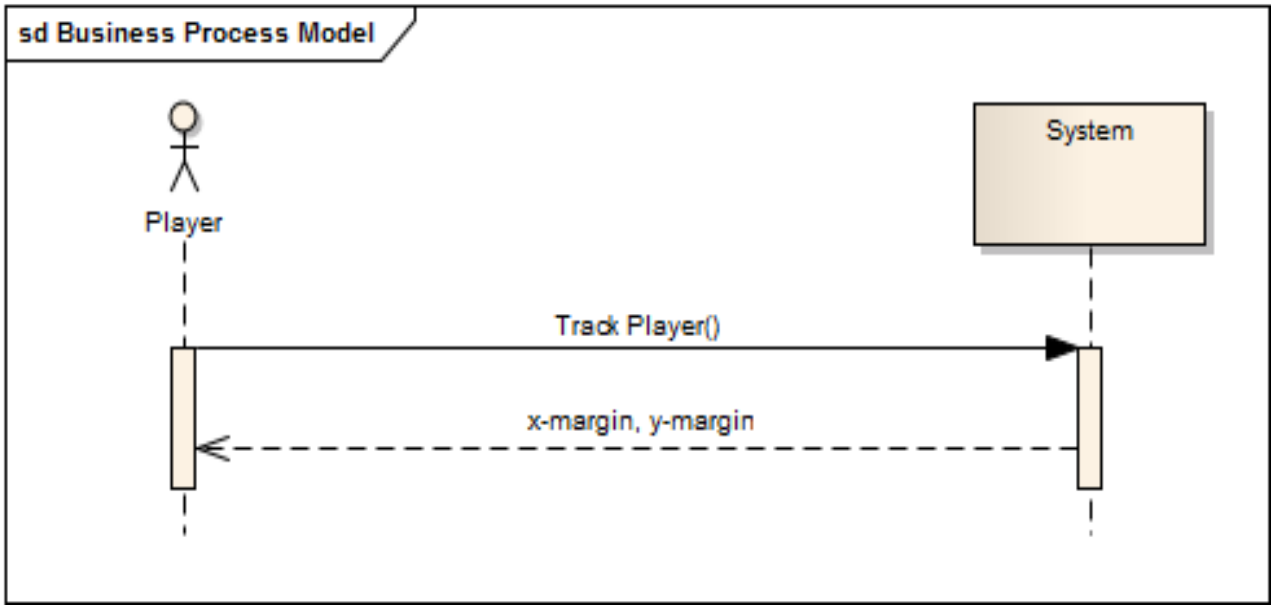
- ***START GAME***



- ***EXIT GAME***



- ***MOVE PLAYER***



4. OPERATION CONTRACTS

Operation Contract # 1

Name: TrackPlayer();

Responsibility: It starts the game, assign puzzle, create bots, and displays the whole interface**.**

Type: System

Cross Reference: UseCase1 Start game

Exceptions: Nil

Pre-Conditions: System supports the game configuration.

Post Conditions:

1. An instance trackPlayer for start game was generated.
2. It was associated with puzzle and bots.
3. X and y margin were modified.

Operation Contract #2

Name: ExitGame();

Responsibility: Let the player to leave from the game.

Type: System

Cross Reference: UseCase2 Exit Game

Exceptions: nil

Pre-Conditions: Player has entered the game process.

Post Conditions:

1. An instance trackPlayer for exit game was generated.
2. It was associated with player controller and bots.
3. Update was destroyed

Operation Contract #3

Name: Get_inputkey();

Responsibility: Accountable for movement.

Type: System

Cross Reference: UseCase3 Move Player

Exceptions: nil

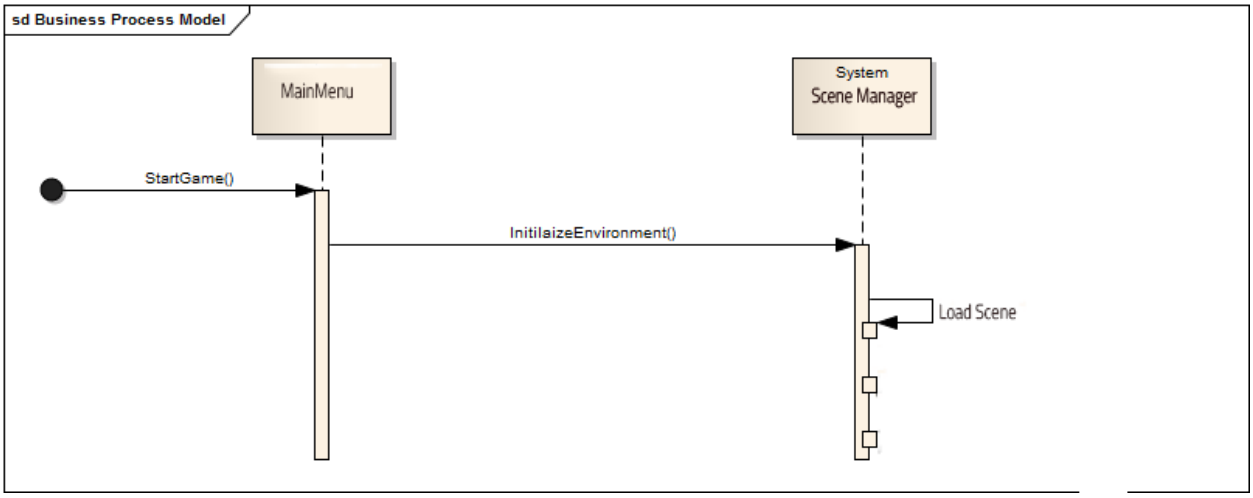
Pre-Conditions: Player playing game.

Post Conditions:

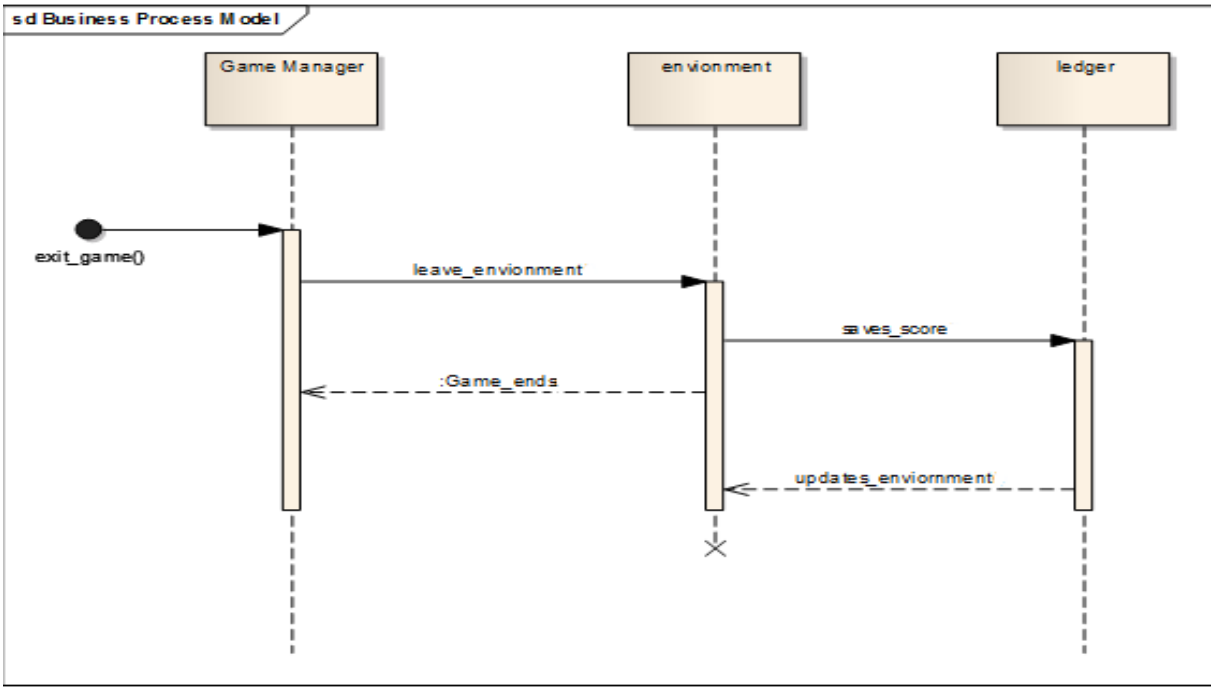
1. An instance for playerTrack was generated.
2. It was associated with player.
3. X and y margin were modified

5. SEQUENCE DIAGRAM

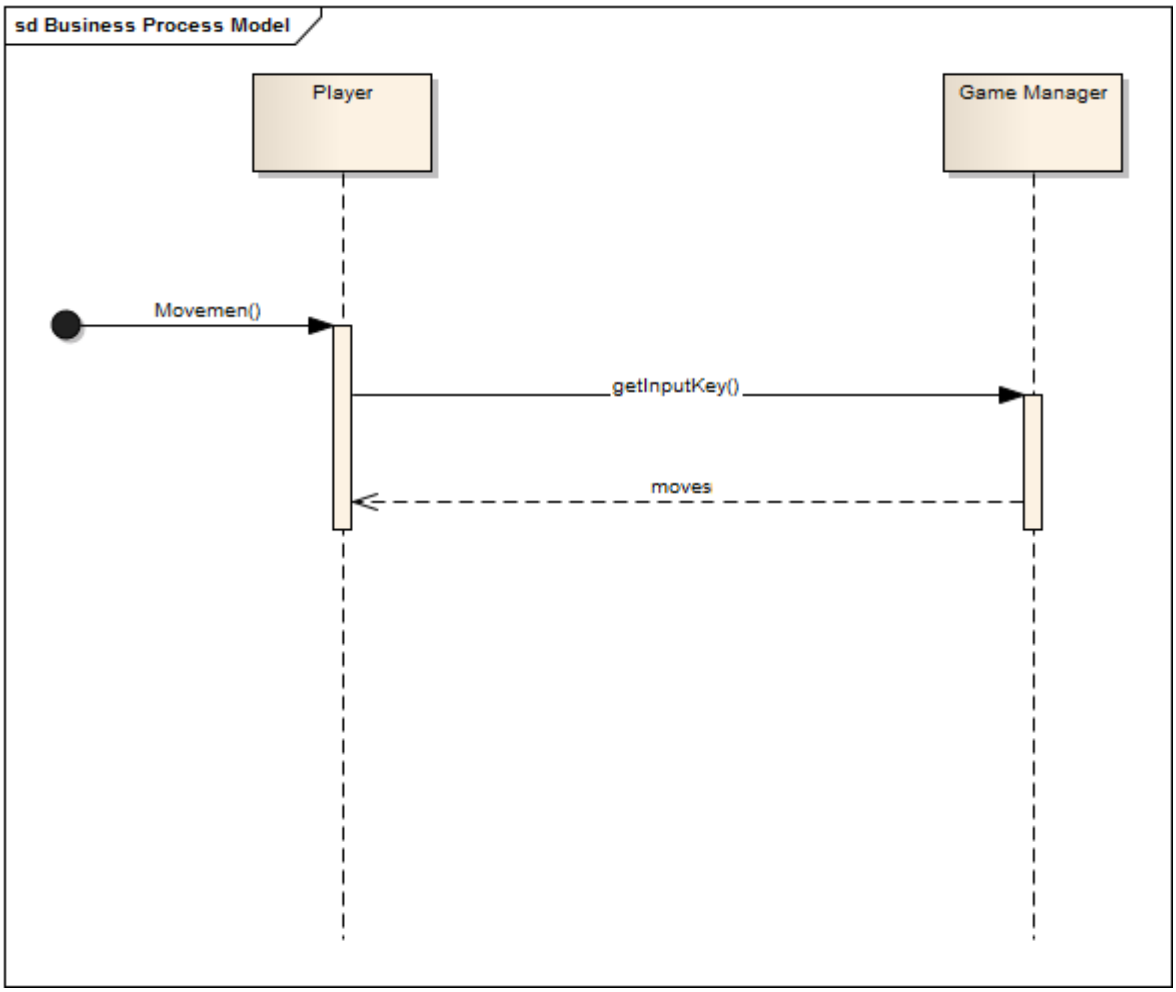
- Start Game



- Exit Game



- Move Player


sd Business Process Model
Player
Game Manager
Movemen()
getInputKey()
moves

ITERATION 2

1. EXPANDED USECASES

USE CASE NAME	**Kill bot**
Scope	The BreakOut
Level	User goal
Primary actors	Player
Stakeholders & Interest	Player wants to kill the bot
Pre-Conditions	User is playing the game
Post Conditions	The bot will die and the score will be updated.
Main Success Scenario **Actions** 1. Player shots the bot.	**System Response** 2. System kills the bot. 3. System updates and saves the score.
Extensions	Nil
Special Requirements	Game executes properly in the environment
Technology and Data Variation List	Nil
Frequency of occurrence	Nil

USE CASE NAME	**How to Play**
Scope	The BreakOut
Level	User goal
Primary actors	Player
Stakeholders & Interest	Player needs help to understand the game
Pre-Conditions	User is in the main menu
Post Conditions	System will guide the user
Main Success Scenario **Actions** 1. Player clicks on the how to play option.	**System Response** 2. System guides the user.
Extensions	Nil
Special Requirements	Game executes properly in the environment
Technology and Data Variation List	Nil
Frequency of occurrence	Nil

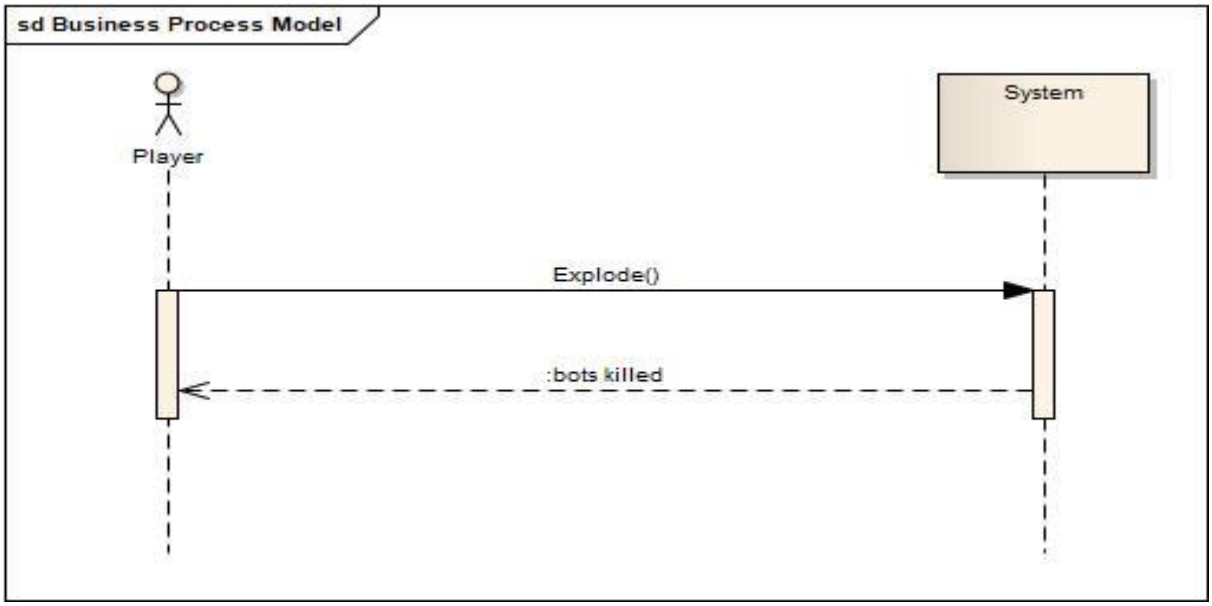
USE CASE NAME	Pick Puzzle
Scope	The BreakOut
Level	User goal
Primary actors	Player
Stakeholders & Interest	Player selects the puzzle
Pre-Conditions	User is playing the game
Post Conditions	The player is moved to the next level.
Main Success Scenario **Actions** 1. Player picks up the puzzle.	**System Response** 2. System upgrades the player level.
Extensions	Nil
Special Requirements	Game executes properly in the environment
Technology and Data Variation List	Nil
Frequency of occurrence	Nil

USE CASE NAME	Pick Health
Scope	The BreakOut
Level	User goal
Primary actors	Player
Stakeholders & Interest	Player selects the health
Pre-Conditions	User is playing the game
Post Conditions	The strength of player is increased.
Main Success Scenario **Actions** 1. Player picks up the health.	**System Response** 2. System increases the player's strength.
Extensions	Nil
Special Requirements	Game executes properly in the environment
Technology and Data Variation List	Nil
Frequency of occurrence	Nil

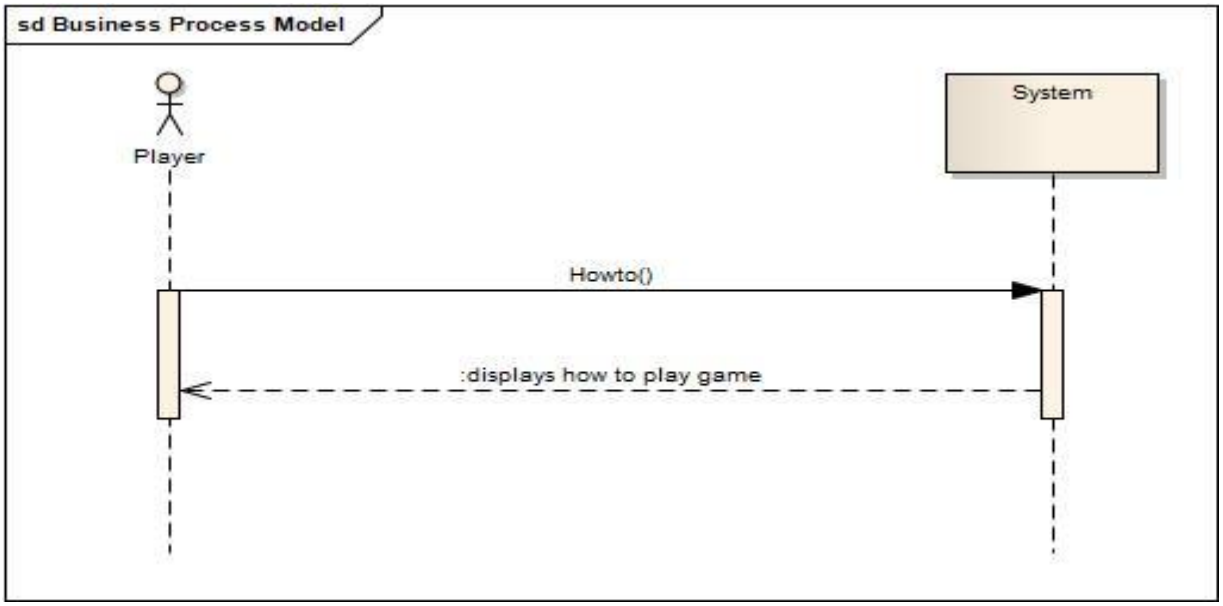
USE CASE NAME	Go to next level
Scope	The BreakOut
Level	User goal
Primary actors	Player
Stakeholders & Interest	Player selects the next level
Pre-Conditions	User is playing the game
Post Conditions	The player is moved to the next level.
Main Success Scenario **Actions** 1. Player selects the level option.	**System Response** 2. System upgrades the player level.
Extensions	Nil
Special Requirements	Game executes properly in the environment
Technology and Data Variation List	Nil
Frequency of occurrence	Nil

3. SYSTEM SEQUENCE DIAGRAM

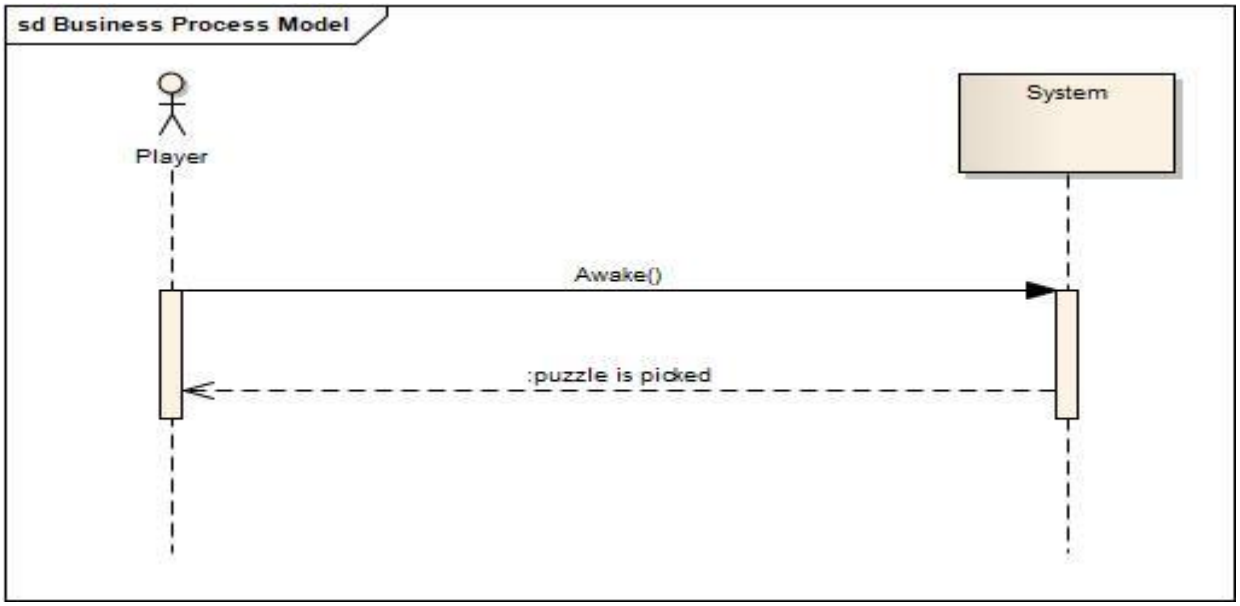
- Kill Bot



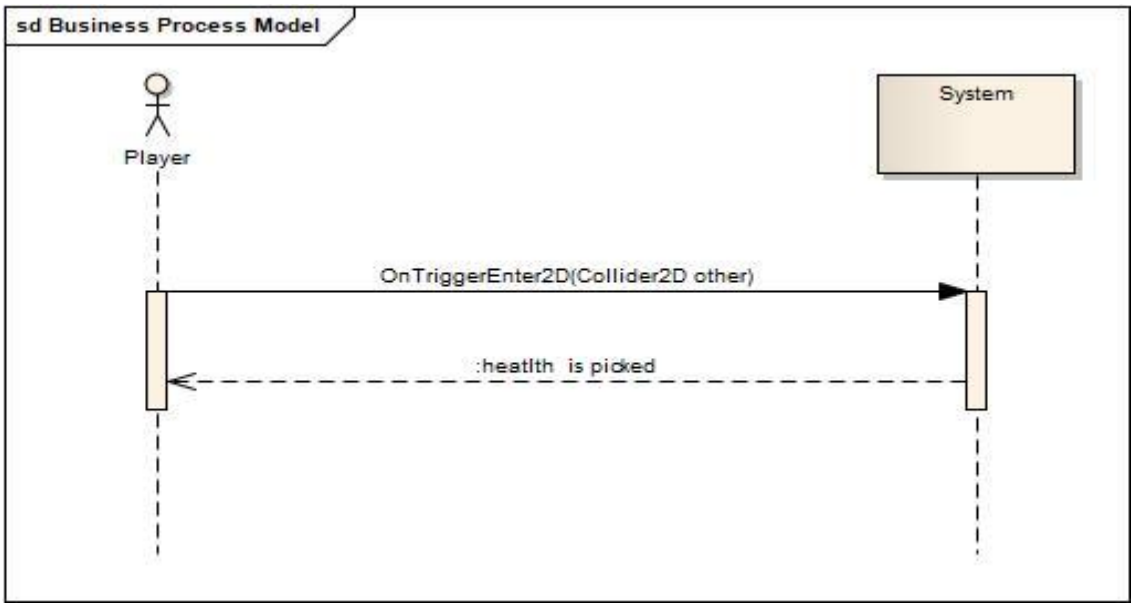
- How to Play



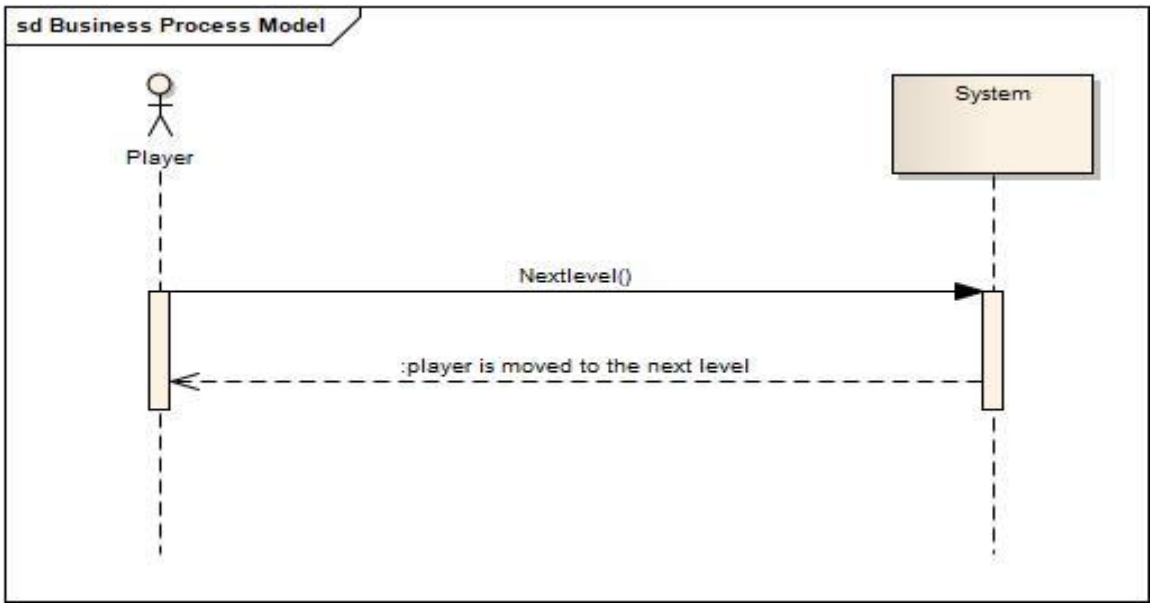
- Pick Puzzle



- Pick Health

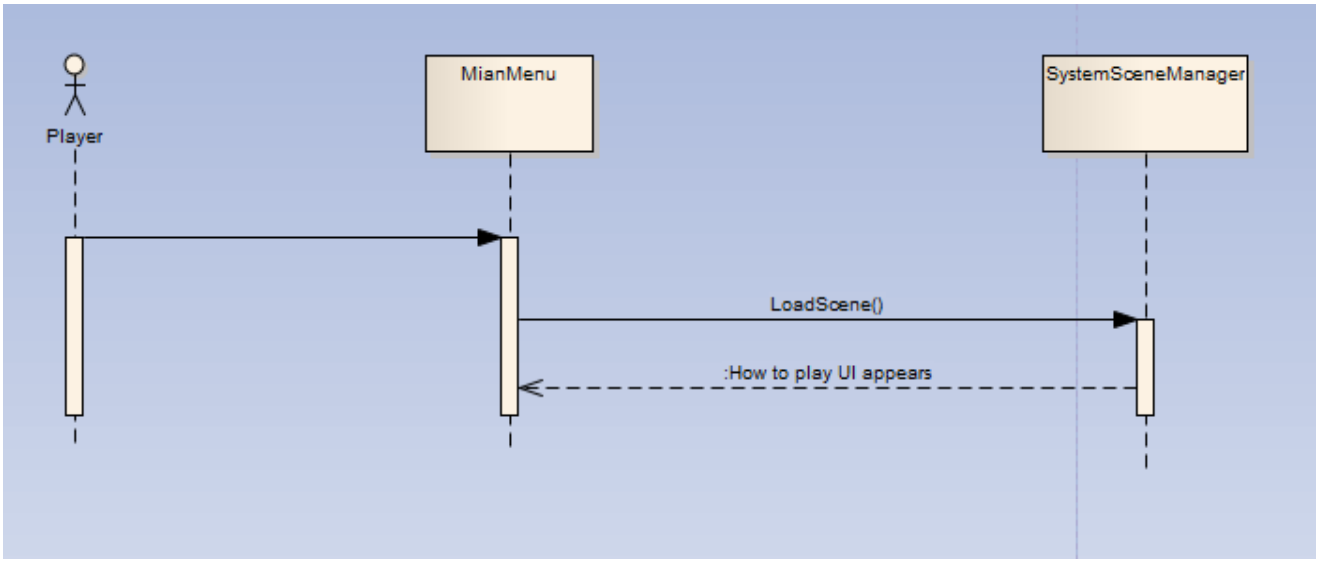


- Go to next level

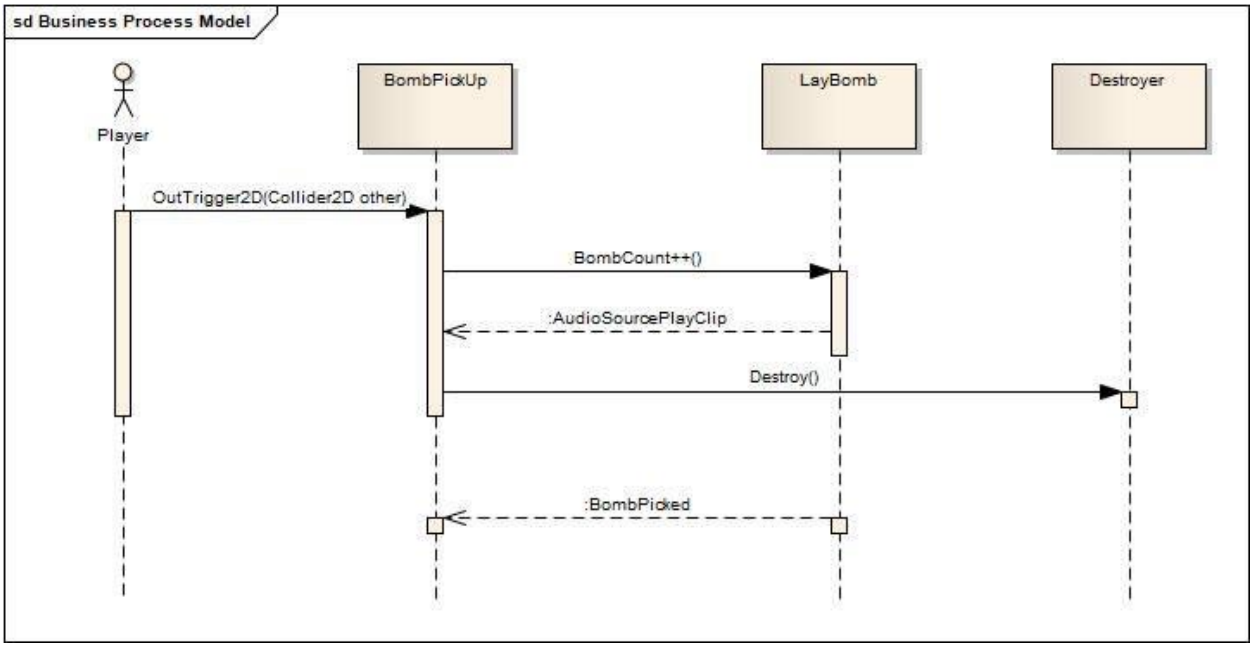


4. SEQUENCE DIAGRAM

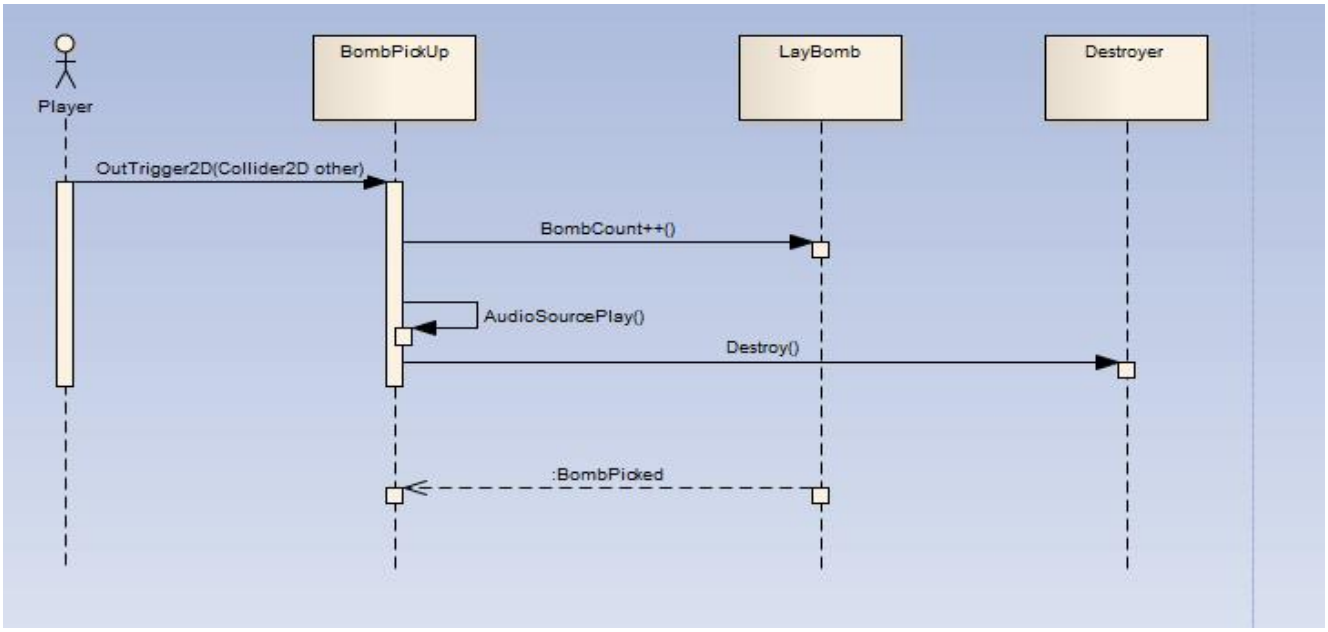
1. How to Play



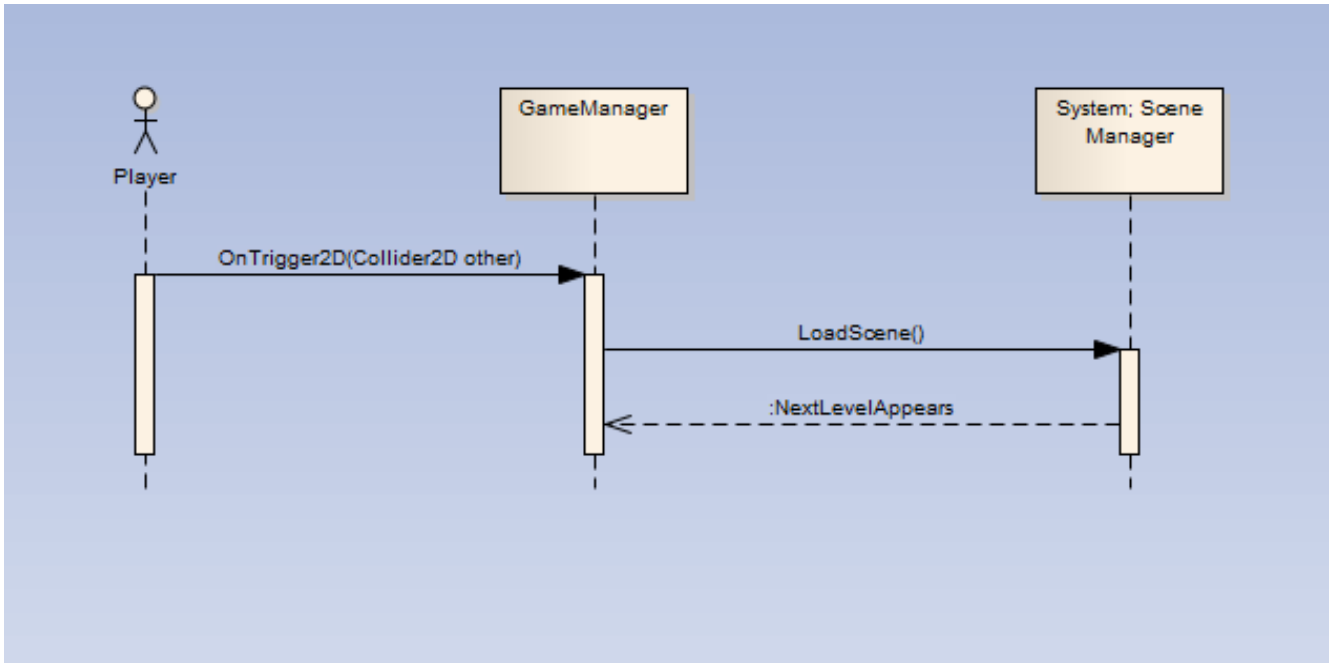
2. Pick Bomb



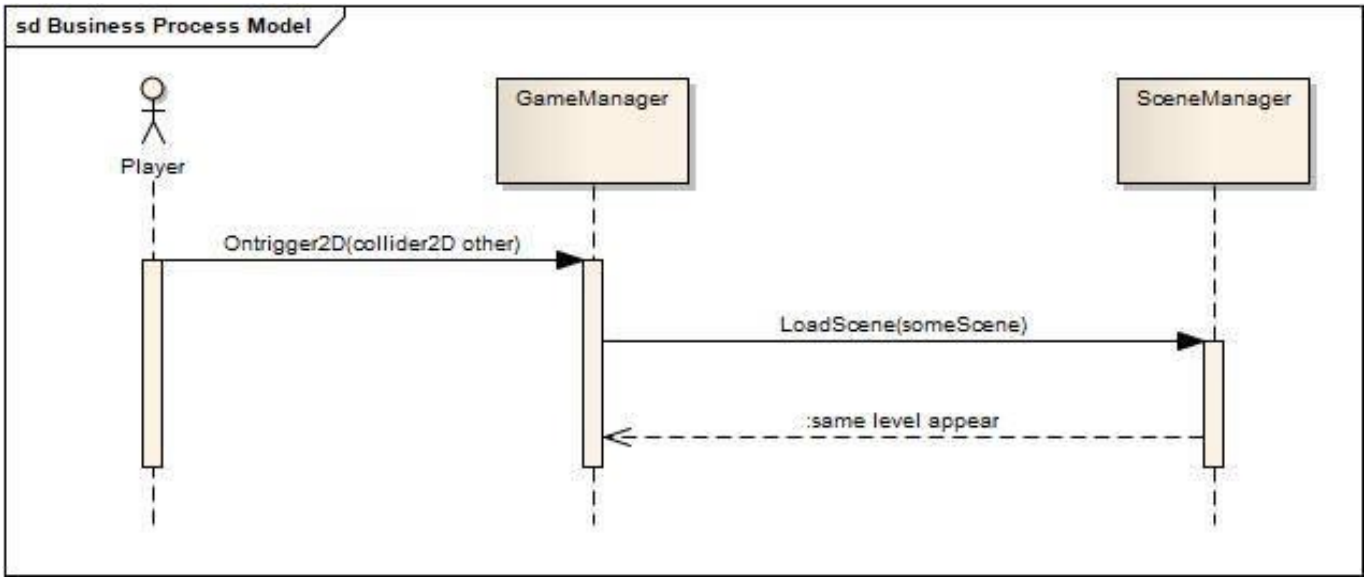
3. Pick Health



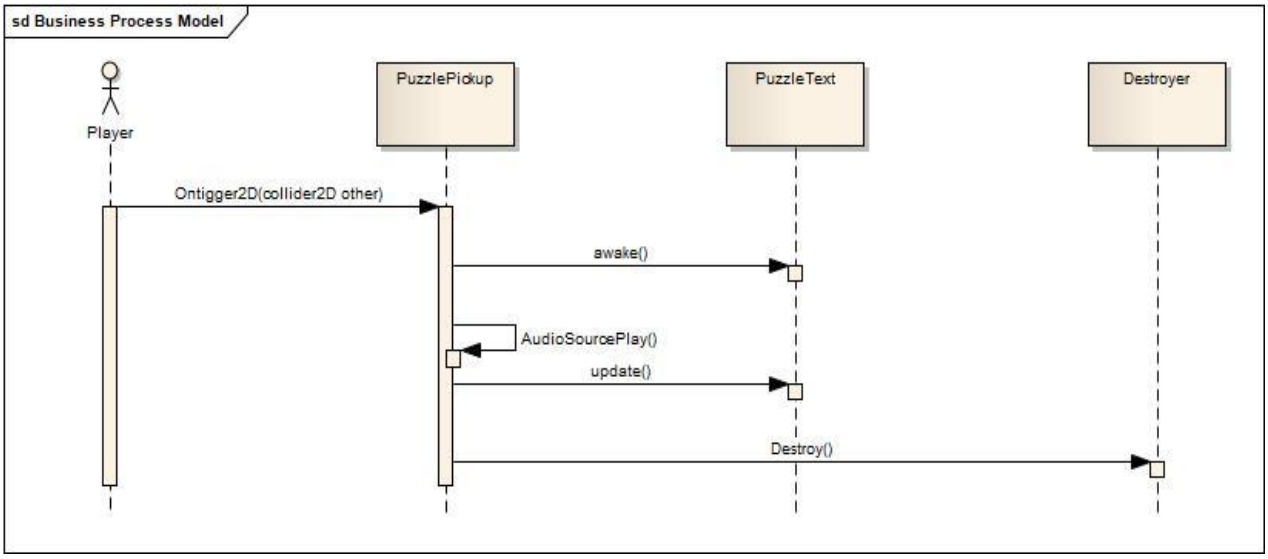
4. Go to Next Level



5. Restart Level



6. Pick Puzzle



5. OPERATION CONTRACTS

Operation Contract # 4

Name: Death();

Responsibility: It shoots the fire, kill bots, and updates the score.

Type: System

Cross Reference: UseCase4 Kill bot

Exceptions: Nil

Pre-Conditions: Player is in the playing environment.

Post Conditions:

4. An instance of bomb was generated.
5. It was associated with bots.

Operation Contract # 5

Name: HowTo();

Responsibility: It displays the how to play page.

Type: System

Cross Reference: UseCase5 How to Play

Exceptions: Nil

Pre-Conditions: Player has entered the game process

Post Conditions:

1. An instance of loadScene was generated.

Operation Contract # 6

Name: OnTrigger2d();

Responsibility: Allows the player to eat a certain puzzle.

Type: System

Cross Reference: UseCase6 Pick Puzzle

Exceptions: Nil

Pre-Conditions: Player is in the playing environment.

Post Conditions:

1. An instance of puzzle was generated.
2. It was associated with environment and player.

Operation Contract # 7

Name: OnTriggerEnter2D();

Responsibility: Allows the player to select a certain health.

Type: System

Cross Reference: UseCase7 Pick Health

Exceptions: Nil

Pre-Conditions: Player is in the playing environment.

Post Conditions:

1. An instance of health was generated.
2. It was associated with environment and player.

Operation Contract # 8

Name: NextLevel();

Responsibility: Allows the player to go to the next level.

Type: System

Cross Reference: UseCase8 Go to next Level

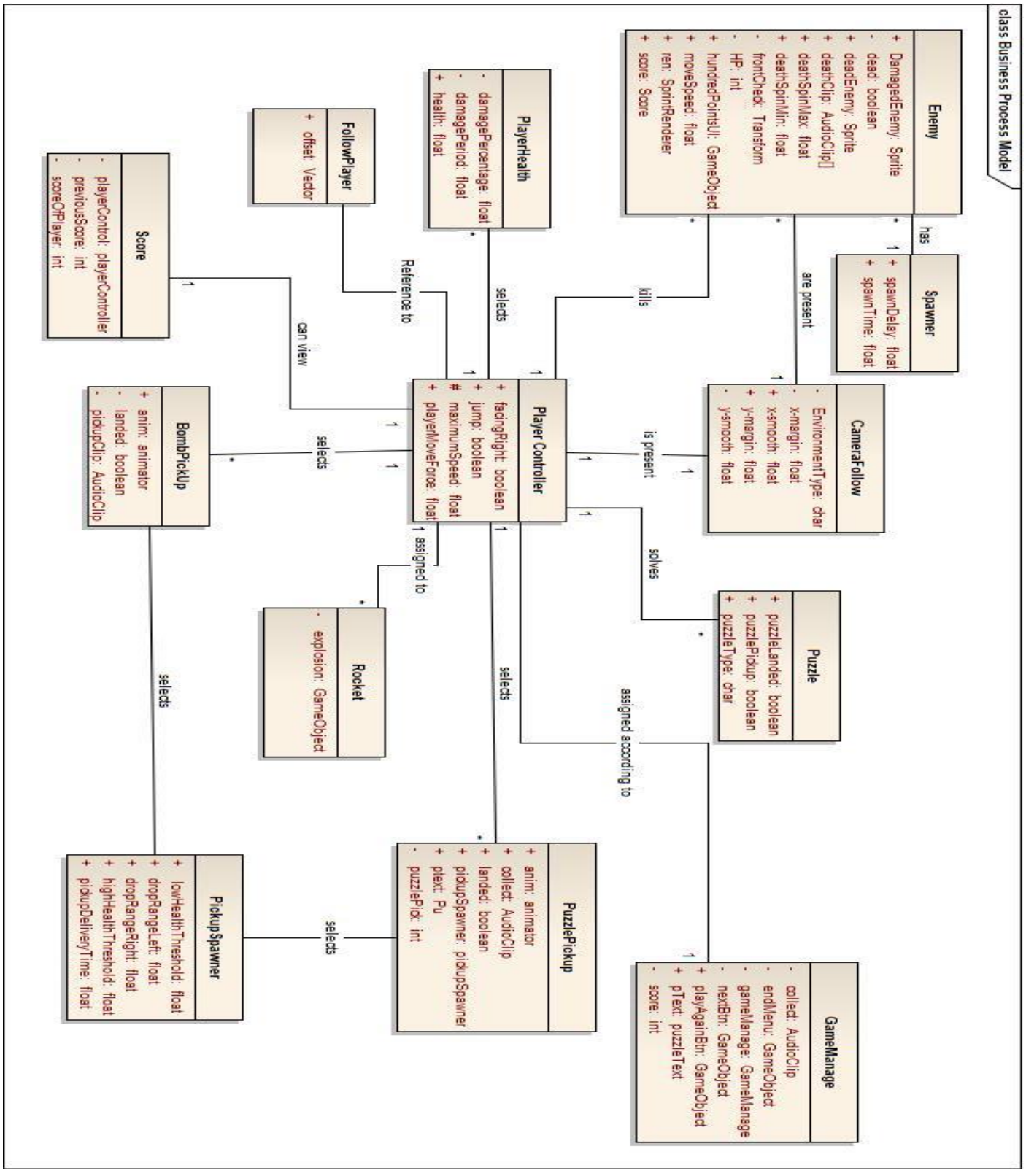
Exceptions: Nil

Pre-Conditions: Player is in the playing environment.

Post Conditions:

1. An instance of level was generated.
2. It was associated with environment and player.

6. DOMAIN MODEL



7. CLASS DIAGRAM