
Software Requirements Specification and Analysis

for



Prepared by:

Yasin Sazid: BSSE1006

Billal Hossain Nirab: BSSE1020

Institute of Information Technology, University of Dhaka.

Date created: 21/10/2020

Table of Contents

Table of Contents	ii
Revision History	iv
1. Introduction.....	5
1.1 Purpose	5
1.2 Document Conventions	5
1.3 Intended Audience and Reading Suggestions	5
1.4 Product Scope	5
2. Overall Description	6
2.1 Identifying stakeholders	6
2.2 Identifying and resolving multiple viewpoints	6
2.3 Product function summary	7
2.4 Operating Environment	7
2.5 User Documentation	7
2.6 Assumptions and Dependencies	7
3. External Interface Requirements	7
3.1 User Interfaces	7
3.2 Software Interfaces	7
3.3 Communications Interfaces	7
4. System Requirements	8
4.1 Normal Requirements	8
4.2 Expected Requirements	8
4.3 “WOW” Requirements	9
5. System Features.....	10
5.1 Open world	10
5.2 Shooting.....	10
5.3 Multiplayer	10
5.4 Character Selection	10
5.5 AI Bot System for single player	10
5.6 Graphics Scalability Settings	10
5.7 Create User Profile	11
5.8 Group chatting to communicate	11
5.9 Interactive object	11
5.10 Smart health system.....	11
5.11 Automatic/Manual Ammo reload system.....	11
5.12 Weapon drop	11
5.13 Run time weapon snatch system.....	12
5.14 Advanced locomotion.....	12
5.15 Photorealistic environment.....	12
5.16 Advanced optimization for higher frame rate	12
5.17 Custom build UV map.....	12
5.18 Localization	12
5.19 Save Game State.....	13
6. Use case diagram.....	14
6.1 Use case diagram level: 0	14
6.2 Use case diagram level: 1	15
6.3 Use case diagram level: 1.2	17
6.4 Use case diagram level: 1.4	18
6.5 Use case diagram level: 1.6	19
6.6 Use case diagram level: 1.6.2	20
6.7 Use case diagram level: 1.9	22
6.8 Use case diagram level: 1.10	23
6.9 Use case diagram level: 1.11	24

6.10	Use case diagram level: 1.12	26
7.	Activity Diagrams:	27
7.1	Activity diagram ID: 1	27
7.2	Activity diagram ID: 2	28
7.3	Activity diagram ID: 3	29
7.4	Activity diagram ID: 4	30
7.5	Activity diagram ID: 5	31
7.6	Activity diagram ID: 6	32
7.7	Activity diagram ID: 7	33
7.8	Activity diagram ID: 8	34
7.9	Activity diagram ID: 9	35
7.10	Activity diagram ID: 10	36
7.11	Activity diagram ID: 11	37
7.12	Activity diagram ID: 12	38
7.13	Activity diagram ID: 13	39
7.14	Activity diagram ID: 14	40
7.15	Activity diagram ID: 15	41
7.16	Activity diagram ID: 16	42
7.17	Activity diagram ID: 17	43
7.18	Activity diagram ID: 18	44
7.19	Activity diagram ID: 19	45
7.20	Activity diagram ID: 20	46
7.21	Activity diagram ID: 21	47
7.22	Activity diagram ID: 22	48
7.23	Activity diagram ID: 23	49
7.24	Activity diagram ID: 24	50
7.25	Activity diagram ID: 25	51
7.26	Activity diagram ID: 26	52
7.27	Activity diagram ID: 27	53
8.	Swimlane Diagrams:	54
8.1	Swimlane Diagram ID: 1	54
8.2	Swimlane Diagram ID: 2	55
8.3	Swimlane Diagram ID: 3	56
8.4	Swimlane Diagram ID: 4	57
8.5	Swimlane Diagram ID: 5	58
8.6	Swimlane Diagram ID: 6	59
8.7	Swimlane Diagram ID: 7	60
8.8	Swimlane Diagram ID: 8	61
8.9	Swimlane Diagram ID: 9	62
8.10	Swimlane Diagram ID: 10	63
8.11	Swimlane Diagram ID: 11	64
8.12	Swimlane Diagram ID: 12	65
8.13	Swimlane Diagram ID: 13	66
8.14	Swimlane Diagram ID: 14	67
8.15	Swimlane Diagram ID: 15	68
8.16	Swimlane Diagram ID: 16	69
8.17	Swimlane Diagram ID: 17	70
8.18	Swimlane Diagram ID: 18	71
8.19	Swimlane Diagram ID: 19	72
8.20	Swimlane Diagram ID: 20	73
8.21	Swimlane Diagram ID: 21	74
8.22	Swimlane Diagram ID: 22	75
8.23	Swimlane Diagram ID: 23	76
8.24	Swimlane Diagram ID: 24	77
8.25	Swimlane Diagram ID: 25	78
8.26	Swimlane Diagram ID: 26	79
8.27	Swimlane Diagram ID: 27	80

9. Database modeling	81
9.1 Data Object Identification	81
9.2 Data Objects	83
9.3 Analysis:	83
9.4 Final Data Objects	83
9.5 Schema	84
10. Class Based Modeling	85
10.1 General classification	85
10.2 Selection criteria	86
10.3 List of verbs	87
10.4 Potential Classes	88
10.5 Analysis:	90
10.6 Class Cards	90
10.7 CRC Diagram:	97
11. Behavioral Modeling	98
11.1 Event Identification Table:	98
11.2 State transition diagrams	105
11.3 Sequence diagram	114
12. Appendix	115
12.1 Glossary	115

Revision History

Name	Date	Reason For Changes	Version
First creation	21/10/20	This is the first creation of the document	1.0

1. Introduction

1.1 Purpose

Moitrigor is an open world third person PC shooting game that will have both single player and multiplayer modes. Gaming scenario in Bangladesh has evolved from arcade gaming to computer and console gaming. Now is the time of online-based multiplayer games. Competitive gaming is gaining popularity among the youth, opening a huge field for game development. This industry is still trying to find its feet here in Bangladesh. But globally, gaming industry has been a well settled one for a long time. So there is a lot of prospect and opportunity in this industry in Bangladesh. We want to be a part of it. This project of ours is our first attempt to do so. This project is going to be a primary learning curve for us and we want to ultimately release our project as a product in the local market.

1.2 Document Conventions

The document is typed font Arial, size 11. The structure of the document has been maintained close to that provided by IEEE for SRS template.

1.3 Intended Audience and Reading Suggestions

This document is intended for software developers, project managers, marketing staff, users, testers, and documentation writers. The rest of the document contains SRSA (software requirement specification and analysis) starting from product features, all the way to diagrams and plan of implementation. This document is best read serially from top to bottom.

1.4 Product Scope

Developing a PC shooting game that can be played in both single player and multiplayer modes. We will have AI bot system for single player mode. It will be an open world game where players can move freely and interact with the photorealistic environment.

2. Overall Description

2.1 Identifying stakeholders

Stakeholder refers to any person or group who will be affected directly or indirectly by the system. Stakeholders, in this case, they are:

1. Gamers
2. Developers
3. Guardians of Gamers

All stakeholders are expected to have a very basic technical expertise necessary to operate a computer (or similar device) with an internet connection. Users of any level of education with or above basic literacy are welcome to use the product.

2.2 Identifying and resolving multiple viewpoints

2.2.1 Gamers' viewpoints:

Gamers want the excitement of playing in local locations. They also wanted the thrill of real life shooting, i.e., weapon recoil etc.

2.2.2 Developers' viewpoints

For this project, the scope of the project is restricted. So the developers can implement only a limited portion of what the gamers' want in a shooting game. Again, not every wish of a gamer is feasible to implement in a game.

2.2.3 Guardians' viewpoints

The game should maintain community standards at all costs.

2.2.4 Contradictory viewpoints

The following demands from gamers contradict that from developers:

1. Detailed weapon and experience system
2. Different experiences of different weapons

Developers' response to this:

These features are not feasible to implement within the current scope of this project.

2.2.5 Resolution

These features will be ignored within the current scope of the project. We may include them in the future.

2.3 Product function summary

A very high-level function summary is provided below. The functions are discussed in greater detail in section 5.

The product will provide:

- Third person gaming experience
- Shooting based game
- Interaction with the photorealistic environment
- Movement in free world
- Graphics settings

2.4 Operating Environment

The product will operate as a native app on windows.

2.5 User Documentation

A hardcopy of user manual is intended to be delivered with the product.

2.6 Assumptions and Dependencies

We assume that the product will be used in a gaming computer.

3. External Interface Requirements

3.1 User Interfaces

User interface consisting of graphical objects are required for the product.

3.2 Software Interfaces

Software interfaces of Unreal Engine by Epic Games.

3.3 Communications Interfaces

For server-client play, interfaces of Unreal engine by Epic Games.

4. System Requirements

4.1 Normal Requirements

The following is the set of requirements that our stakeholders have explicitly demanded from our product:

4.1.1 Movement

Player needs to be able to go in all directions, crouch, walk and jump.

4.1.2 Shooting

Player will be able to shoot at the aimed target.

4.1.3 Weapon

Player can pick, swap or drop weapons.

4.1.4 Health

Player health will be updated when he/she gets shot.

4.1.5 User Profile

Player can create profile with his name and avatar.

4.1.6 Multiplayer

Multiple players can play with or against each other.

4.1.7 Third person view

The gamer will be able to view the open world map from the perspective of a third person.

4.2 Expected Requirements

The following are the requirements that were not demanded explicitly, but are nevertheless necessary for a complete system.

4.2.1 Performance Requirements

Must be able to run smoothly on host device.

4.2.2 Security Requirements

Game host device should be able to control who can and who can not play the game in multiplayer.

4.2.3 Software Quality Attributes

The following quality attributes are desired:

- Adaptability and flexibility
- Availability
- Correctness
- Interoperability
- Maintainability
- Portability
- Reliability
- Reusability
- Robustness
- Testability
- Usability

4.2.4 Menu/Lobby

There needs to be a menu from where the users can take actions.

4.3 “WOW” Requirements

4.3.1 Open World

Player can move freely inside the game world.

4.3.2 Graphics Settings

Player can change the graphics settings for the game.

4.3.3 AI Bot

Player can play against artificially intelligent robot players.

4.3.4 Chatting

Players can chat with each other during a game session.

4.3.5 Photorealistic Environment

The game’s environment will be similar to real life.

4.3.6 Interactive Environment

Players will be able to interact with certain objects in the game environment.

5. System Features

In order to satisfy the stated requirements, the following features are planned for implementation.

5.1 Open world

Players can roam freely through virtual world. There is a lot of macro details to beautify the map.

5.2 Shooting

Player can shoot. Player can see how many opponents he/she has killed during a match.

5.3 Multiplayer

Player can host game by creating listen server and client can join game through IP. Server has ability to kick player from match.

5.4 Character Selection

Without selecting character, player can not start the game. There is 8 character options to choose from.

5.5 AI Bot System for single player

When player wants to play at single player mode, he/she can play with interactive bots.

5.6 Graphics Scalability Settings

For lower specifications, player can change graphics settings to enjoy smooth gameplay. There are 4 graphics options available to user - Low, Medium, High and Ultra. User has Control to use manual graphics settings by changing shadow quality, anti aliasing, FPS, post processing and so on.

5.7 Create User Profile

If player has no profile when he/she starts the game, system will automatically recommend to create user profile with name and avatar and store it into database provided by engine.

5.8 Group chatting to communicate

User can send text message to each other to communicate. User can send text message when he is in Lobby and also during gameplay. The power of communication enhances gameplay experience.

5.9 Interactive object

For better gameplay experience and feel the art of realism, player can interact with game object. For example when player touches butterfly, it starts flying and player is able to hear the sound of its wings. There are interactive door system also.

5.10 Smart health system

There are smart health system. For example, damage of sniper is more than rifle. System can automatically detect it and apply damage to player. There is also a first aid kit to recover health.

5.11 Automatic/Manual Ammo reload system

When ammos are finished, automatic reload system reload ammos and user can also reload ammos by pressing the reload button.

5.12 Weapon drop

When ammos are totally finished, player can drop his/her weapon and also pick another one from the shop. Shop is available on the map.

5.13 Run time weapon snatch system

User can pick multiple weapons and snatch weapons during gameplay.

5.14 Advanced locomotion

User can experience advanced locomotion. Player can walk, run and jump. Crouching and aim offset make user gameplay experience to the next level.

5.15 Photorealistic environment

Assets from Megascan library which is scanned from real world make scene more photorealistic. Custom generated normal map and photo scanned from real world enhances photorealism.

5.16 Advanced optimization for higher frame rate

Multi level of detail (LOD) optimizes fps by increment and decrement vertices count of mesh depending on camera distance. Use of low resolution but seamless texture is used to optimize render time.

5.17 Custom build UV map

-For custom designed mesh: Scan photo from real world and generate 3D mesh using Meshroom.

- For custom designed UV map: Scan texture From real world and generate height map using Photoshop.

5.18 Localization

Game Environment is based on Bangladeshi local Culture. We will add Bangladeshi local flavor and we will use Bangla graffiti on the walls and name the places in Bengali.

5.19 Save Game State

Player can pause and resume the game in the single player mode. Game state will be saved.

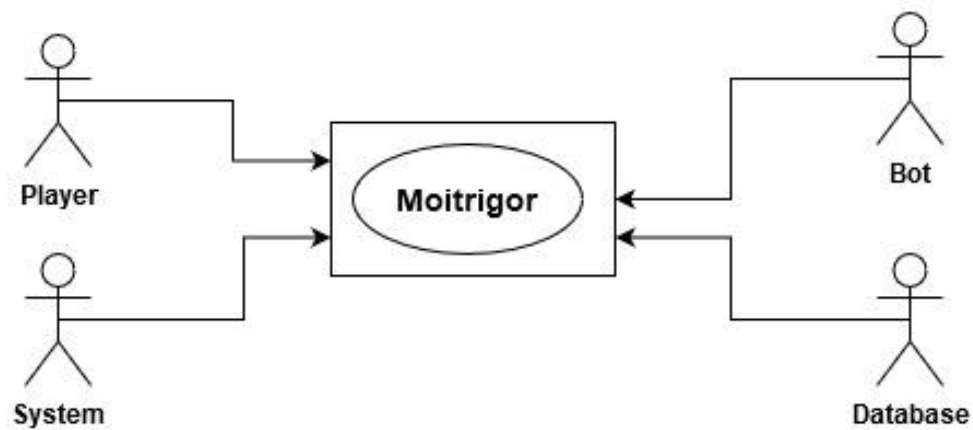
6. Use case diagram

6.1 Use case diagram level: 0

ID: 1

Name: Moitrigor

Level 0



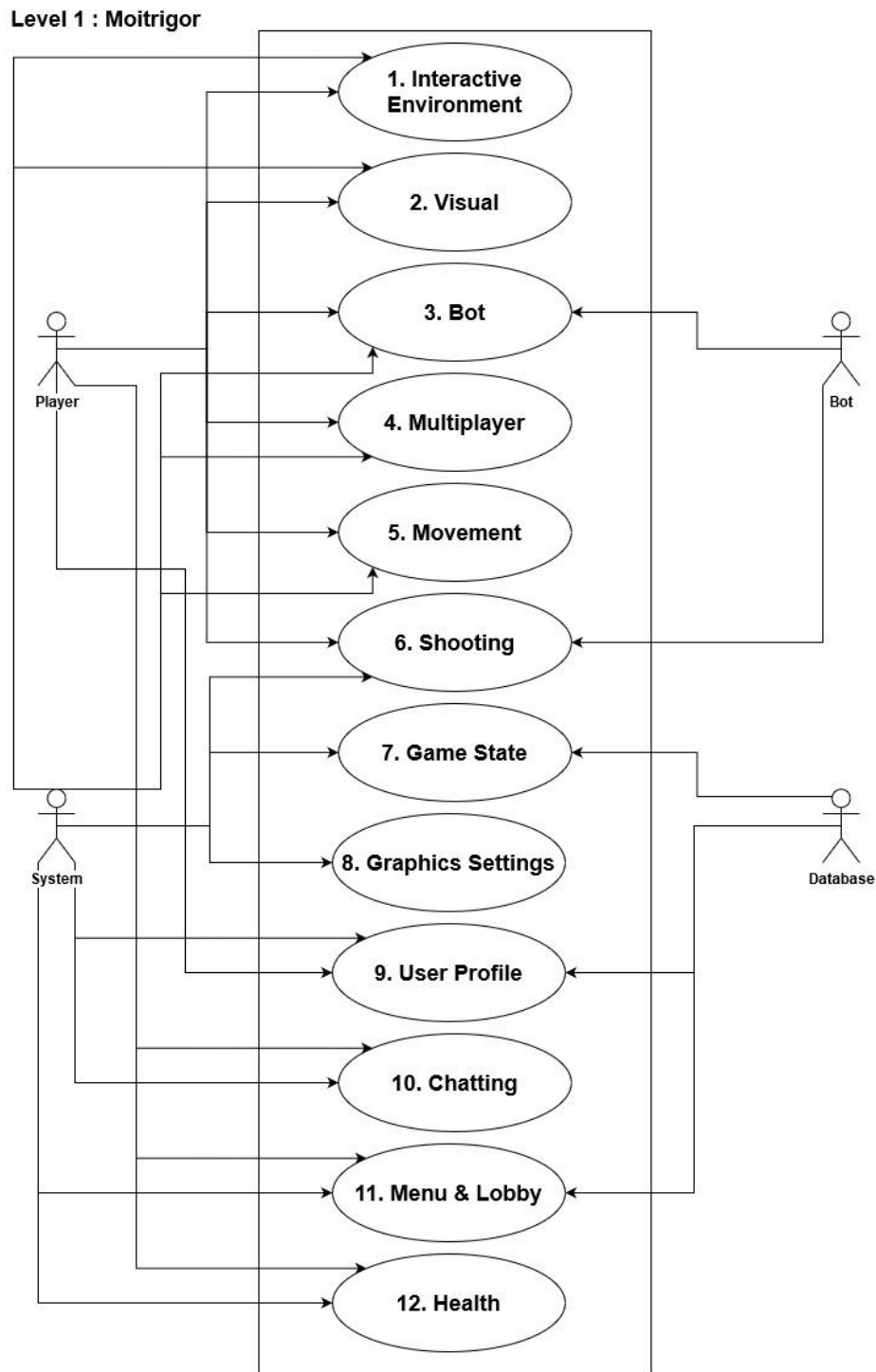
Description: Here, the usecase “Moitrigor” represents the entire game system. Interacting with it are the four actors:

- | | | |
|-------------|---|-----------------|
| 1. Player | - | primary actor |
| 2. System | - | primary actor |
| 3. Bot | - | secondary actor |
| 4. Database | - | secondary actor |

6.2 Use case diagram level: 1

ID: 2

Name: Sub Systems of Moitrigor



Description: The entire system is divided into 12 major parts.

Level 1.1: Interactive Environment

This module will deal with the interaction between the player and certain objects in the game world. When the player tries to interact with this kind of objects, the object will produce a response that will be shown in the game world.

Level 1.3: Bot

This module will deal with the AI bot response. Certain player movement and action will trigger nearby AI bots and their response will be managed in this module.

Level 1.5: Movement

This module will deal with player movements like running, walking, crouching, jumping and giving directions.

Level 1.7: Game State

This module will be used to save an ongoing game in the database.

Level 1.8: Graphics Settings

This module will deal with the changes in graphics settings made by the player.

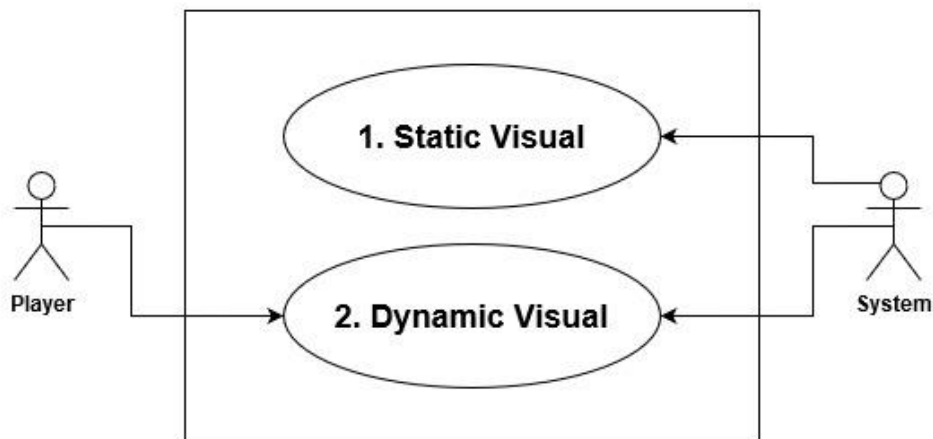
Other modules will be discussed thoroughly in the later sections.

6.3 Use case diagram level: 1.2

ID: 3

Name: Visual

Level 1.2: Visual



Description: This module deals with all kinds of visual outputs of the game environment. It contains two sub-modules –

Level 1.2.1: Static Visual

This module deals with the visuals of game environment that do not change with time. These are shown to the players based on their location and movement.

Level 1.2.2: Dynamic Visual

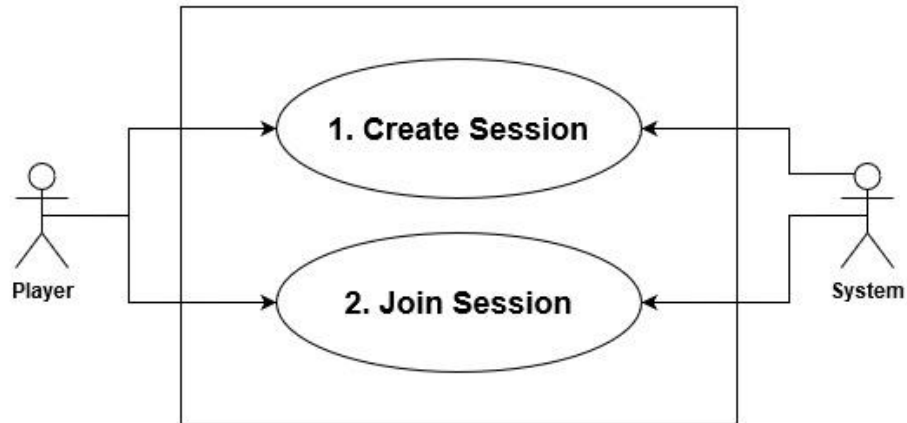
This module deals with the visuals of game environment that change with time. This contains 3D animation of movements of other players, shooting and weapon animation etc. These are shown to the players based on different actions and inputs from them.

6.4 Use case diagram level: 1.4

ID: 4

Name: Multiplayer

Level 1.4: Multiplayer



Description: This module deals with the multiplayer sessions of the game where players can play with their friends from the real world. It contains two sub-modules –

Level 1.4.1: Create Session

This module lets the player create a listen server that allows other players to join as clients and play the game together.

Level 1.4.2: Join Session

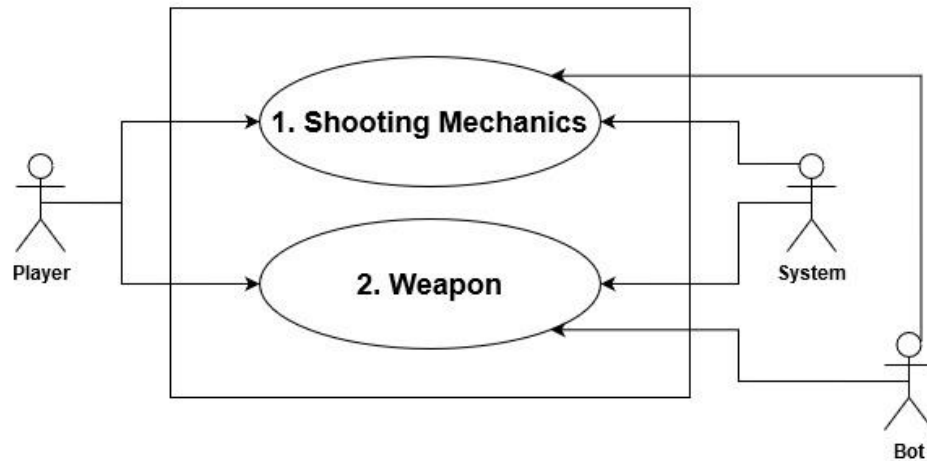
This module lets players join any game session using that session's server ip as clients so that they can play together.

6.5 Use case diagram level: 1.6

ID: 5

Name: Shooting

Level 1.6: Shooting



Description: This module deals with weapon and shooting mechanics. It has two sub-modules –

Level 1.4.1: Shooting Mechanics

This module deals with shooting input, aiming and the corresponding output.

Level 1.4.2: Weapon

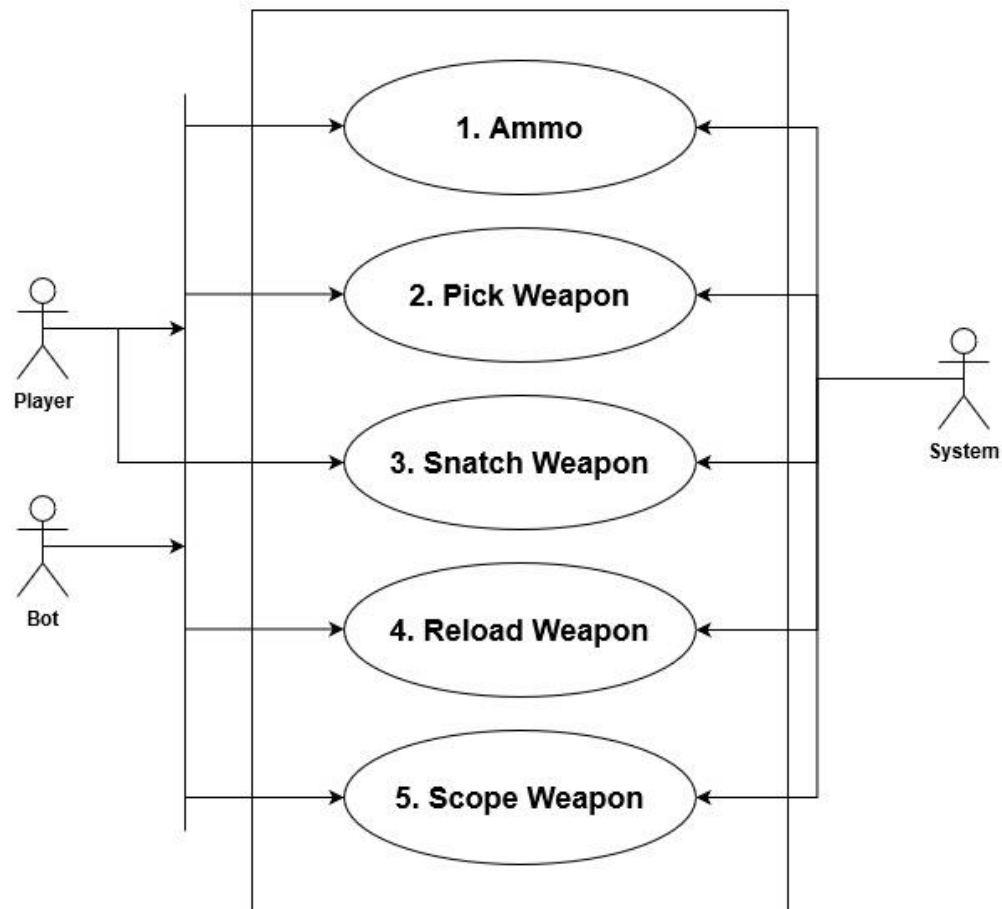
This module will be discussed in detail in the next section.

6.6 Use case diagram level: 1.6.2

ID: 6

Name: Weapon

Level 1.6.2: Weapon



Description: This module deals with all kinds of weapon mechanics. It contains five sub-modules –

Level 1.6.2.1: Ammo

This module deals with picking up extra ammo for weapons.

Level 1.6.2.2: Pick Weapon

This module deals with picking up weapons.

Level 1.6.2.3: Snatch Weapon

This module deals with snatching weapons from in-game bots.

Level 1.6.2.4: Reload Weapon

This module deals with reloading weapons. There are two types of reload. Weapons get automatically reloaded when ammo inside the weapon is finished. Players can also manually reload their weapons at any moment.

Level 1.6.2.5: Scope Weapon

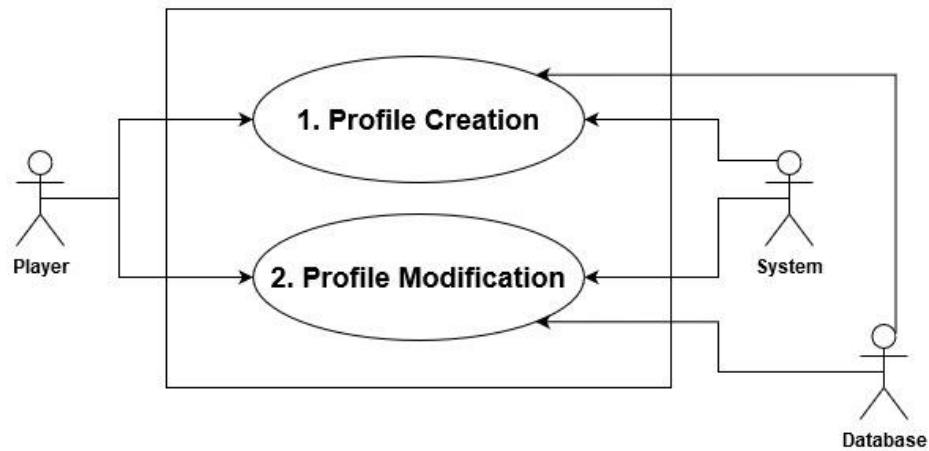
This module deals with scoping or unscoping the current weapon. Scoping allows players to zoom in on the opponents. It makes aiming easier.

6.7 Use case diagram level: 1.9

ID: 7

Name: User Profile

Level 1.9: User Profile



Description: This module deals with player profile management.

It has two sub-modules –

Level 1.9.1: Profile Creation

This module deals with creating player profile with name and avatar information. It also stores the player profile in the database.

Level 1.9.2: Profile Modification

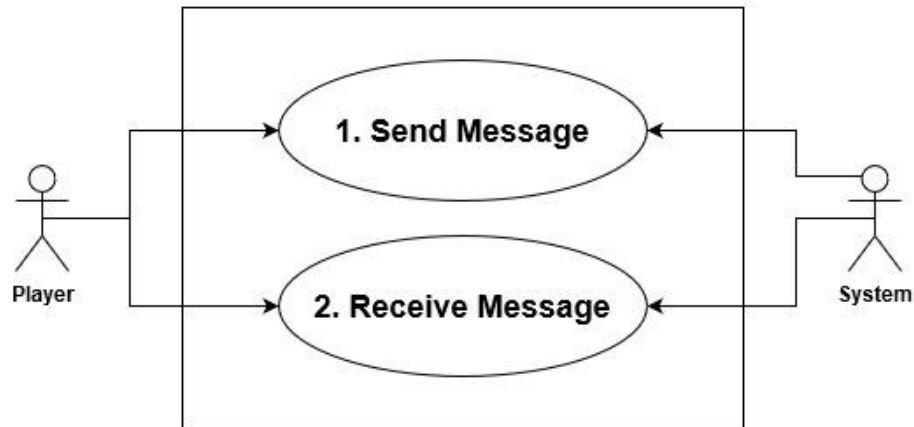
This module deals with modifying information of an already created player profile and stores the modified profile information in the database.

6.8 Use case diagram level: 1.10

ID: 8

Name: Chatting

Level 1.10: Chatting



Description: This module deals with player messages in the lobby.

It has two sub-modules –

Level 1.10.1: Send Message

This module allows players to send message to the lobby containing all players in the session.

Level 1.10.2: Receive Message

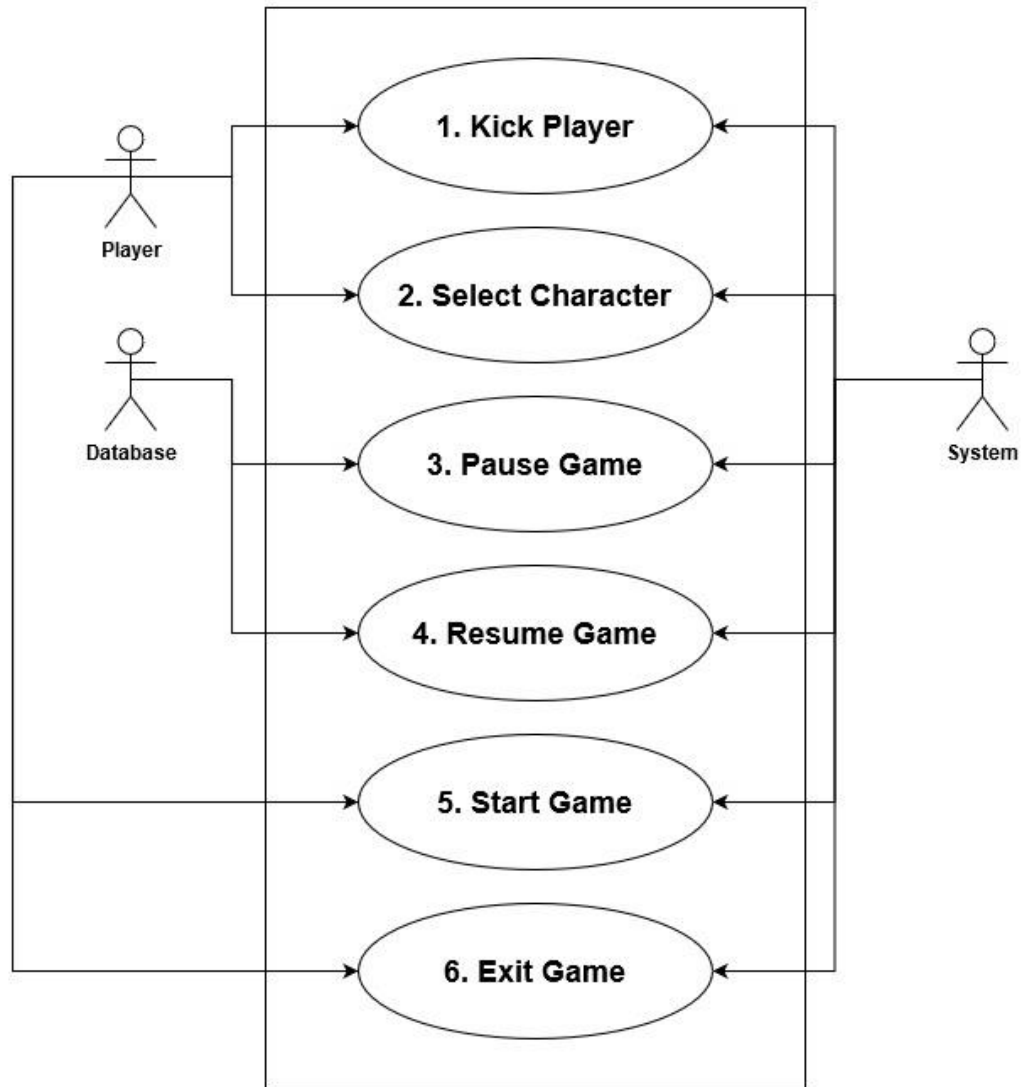
This module shows the sent messages in the lobby containing all players in the session.

6.9 Use case diagram level: 1.11

ID: 9

Name: Menu & Lobby

Level 1.11: Menu & Lobby



Description: This module deals with the in-game lobby management.

It has six sub-modules –

Level 1.11.1: Kick Player

This module deals with kicking players from the multiplayer lobby. Only session host can enforce this action.

Level 1.11.2: Character Selection

This module deals with players selecting different characters to play with.

Level 1.11.3: Pause Game

This module allows players to pause during single player games. Game state is saved in the database.

Level 1.11.4: Resume Game

This module allows players to resume paused single player games. Game state of paused game is retrieved from the database and loaded.

Level 1.11.5: Start Game

This module deals with starting a multiplayer match.

Level 1.11.6: Exit Game

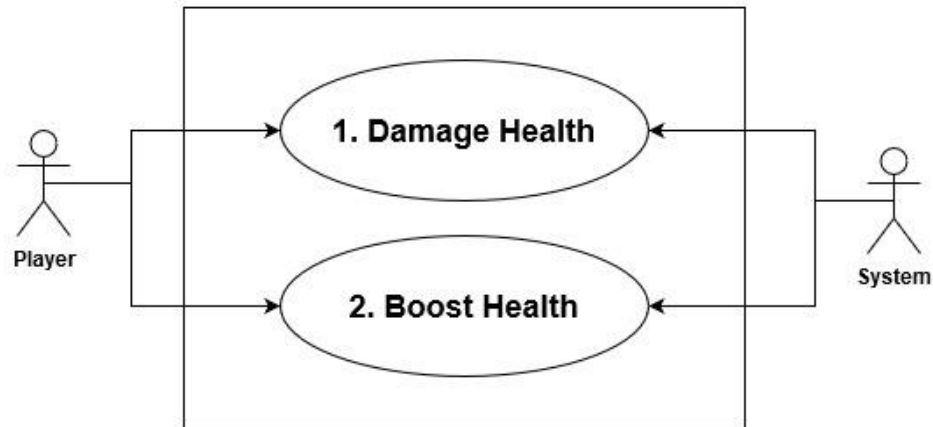
This module allows players to leave or end a game.

6.10 Use case diagram level: 1.12

ID: 10

Name: Health

Level 1.12: Health



Description: This module deals with player health system. It has two sub-modules –

Level 1.12.1: Damage Health

This module deals with decreasing player health value based on whether they have got shot by other players or not.

Level 1.12.2: Boost Health

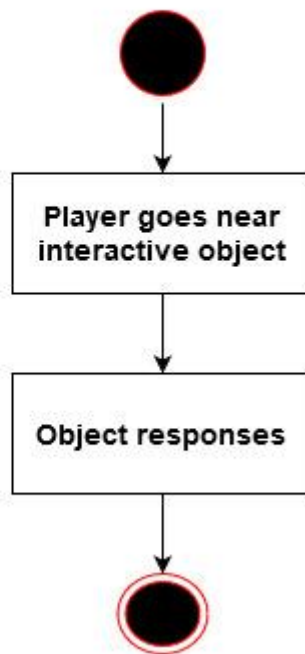
This module deals with boosting player health value if they go to the health boosting positions in the map.

7. Activity Diagrams:

7.1 Activity diagram ID: 1

Name: Interactive Environment

Interactive Environment

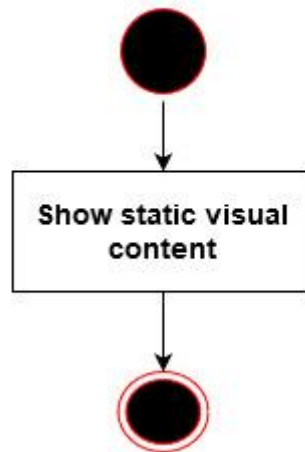


Description: This is the activity diagram of Level 1.1 of the use case diagrams.

7.2 Activity diagram ID: 2

Name: Static Visual

Static Visual

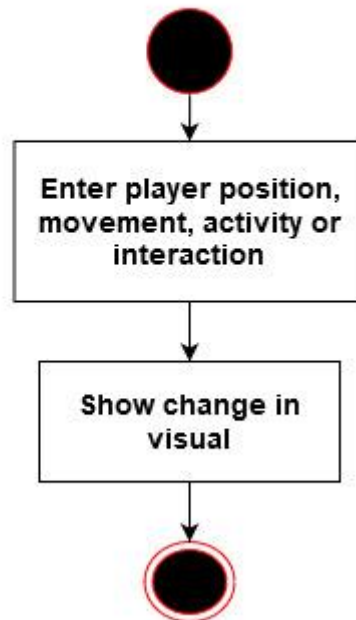


Description: This is the activity diagram of Level 1.2.1 of the use case diagrams.

7.3 Activity diagram ID: 3

Name: Dynamic Visual

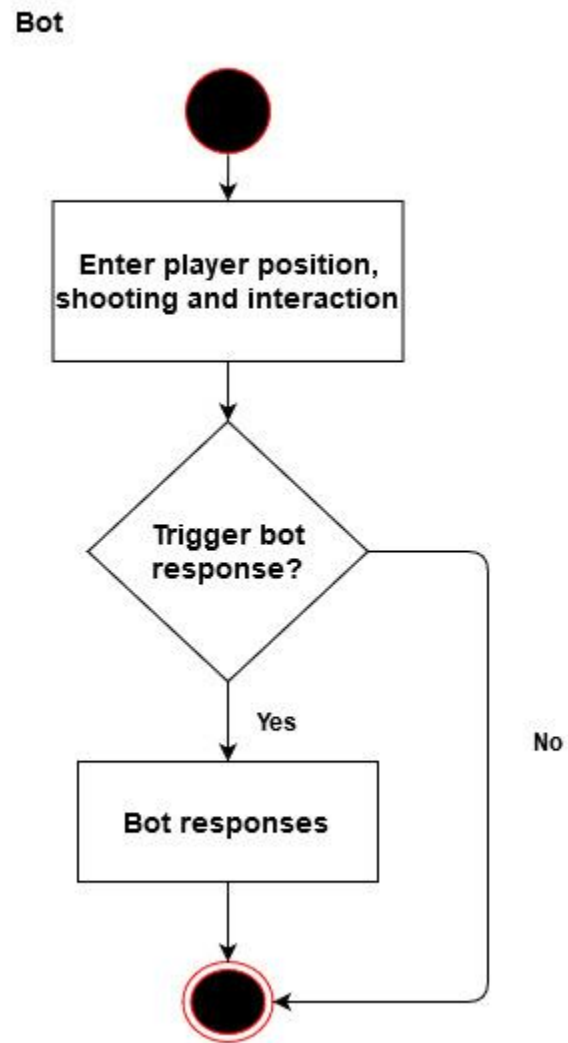
Dynamic Visual



Description: This is the activity diagram of Level 1.2.2 of the use case diagrams.

7.4 Activity diagram ID: 4

Name: Bot

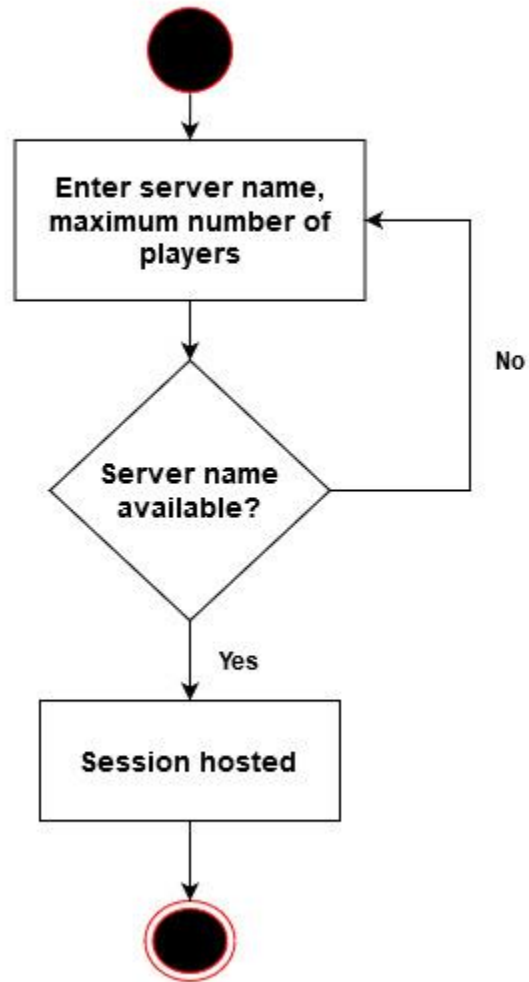


Description: This is the activity diagram of Level 1.3 of the use case diagrams.

7.5 Activity diagram ID: 5

Name: Create Session

Create Session

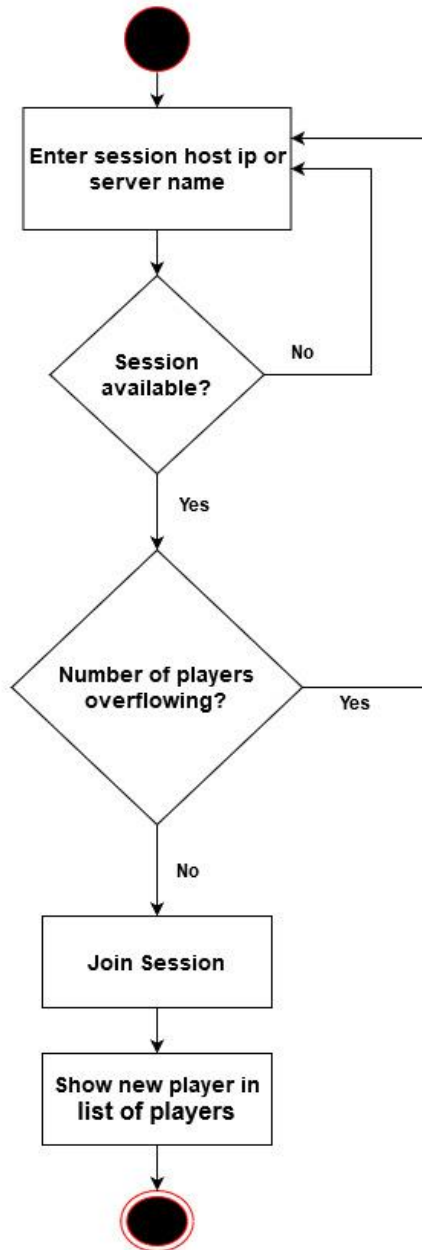


Description: This is the activity diagram of Level 1.4.1 of the use case diagrams.

7.6 Activity diagram ID: 6

Name: Join Session

Join Session

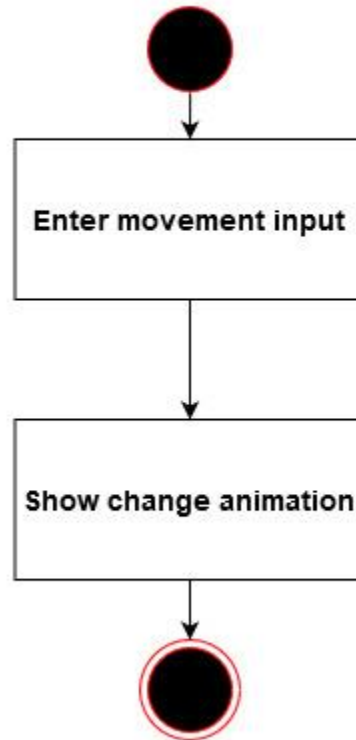


Description: This is the activity diagram of Level 1.4.2 of the use case diagrams.

7.7 Activity diagram ID: 7

Name: Movement

Movement

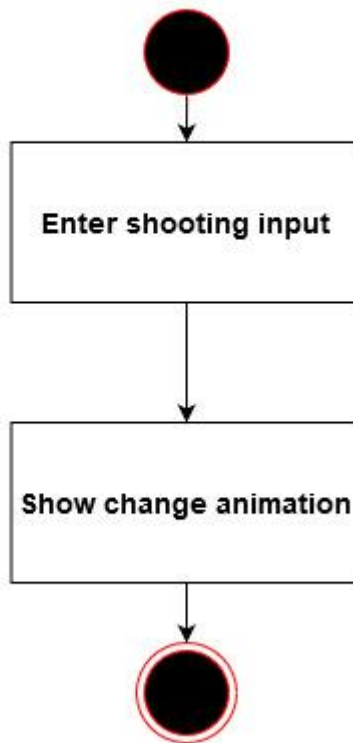


Description: This is the activity diagram of Level 1.5 of the use case diagrams.

7.8 Activity diagram ID: 8

Name: Shooting Mechanics

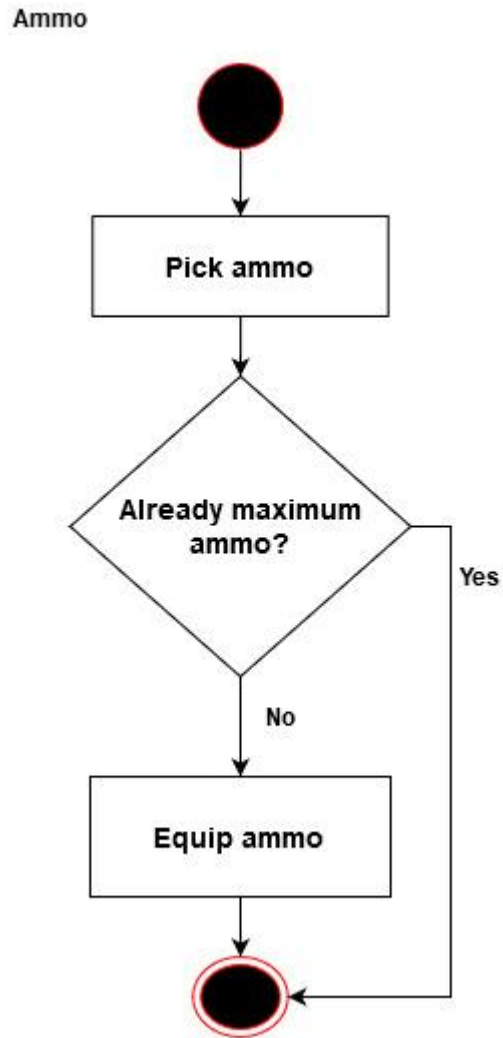
Shooting Mechanics



Description: This is the activity diagram of Level 1.6.1 of the use case diagrams.

7.9 Activity diagram ID: 9

Name: Ammo



Description: This is the activity diagram of Level 1.6.2.1 of the use case diagrams.

7.10 Activity diagram ID: 10

Name: Pick Weapon

Pick Weapon

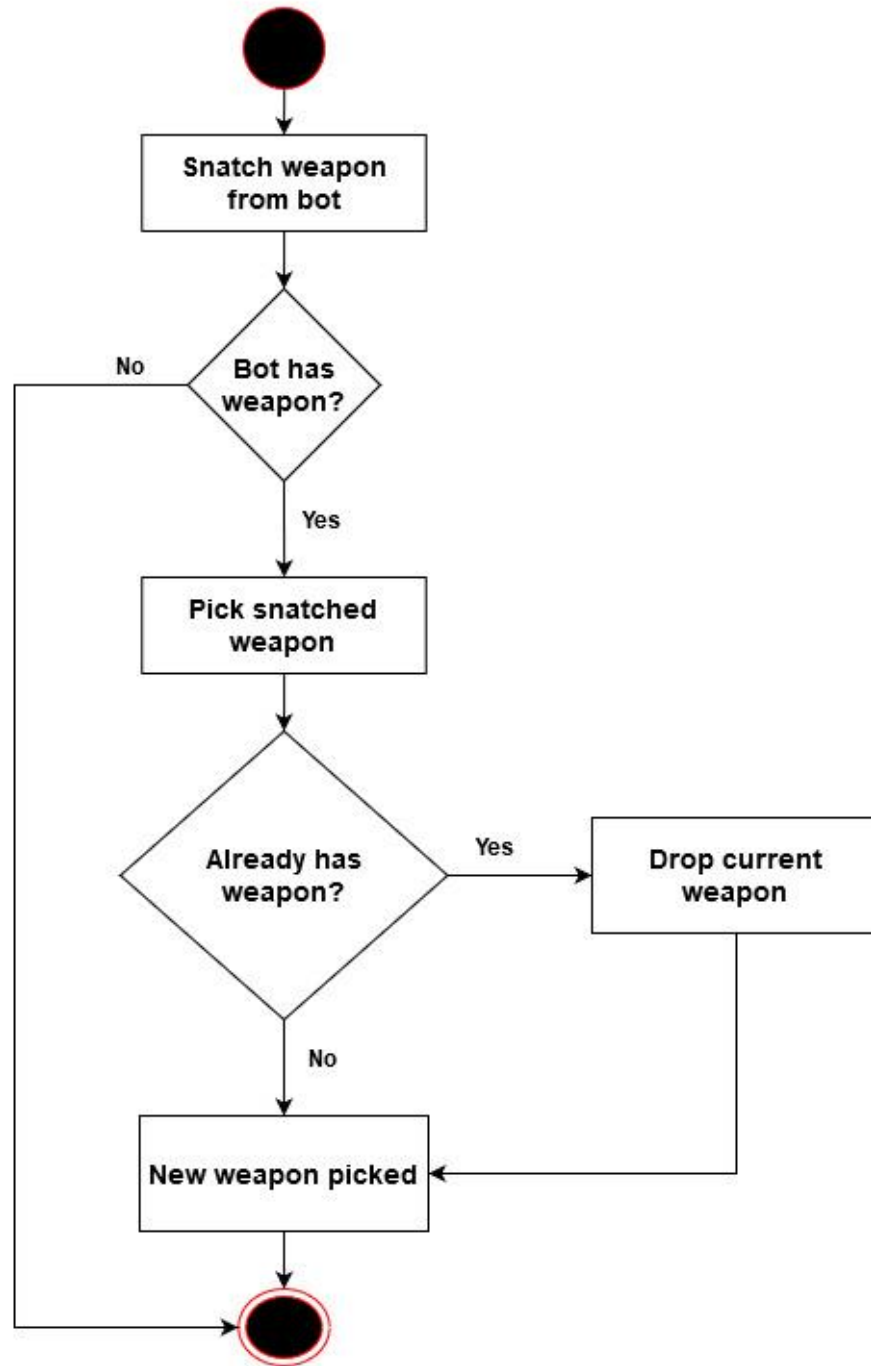


Description: This is the activity diagram of Level 1.6.2.2 of the use case diagrams.

7.11 Activity diagram ID: 11

Name: Snatch Weapon

Snatch Weapon

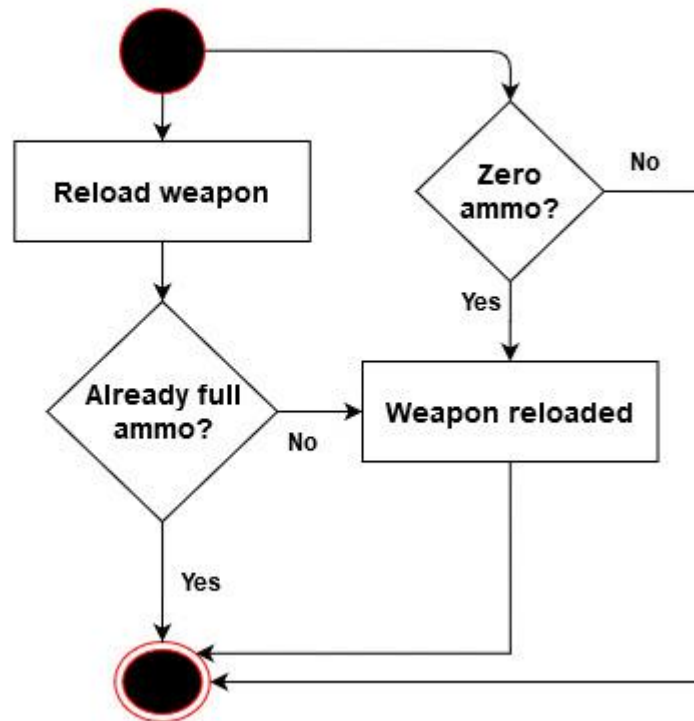


Description: This is the activity diagram of Level 1.6.2.3 of the use case diagrams.

7.12 Activity diagram ID: 12

Name: Reload Weapon

Reload Weapon

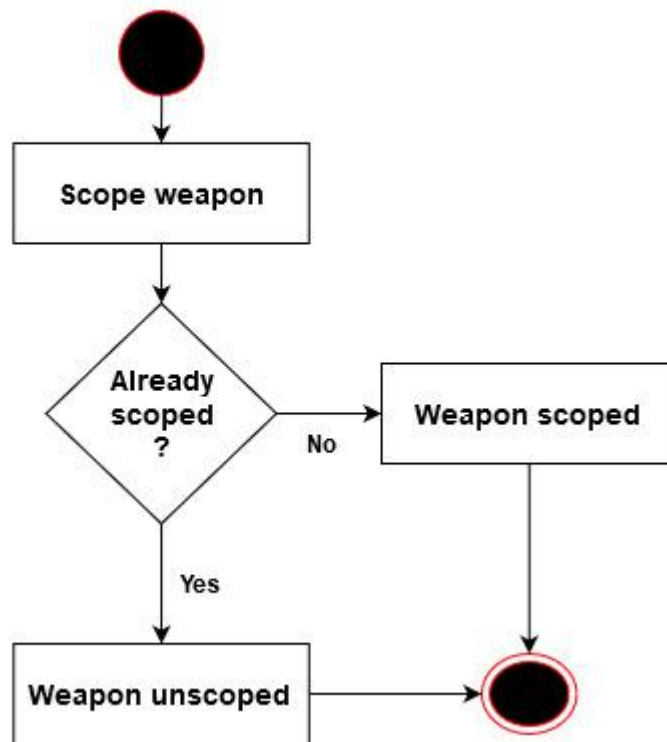


Description: This is the activity diagram of Level 1.6.2.4 of the use case diagrams.

7.13 Activity diagram ID: 13

Name: Scope Weapon

Scope Weapon

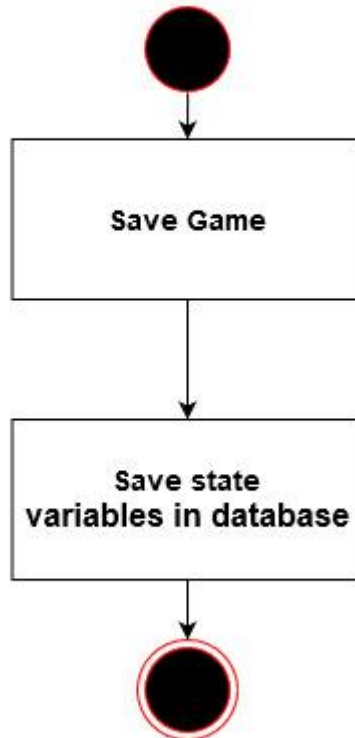


Description: This is the activity diagram of Level 1.6.2.5 of the use case diagrams.

7.14 Activity diagram ID: 14

Name: Game State

Game State

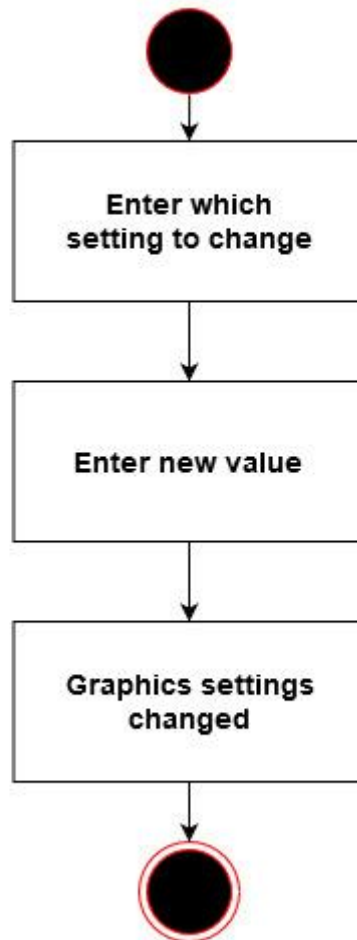


Description: This is the activity diagram of Level 1.7 of the use case diagrams.

7.15 Activity diagram ID: 15

Name: Graphics Settings

Graphics Settings

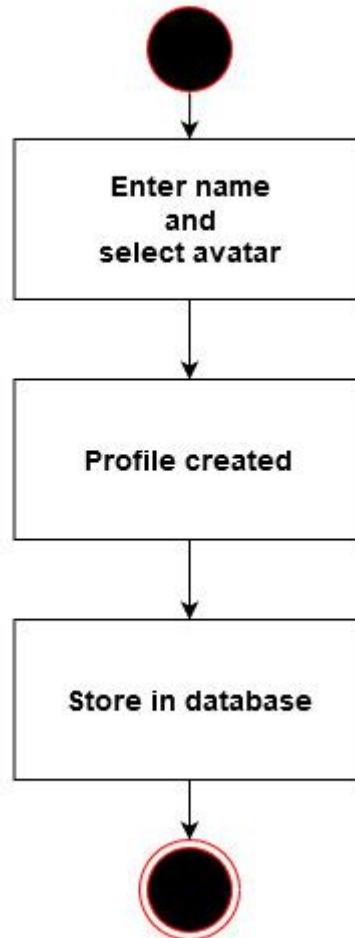


Description: This is the activity diagram of Level 1.8 of the use case diagrams.

7.16 Activity diagram ID: 16

Name: Profile Creation

Profile Creation

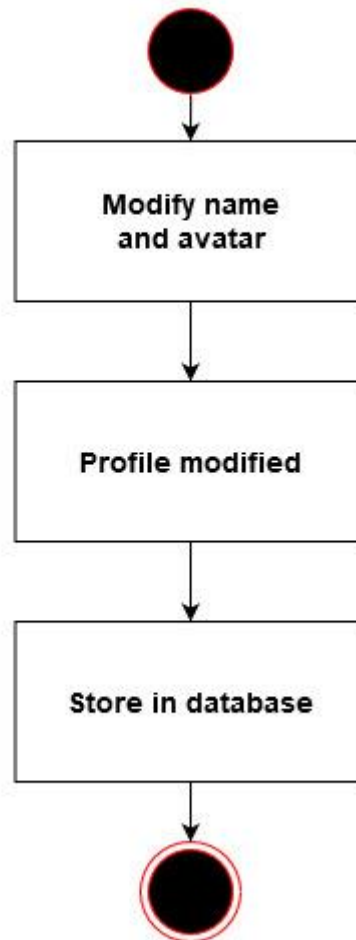


Description: This is the activity diagram of Level 1.9.1 of the use case diagrams.

7.17 Activity diagram ID: 17

Name: Profile Modification

Profile Modification

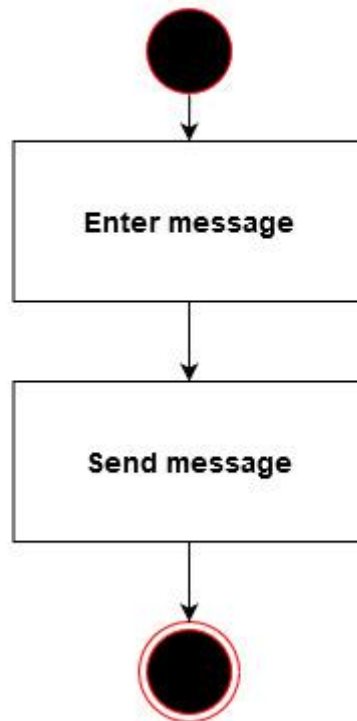


Description: This is the activity diagram of Level 1.9.2 of the use case diagrams.

7.18 Activity diagram ID: 18

Name: Send Message

Send Message

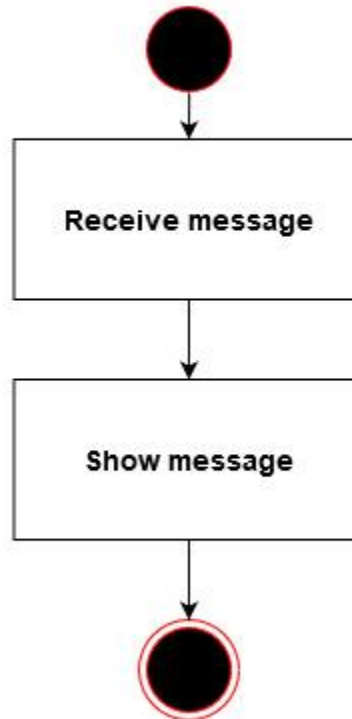


Description: This is the activity diagram of Level 1.10.1 of the use case diagrams.

7.19 Activity diagram ID: 19

Name: Receive Message

Receive Message



Description: This is the activity diagram of Level 1.10.2 of the use case diagrams.

7.20 Activity diagram ID: 20

Name: Kick Player

Kick Player

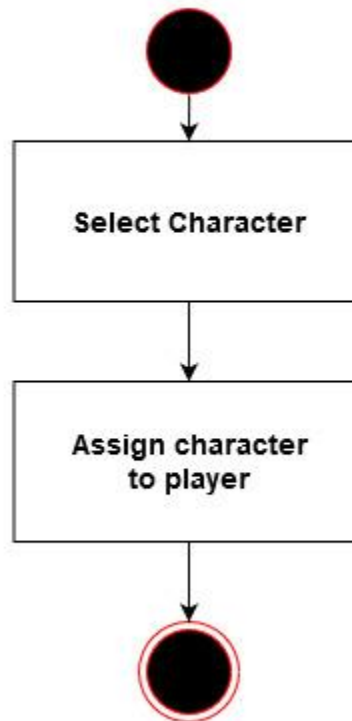


Description: This is the activity diagram of Level 1.11.1 of the use case diagrams.

7.21 Activity diagram ID: 21

Name: Select Character

Select Character

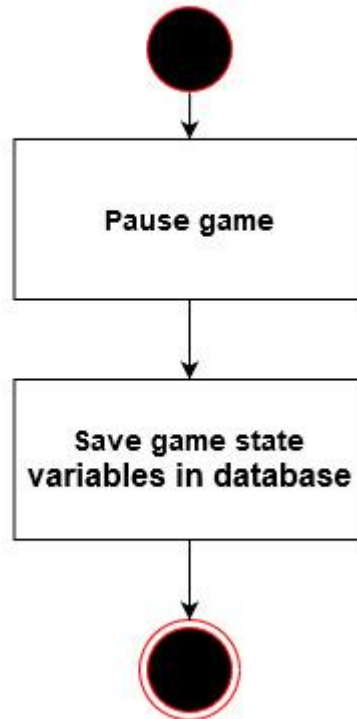


Description: This is the activity diagram of Level 1.11.2 of the use case diagrams.

7.22 Activity diagram ID: 22

Name: Pause Game

Pause Game

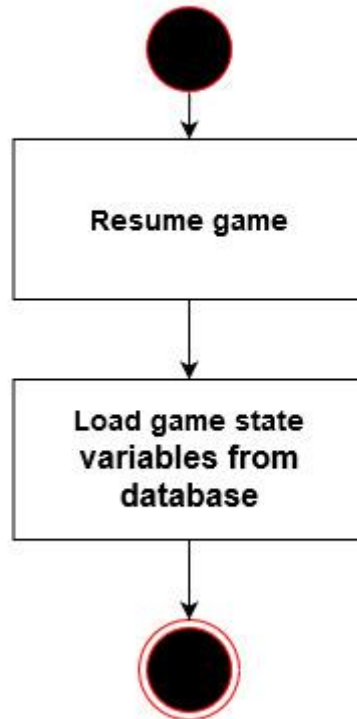


Description: This is the activity diagram of Level 1.11.3 of the use case diagrams.

7.23 Activity diagram ID: 23

Name: Resume Game

Resume Game



Description: This is the activity diagram of Level 1.11.4 of the use case diagrams.

7.24 Activity diagram ID: 24

Name: Start Game

Start Game

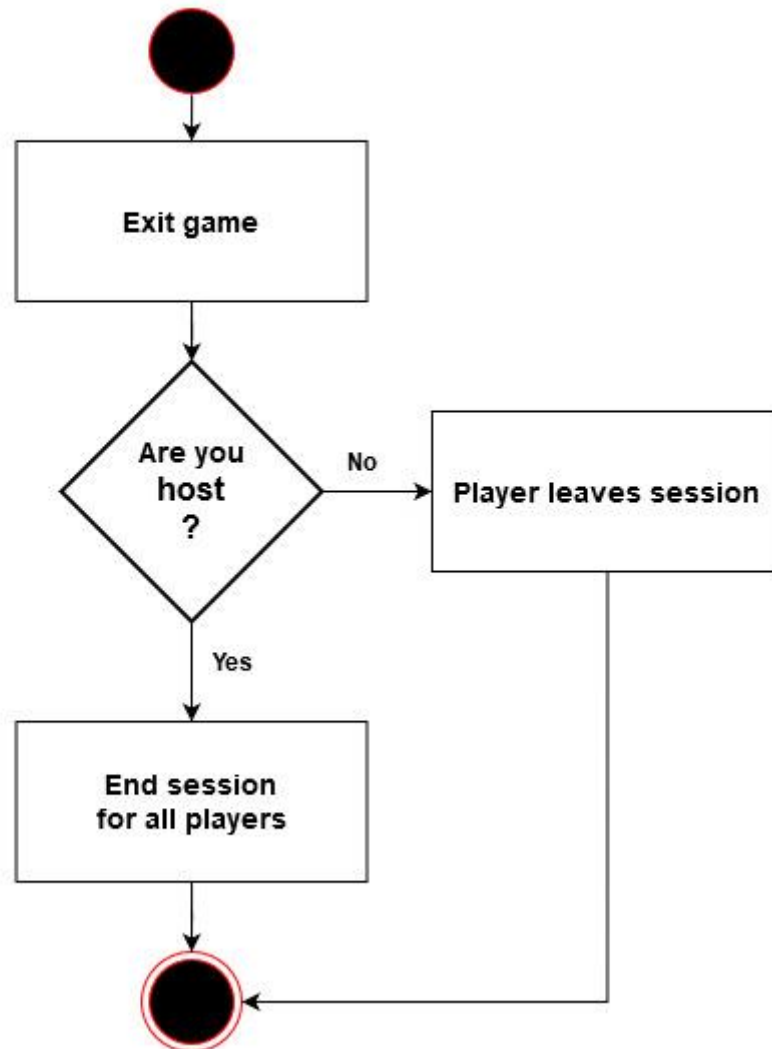


Description: This is the activity diagram of Level 1.11.5 of the use case diagrams.

7.25 Activity diagram ID: 25

Name: Exit Game

Exit Game

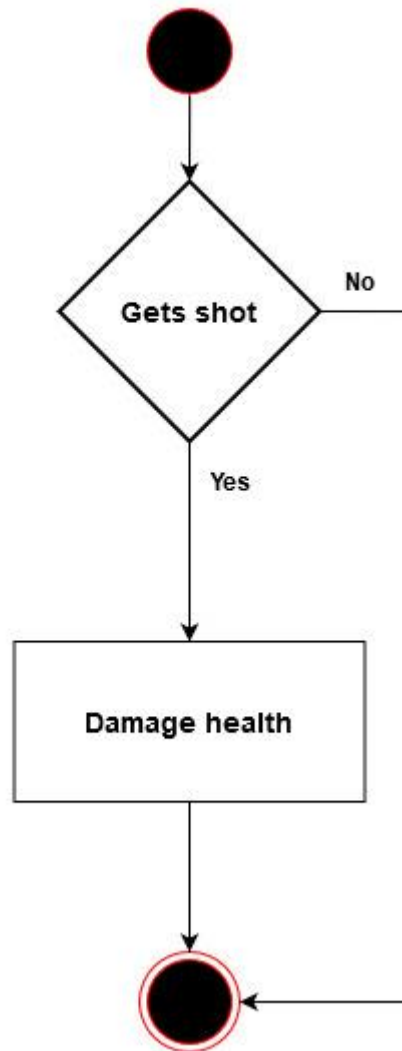


Description: This is the activity diagram of Level 1.11.6 of the use case diagrams.

7.26 Activity diagram ID: 26

Name: Damage Health

Damage Health



Description: This is the activity diagram of Level 1.12.1 of the use case diagrams.

7.27 Activity diagram ID: 27

Name: Boost Health

Boost Health

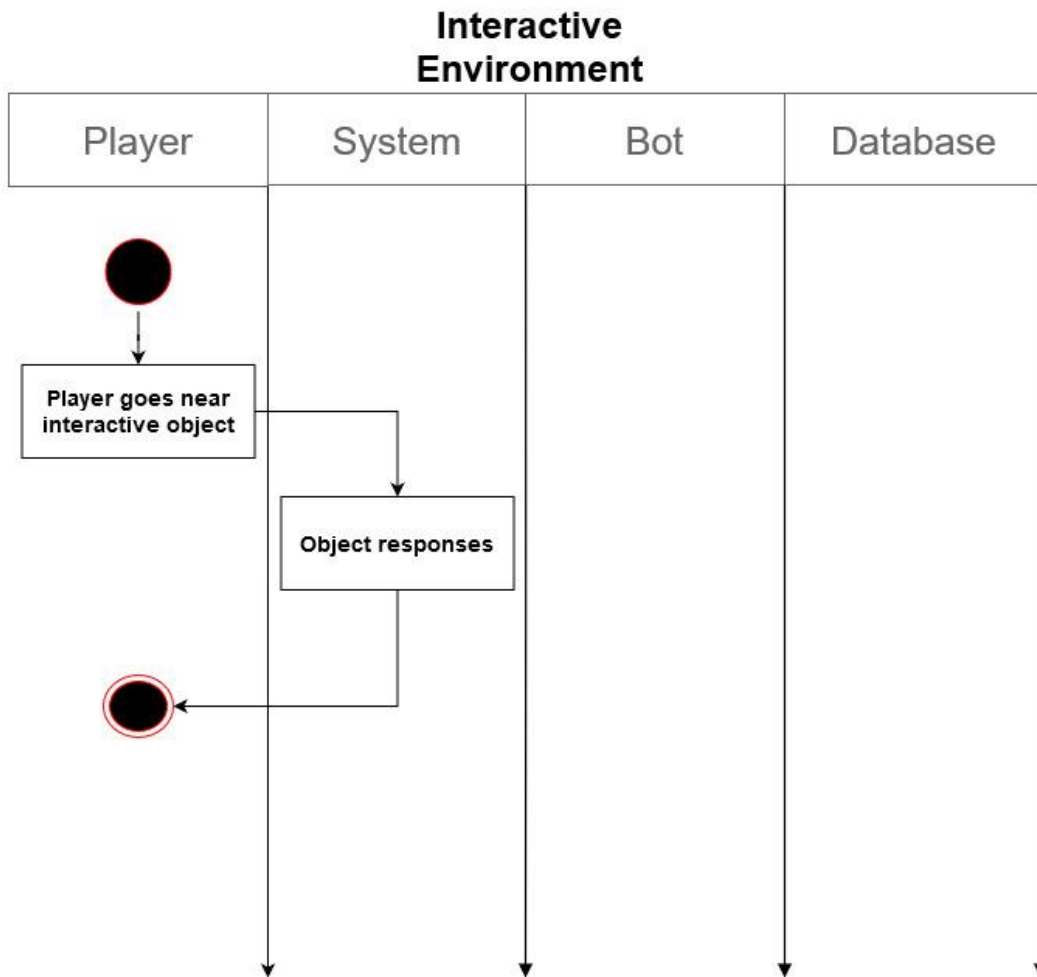


Description: This is the activity diagram of Level 1.12.2 of the use case diagrams.

8. Swimlane Diagrams:

8.1 Swimlane Diagram ID: 1

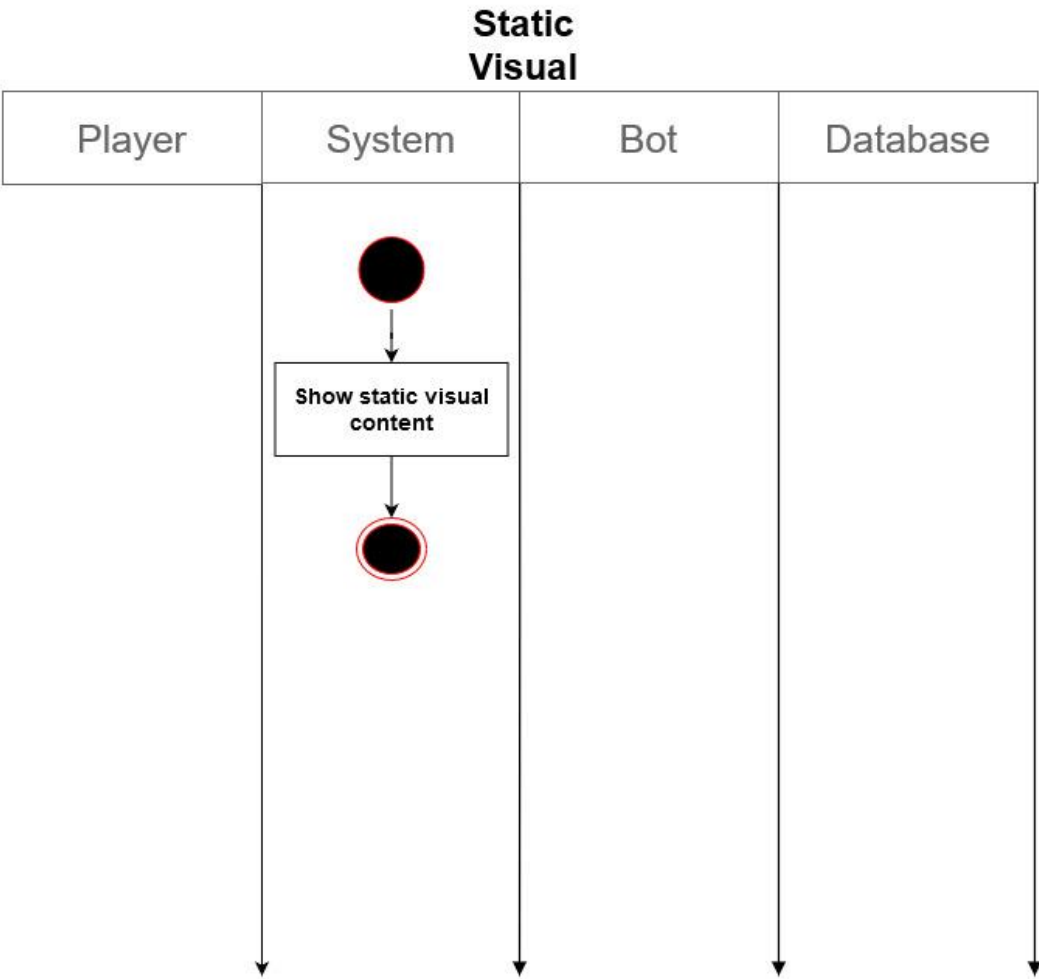
Name: Interactive Environment



Description: This is the swimlane diagram of the corresponding activity diagram ID:1. It shows how different actors are involved with each activities of the Interactive Environment module.

8.2 Swimlane Diagram ID: 2

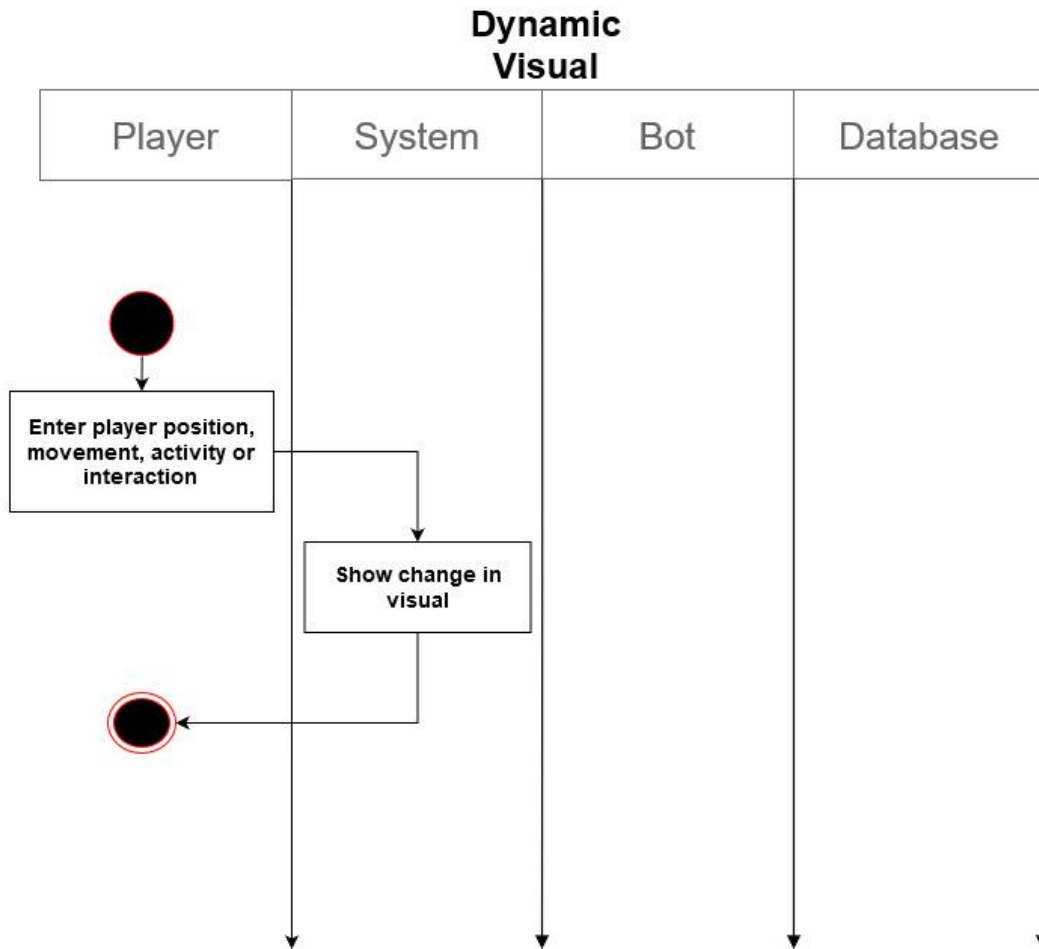
Name: Static Visual



Description: This is the swimlane diagram of the corresponding activity diagram ID:2. It shows how different actors are involved with each activities of the Static Visual module.

8.3 Swimlane Diagram ID: 3

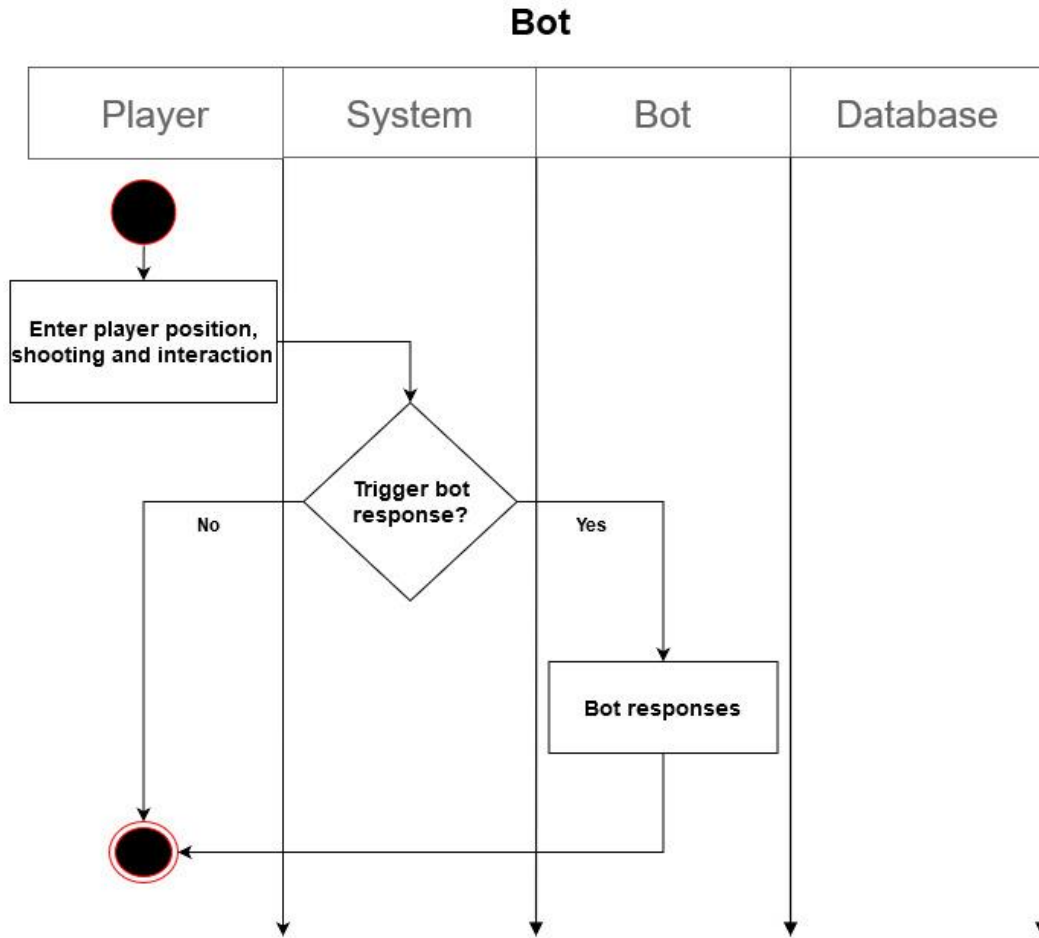
Name: Dynamic Visual



Description: This is the swimlane diagram of the corresponding activity diagram ID:3. It shows how different actors are involved with each activities of the Dynamic Visual module.

8.4 Swimlane Diagram ID: 4

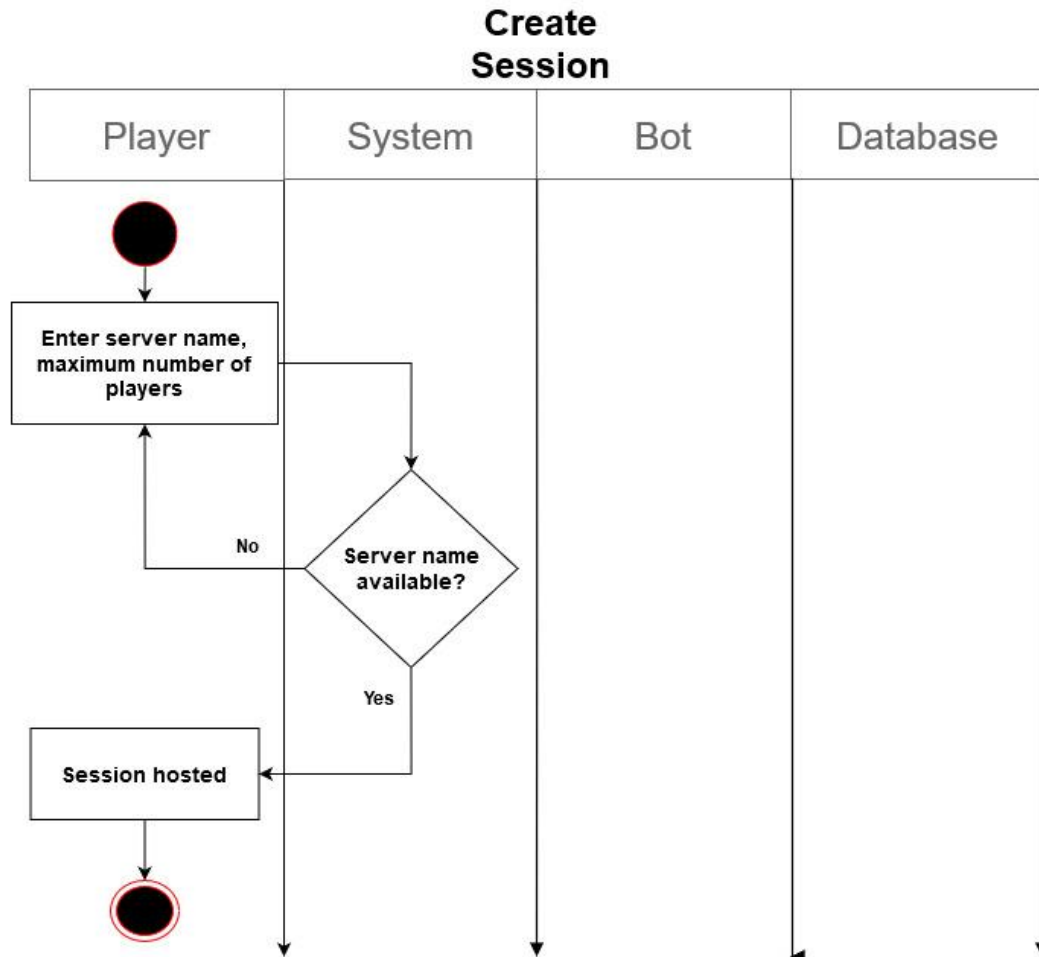
Name: Bot



Description: This is the swimlane diagram of the corresponding activity diagram ID:4. It shows how different actors are involved with each activities of the Bot module.

8.5 Swimlane Diagram ID: 5

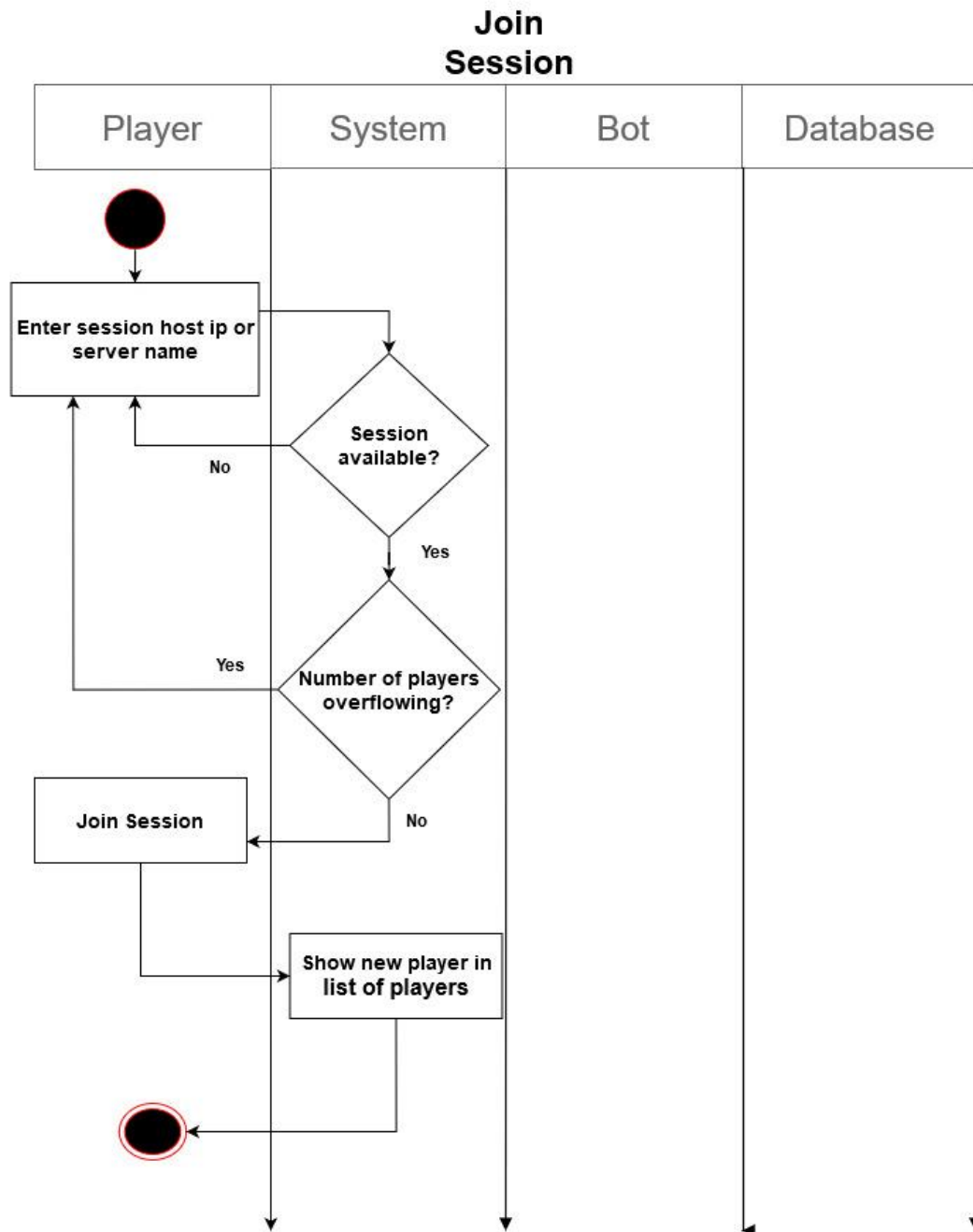
Name: Create Session



Description: This is the swimlane diagram of the corresponding activity diagram ID:5. It shows how different actors are involved with each activities of the module.

8.6 Swimlane Diagram ID: 6

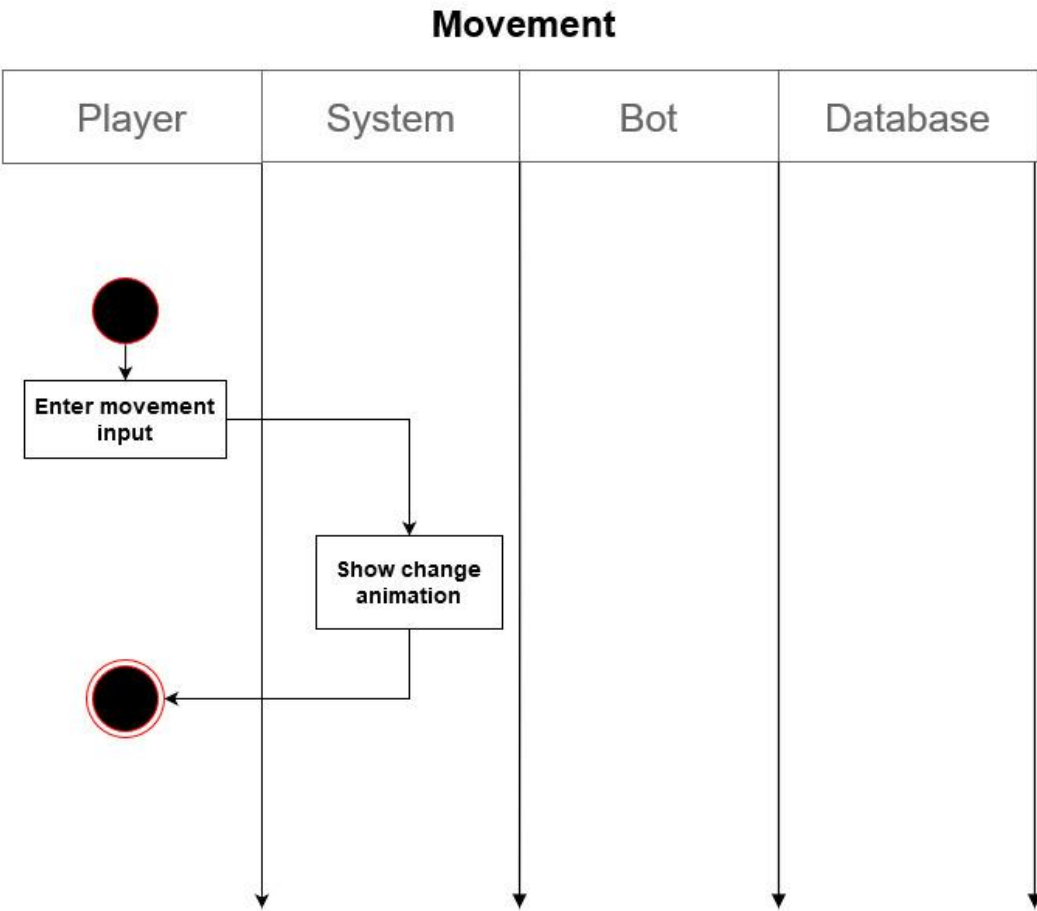
Name: Join Session



Description: This is the swimlane diagram of the corresponding activity diagram ID:6. It shows how different actors are involved with each activities of the module.

8.7 Swimlane Diagram ID: 7

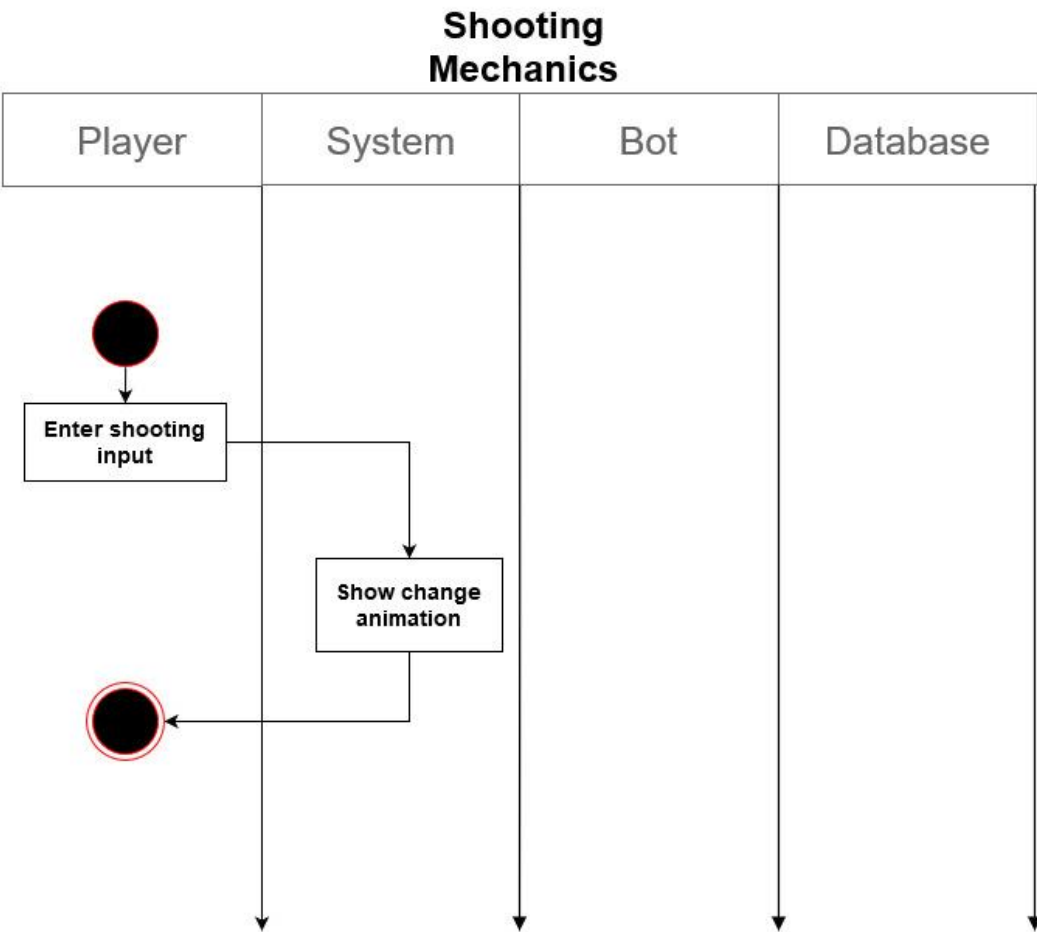
Name: Movement



Description: This is the swimlane diagram of the corresponding activity diagram ID:7. It shows how different actors are involved with each activities of the module.

8.8 Swimlane Diagram ID: 8

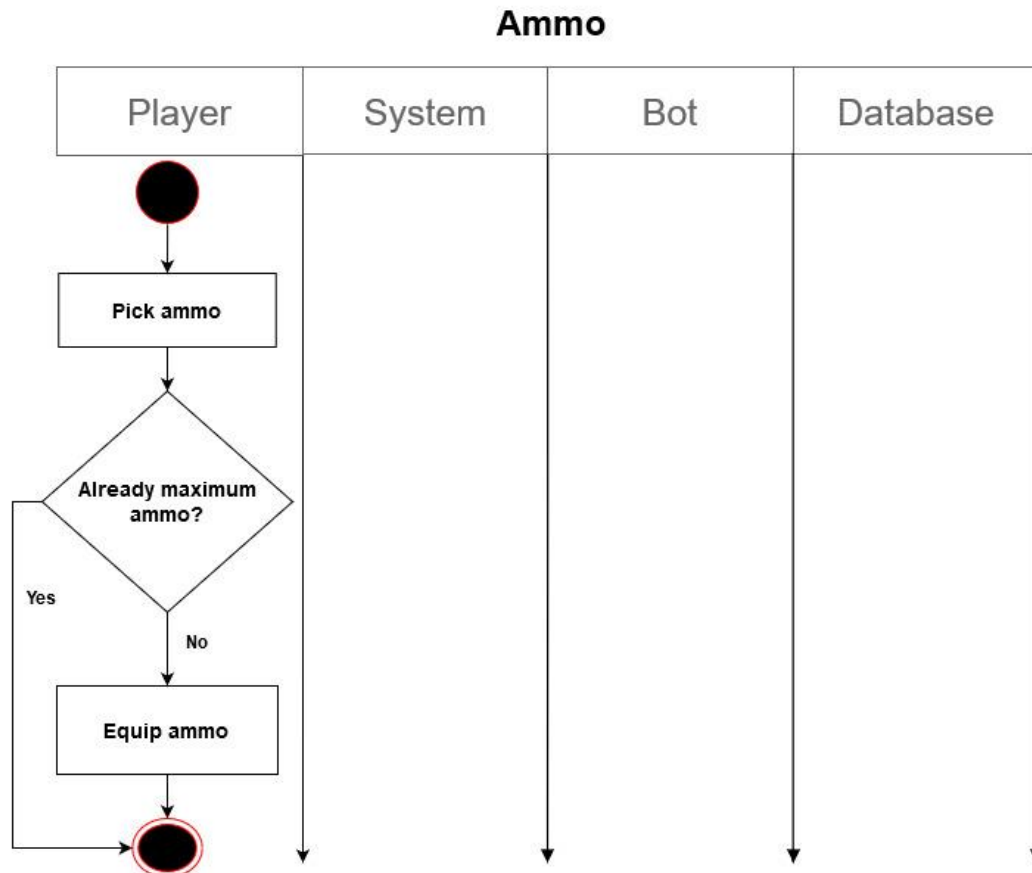
Name: Shooting Mechanics



Description: This is the swimlane diagram of the corresponding activity diagram ID:8. It shows how different actors are involved with each activities of the module.

8.9 Swimlane Diagram ID: 9

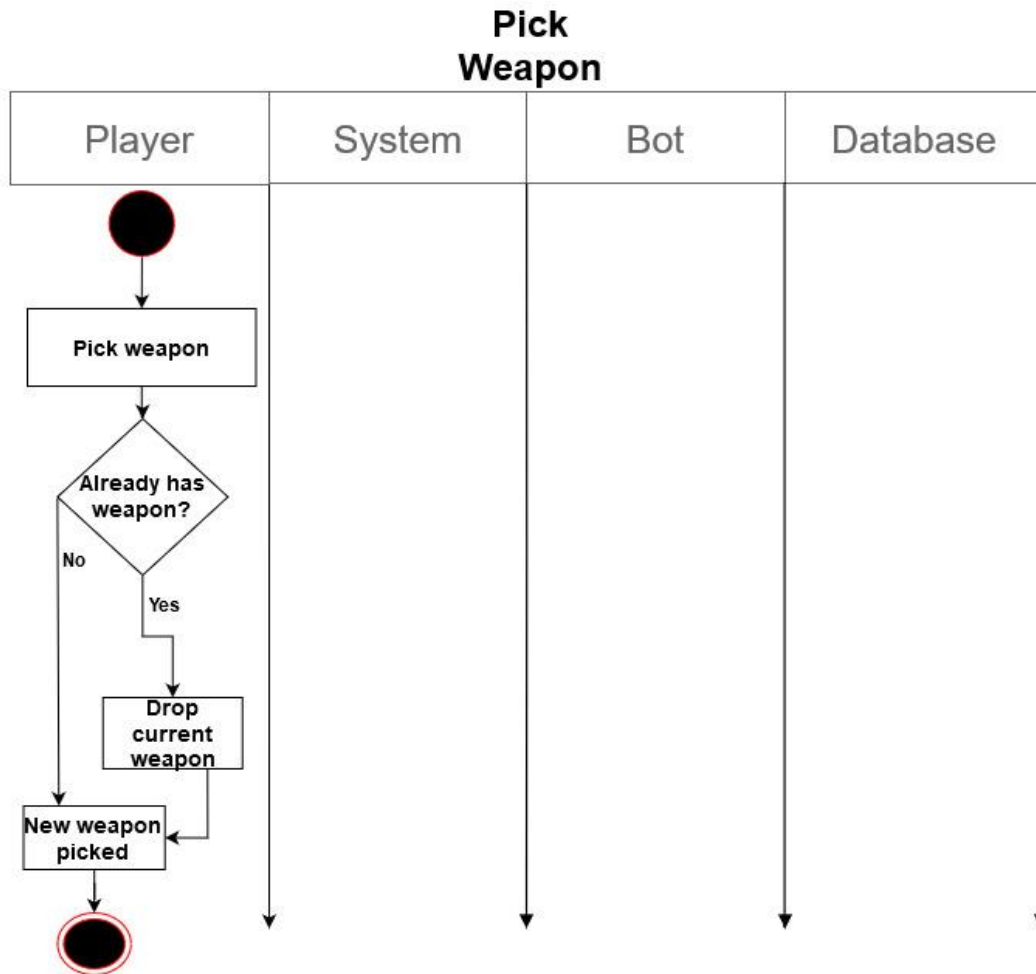
Name: Ammo



Description: This is the swimlane diagram of the corresponding activity diagram ID:9. It shows how different actors are involved with each activities of the module.

8.10 Swimlane Diagram ID: 10

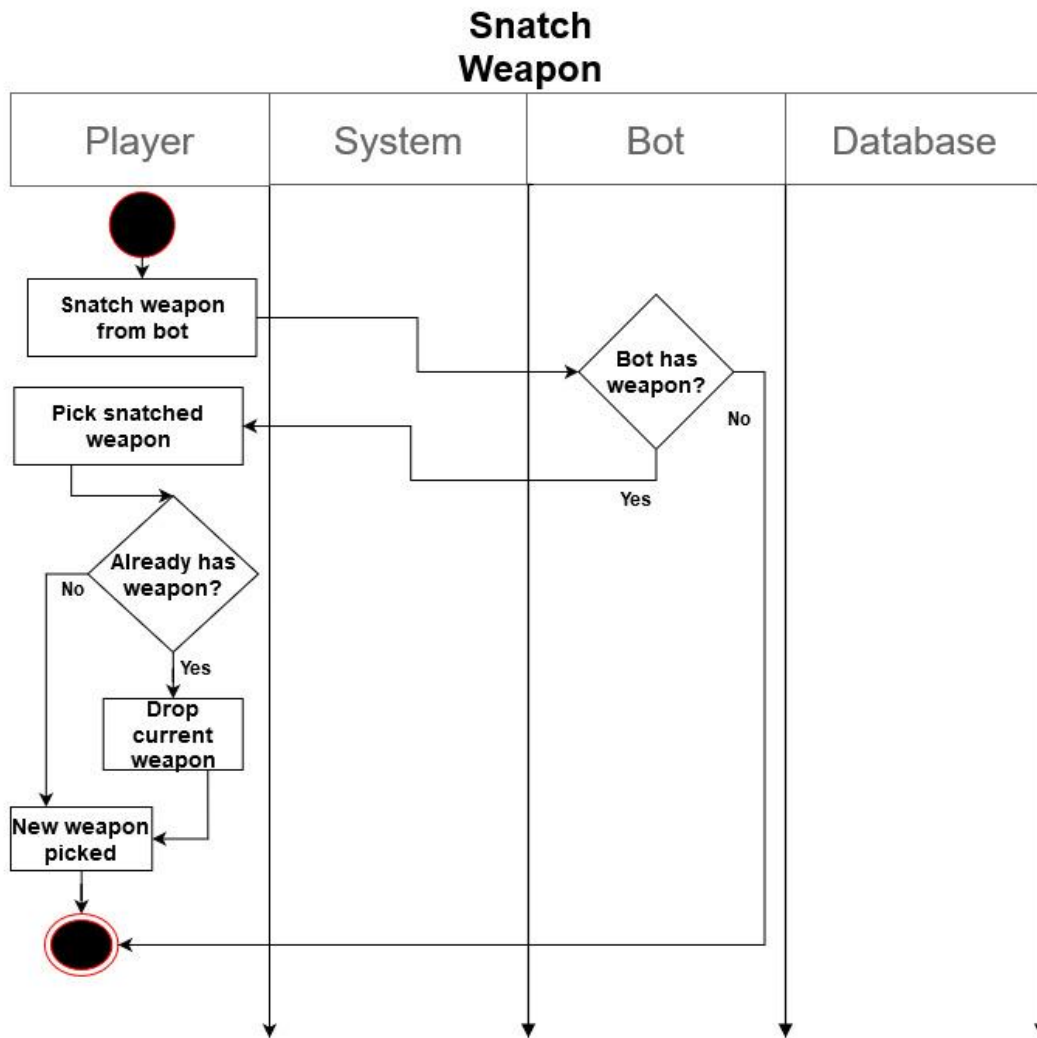
Name: Pick Weapon



Description: This is the swimlane diagram of the corresponding activity diagram ID:10. It shows how different actors are involved with each activities of the Pick Weapon module.

8.11 Swimlane Diagram ID: 11

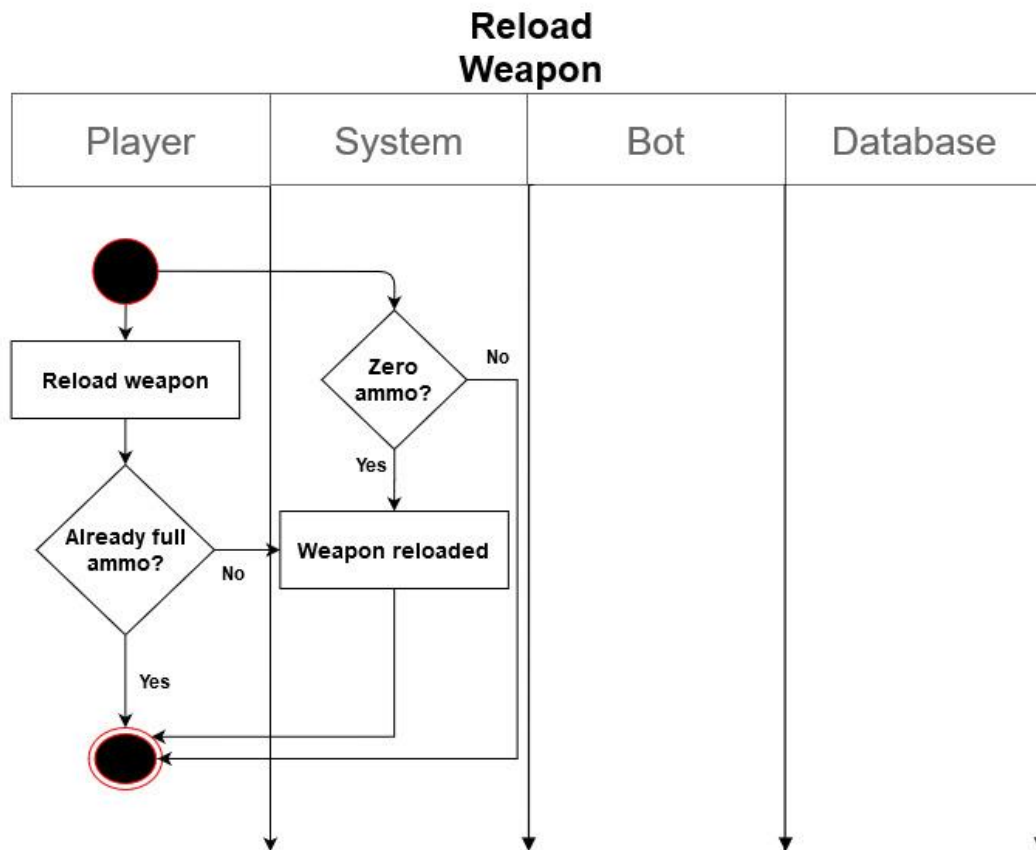
Name: Snatch Weapon



Description: This is the swimlane diagram of the corresponding activity diagram ID:11. It shows how different actors are involved with each activities of the module.

8.12 Swimlane Diagram ID: 12

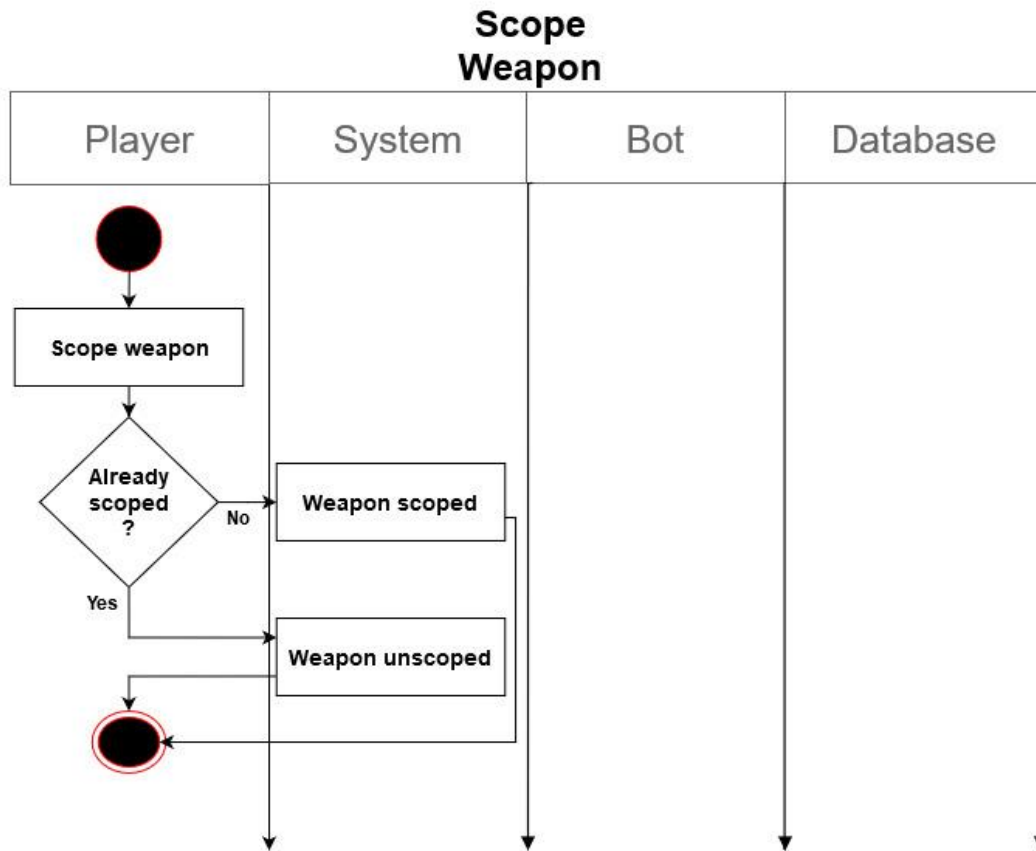
Name: Reload Weapon



Description: This is the swimlane diagram of the corresponding activity diagram ID:12. It shows how different actors are involved with each activities of the module.

8.13 Swimlane Diagram ID: 13

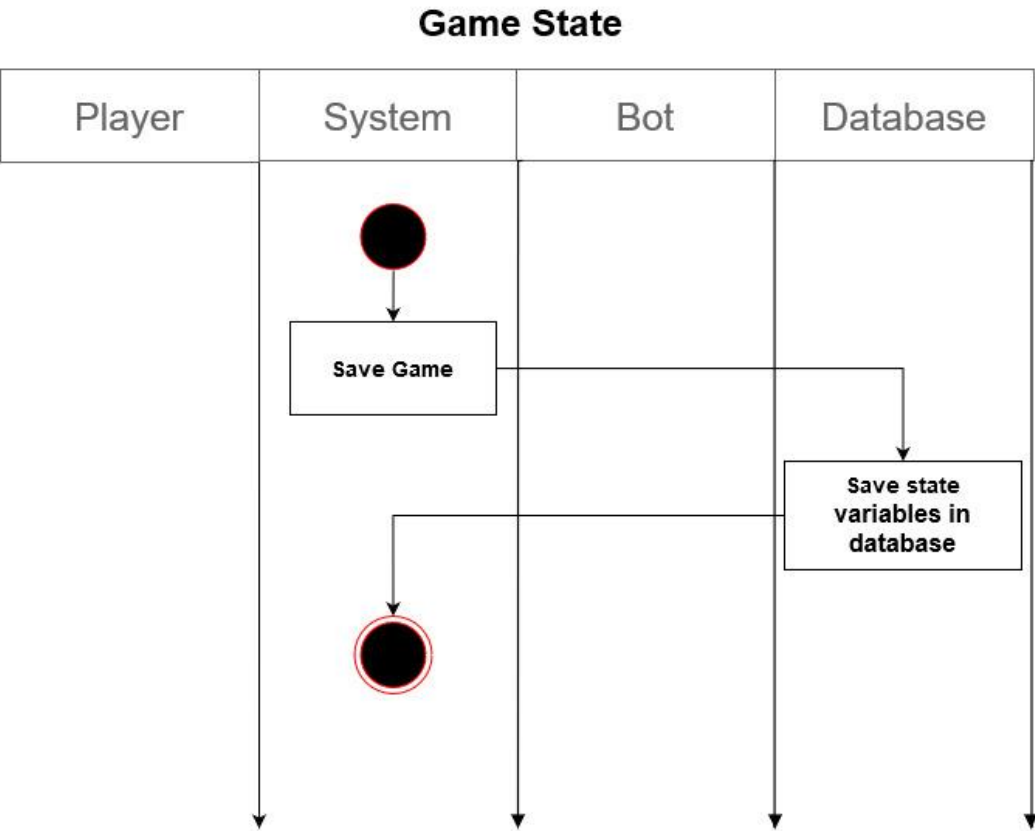
Name: Scope Weapon



Description: This is the swimlane diagram of the corresponding activity diagram ID:13. It shows how different actors are involved with each activities of the module.

8.14 Swimlane Diagram ID: 14

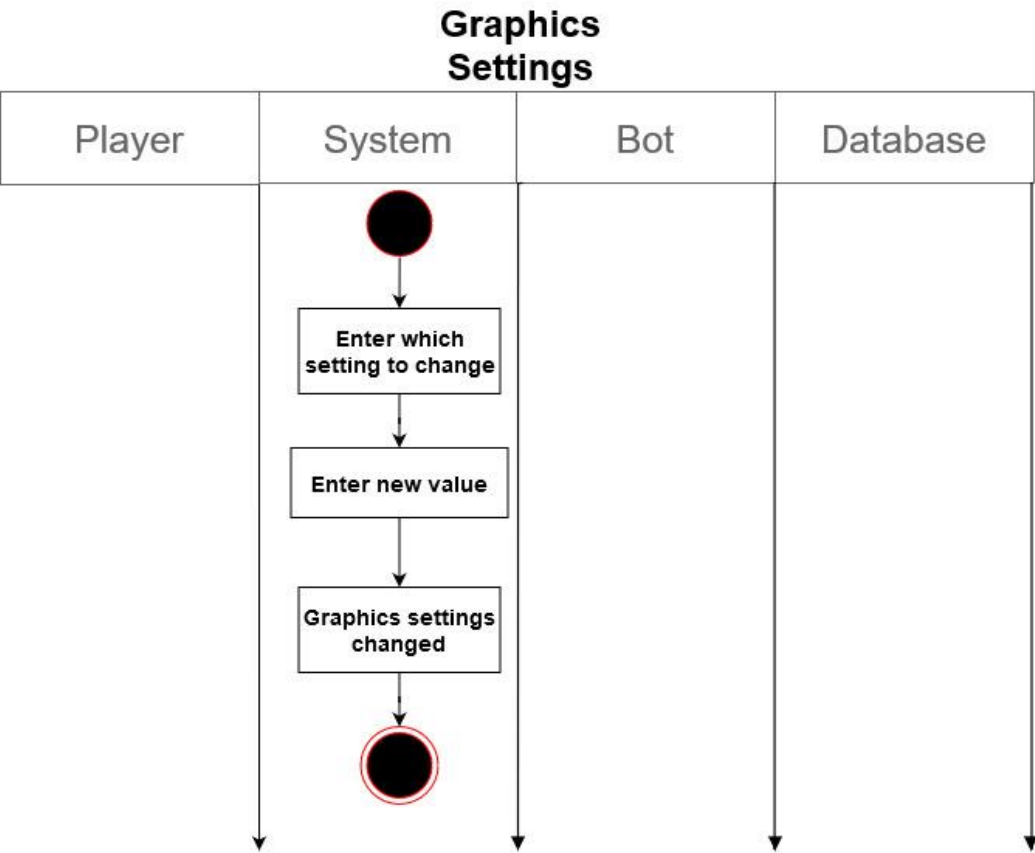
Name: Game State



Description: This is the swimlane diagram of the corresponding activity diagram ID:14. It shows how different actors are involved with each activities of the Game State module.

8.15 Swimlane Diagram ID: 15

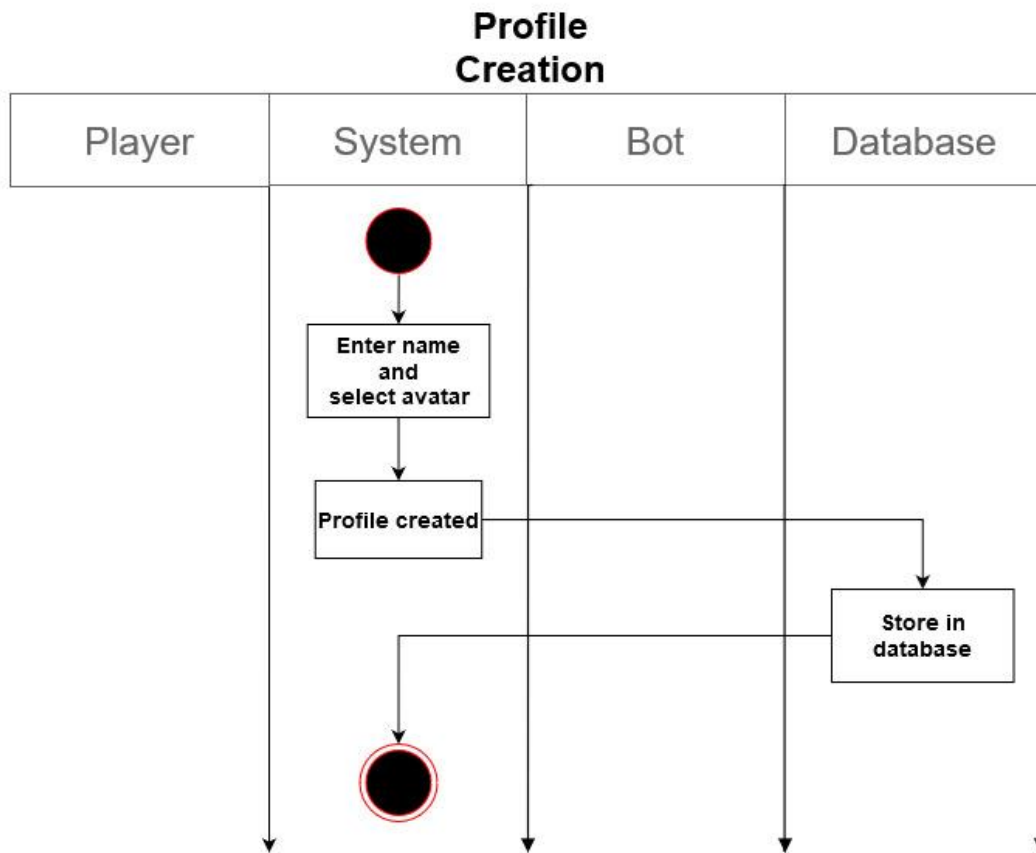
Name: Graphics Settings



Description: This is the swimlane diagram of the corresponding activity diagram ID: 15. It shows how different actors are involved with each activities of the module.

8.16 Swimlane Diagram ID: 16

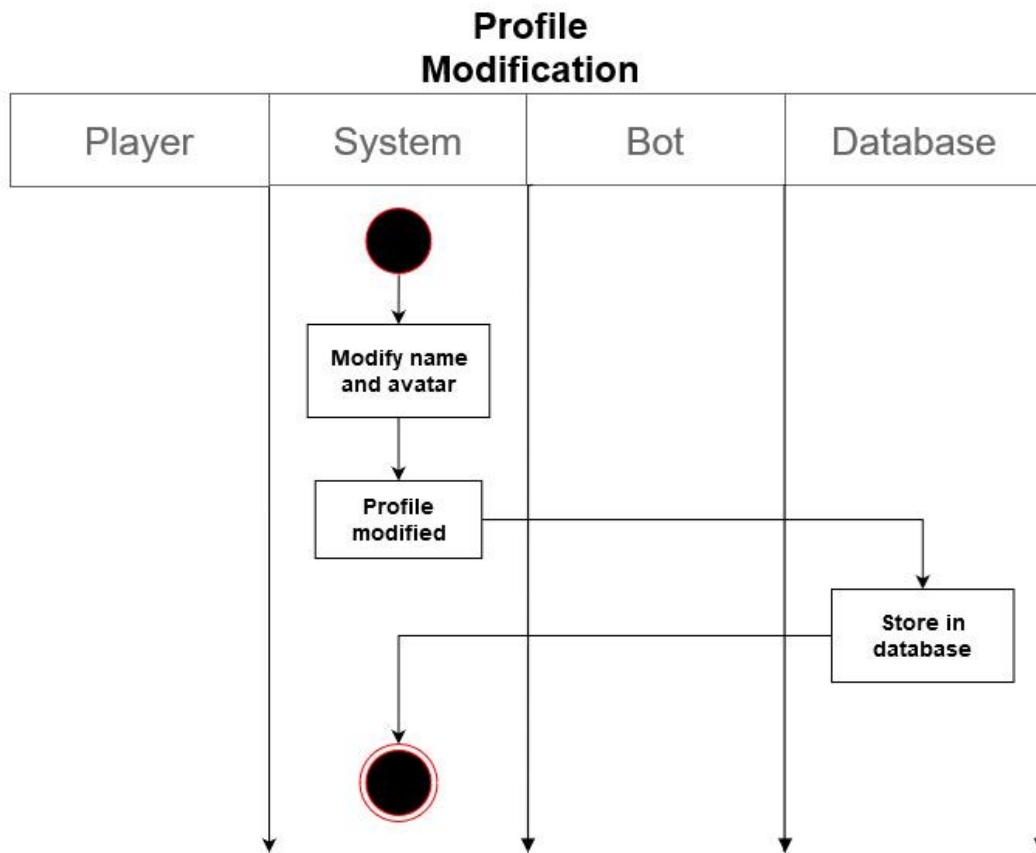
Name: Profile Creation



Description: This is the swimlane diagram of the corresponding activity diagram ID: 16. It shows how different actors are involved with each activities of the module.

8.17 Swimlane Diagram ID: 17

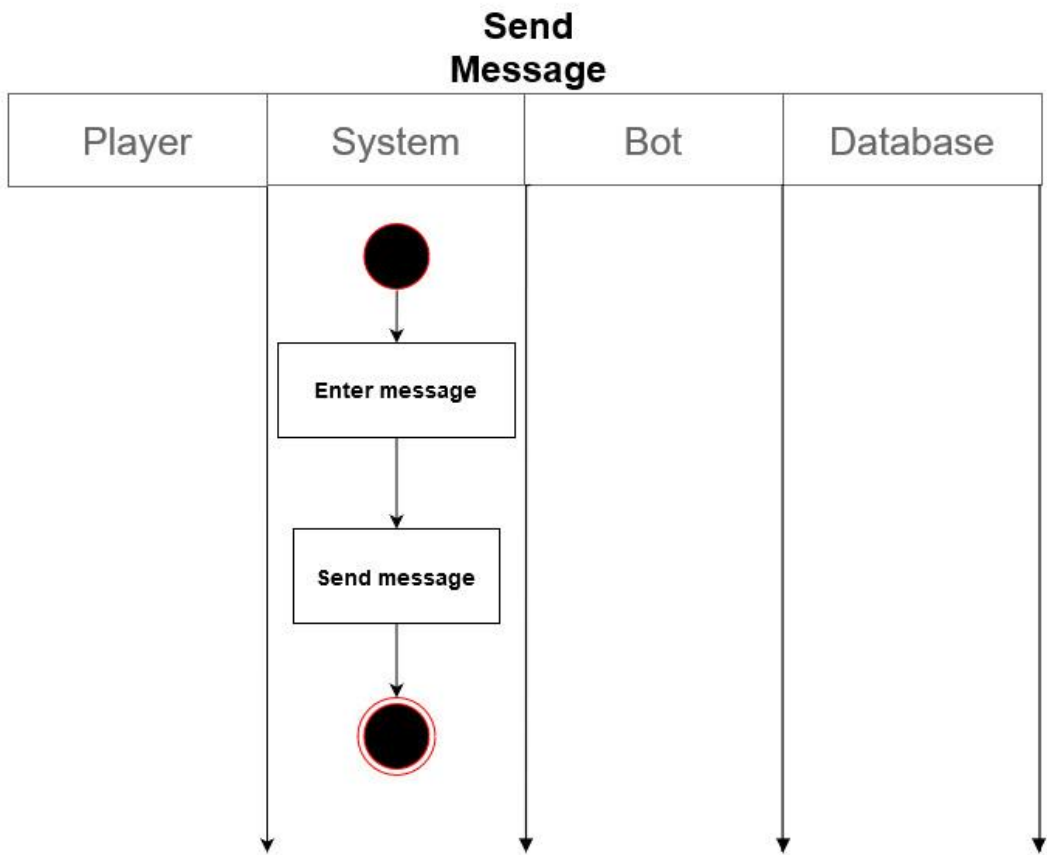
Name: Profile Modification



Description: This is the swimlane diagram of the corresponding activity diagram ID: 17. It shows how different actors are involved with each activities of the module.

8.18 Swimlane Diagram ID: 18

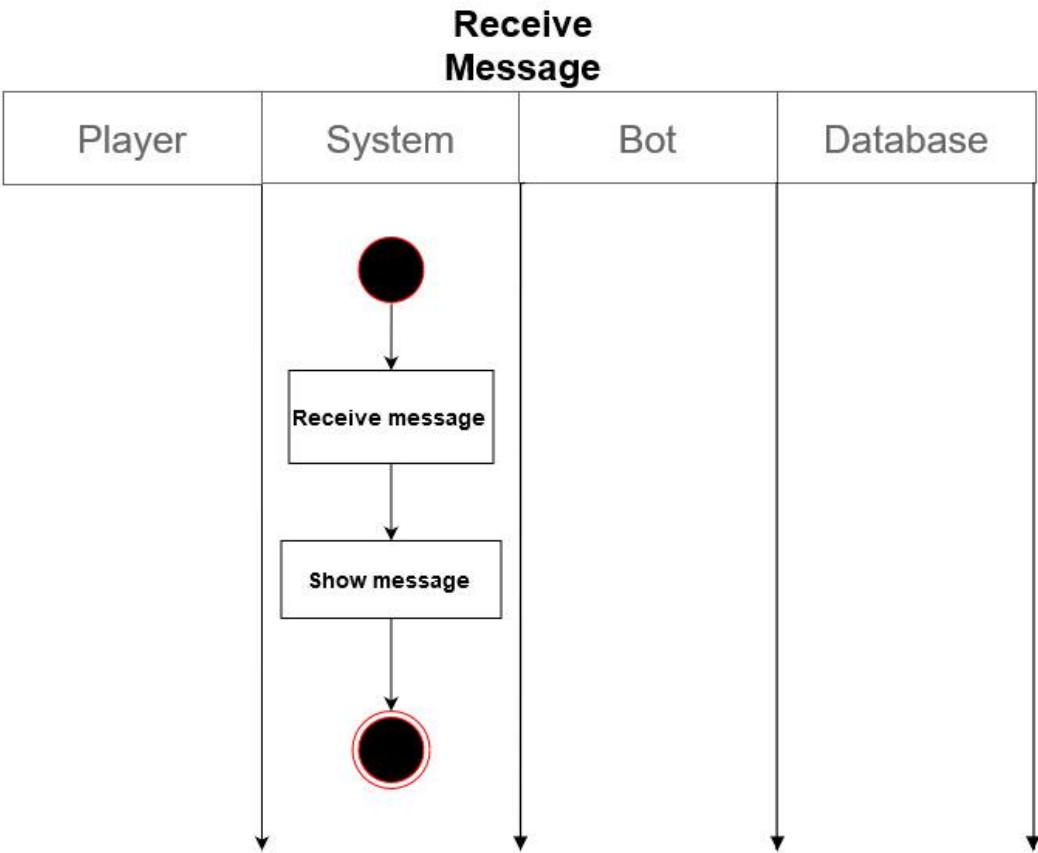
Name: Send Message



Description: This is the swimlane diagram of the corresponding activity diagram ID: 18. It shows how different actors are involved with each activities of the module.

8.19 Swimlane Diagram ID: 19

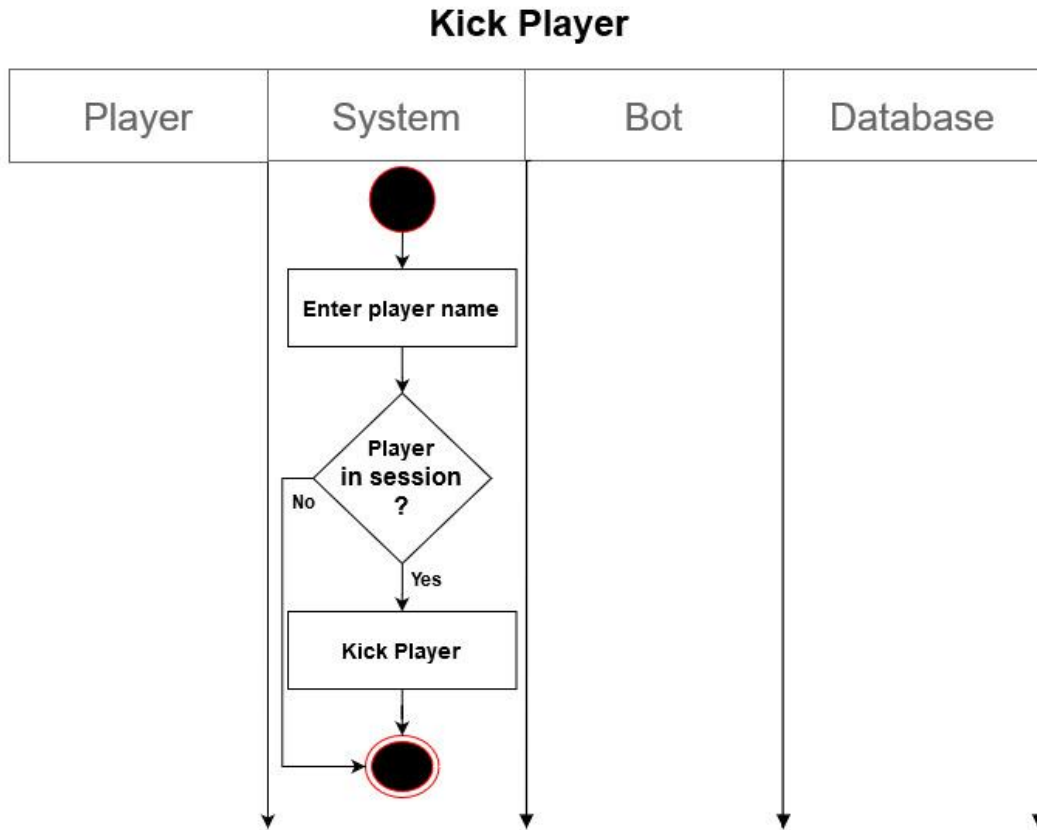
Name: Receive Message



Description: This is the swimlane diagram of the corresponding activity diagram ID: 19. It shows how different actors are involved with each activities of the module.

8.20 Swimlane Diagram ID: 20

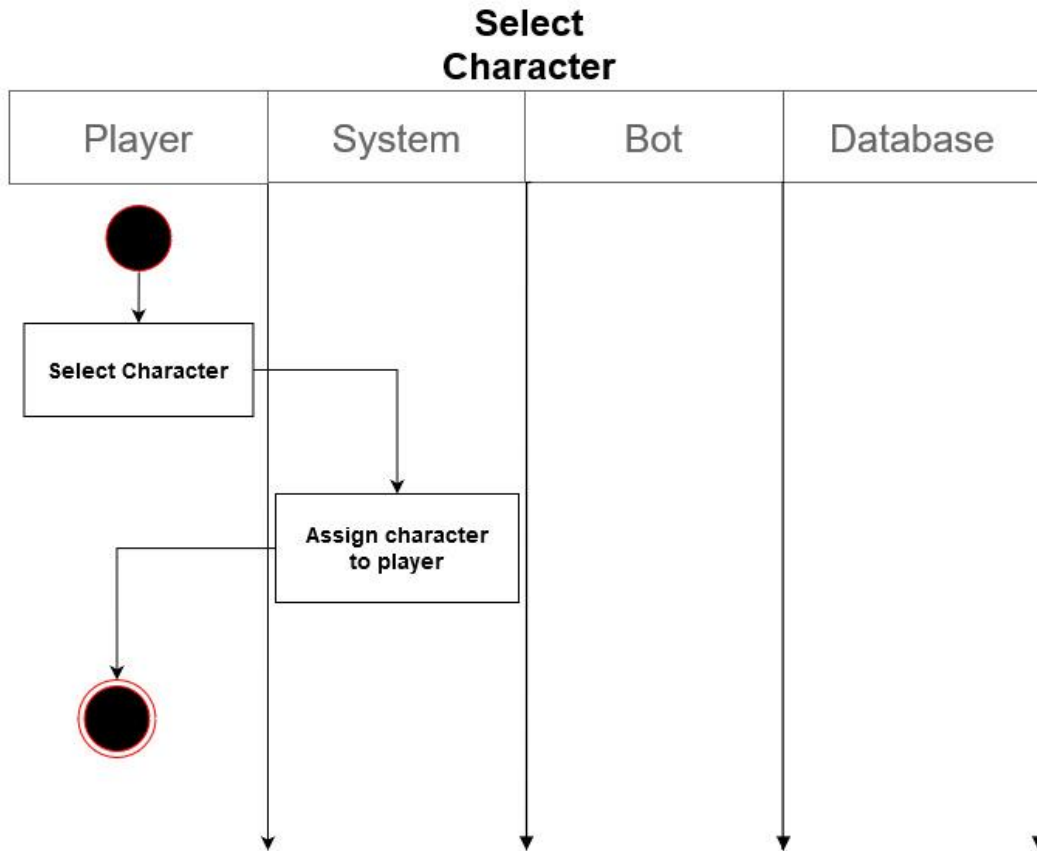
Name: Kick Player



Description: This is the swimlane diagram of the corresponding activity diagram ID: 20. It shows how different actors are involved with each activities of the module.

8.21 Swimlane Diagram ID: 21

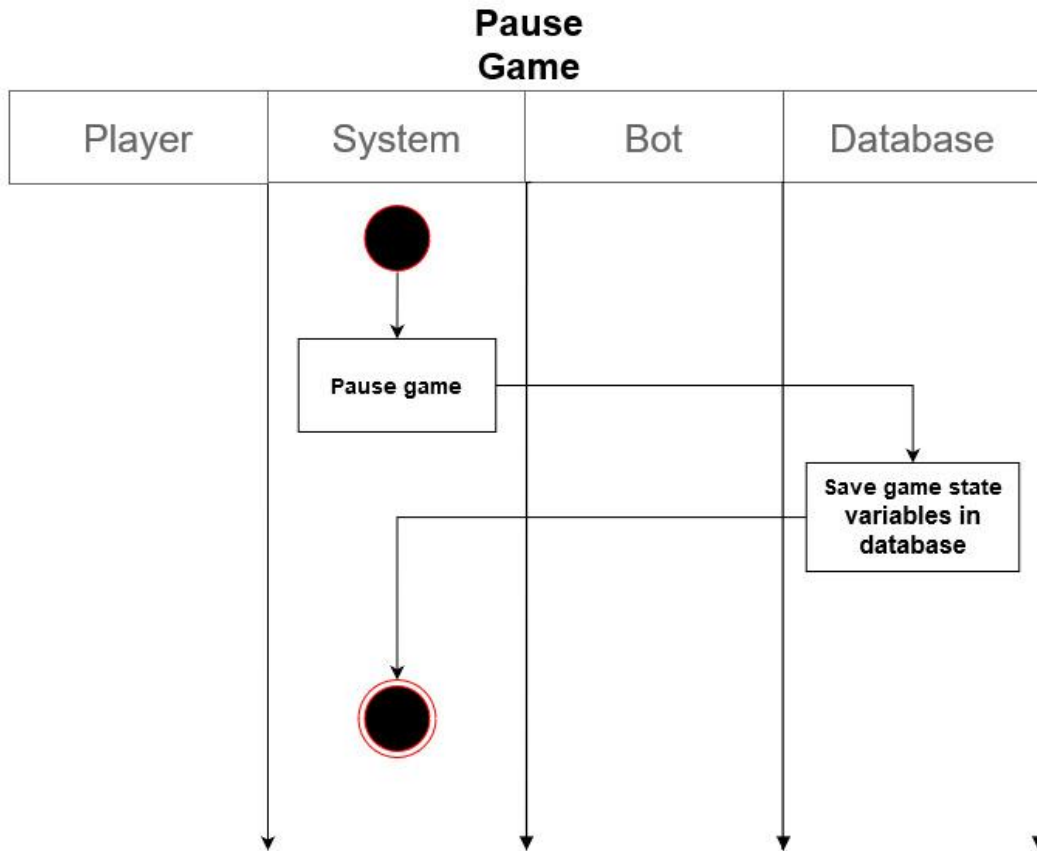
Name: Select Character



Description: This is the swimlane diagram of the corresponding activity diagram ID: 21. It shows how different actors are involved with each activities of the module.

8.22 Swimlane Diagram ID: 22

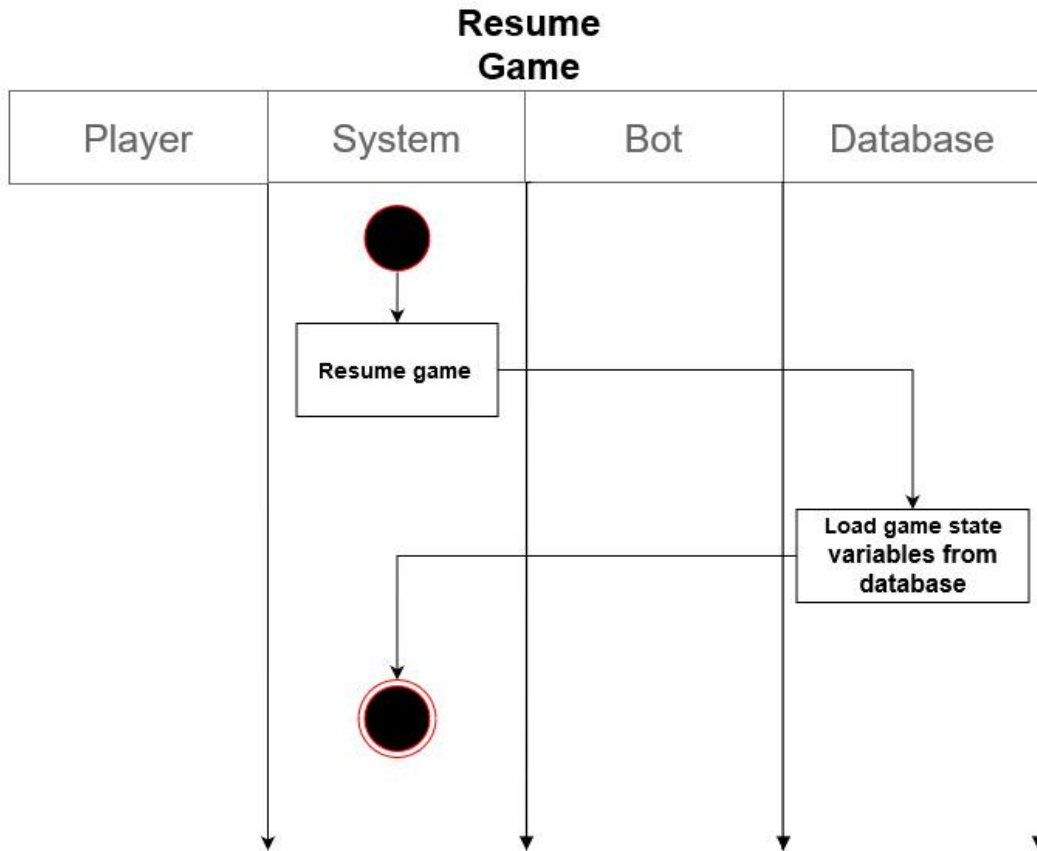
Name: Pause Game



Description: This is the swimlane diagram of the corresponding activity diagram ID: 22. It shows how different actors are involved with each activities of the module.

8.23 Swimlane Diagram ID: 23

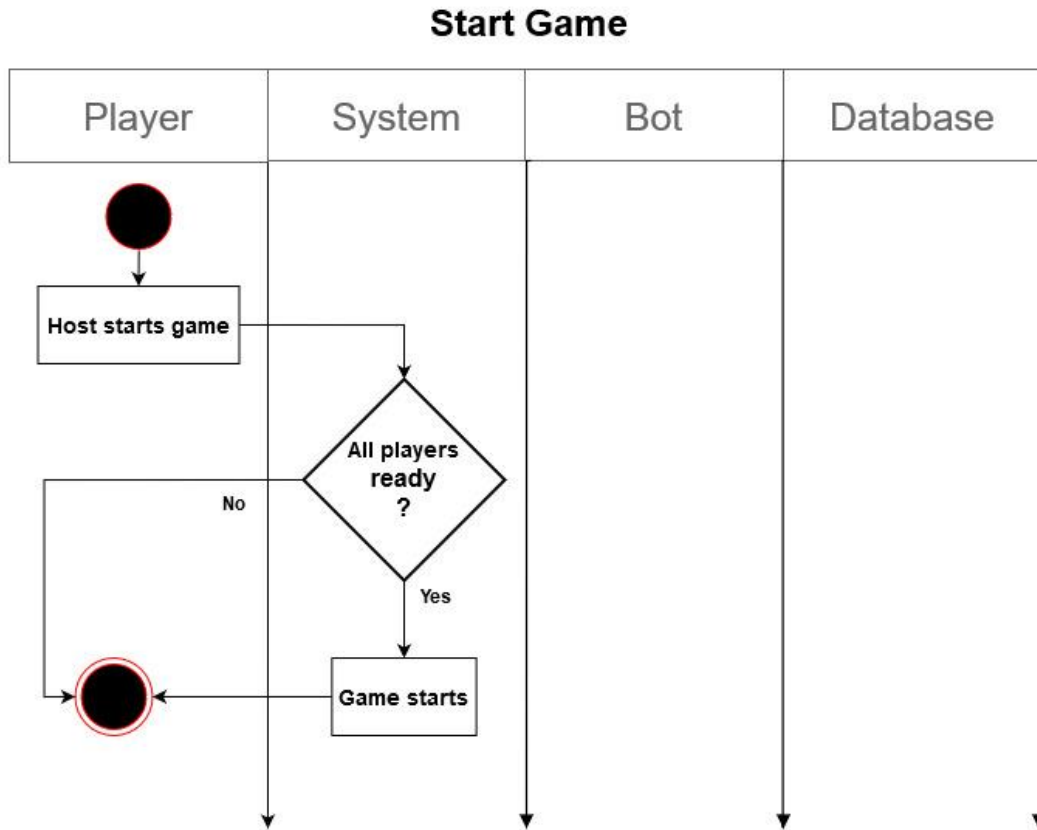
Name: Resume Game



Description: This is the swimlane diagram of the corresponding activity diagram ID: 23. It shows how different actors are involved with each activities of the module.

8.24 Swimlane Diagram ID: 24

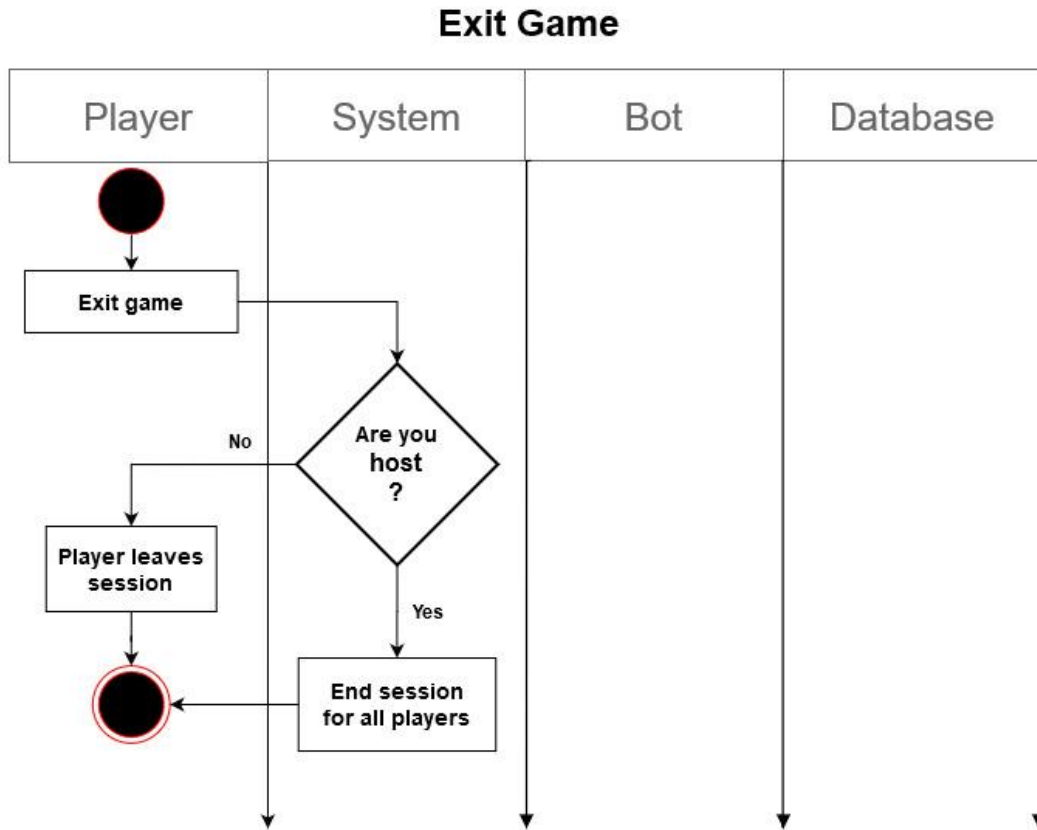
Name: Start Game



Description: This is the swimlane diagram of the corresponding activity diagram ID: 24. It shows how different actors are involved with each activities of the module.

8.25 Swimlane Diagram ID: 25

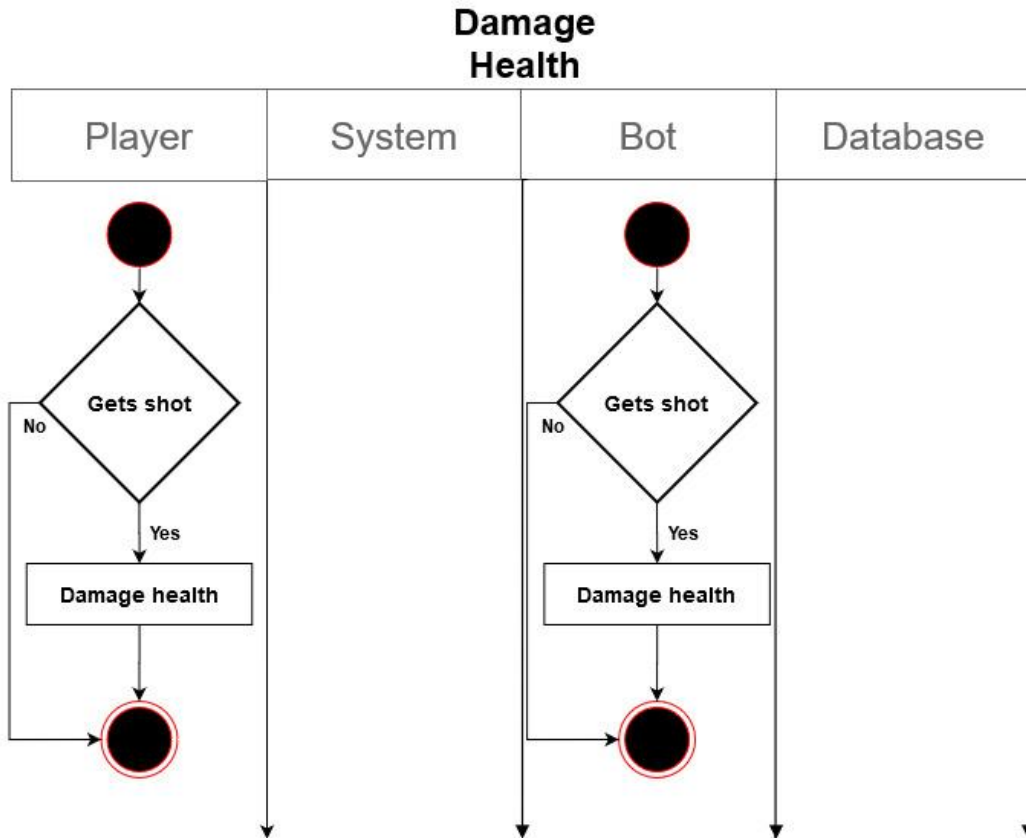
Name: Exit Game



Description: This is the swimlane diagram of the corresponding activity diagram ID: 25. It shows how different actors are involved with each activities of the module.

8.26 Swimlane Diagram ID: 26

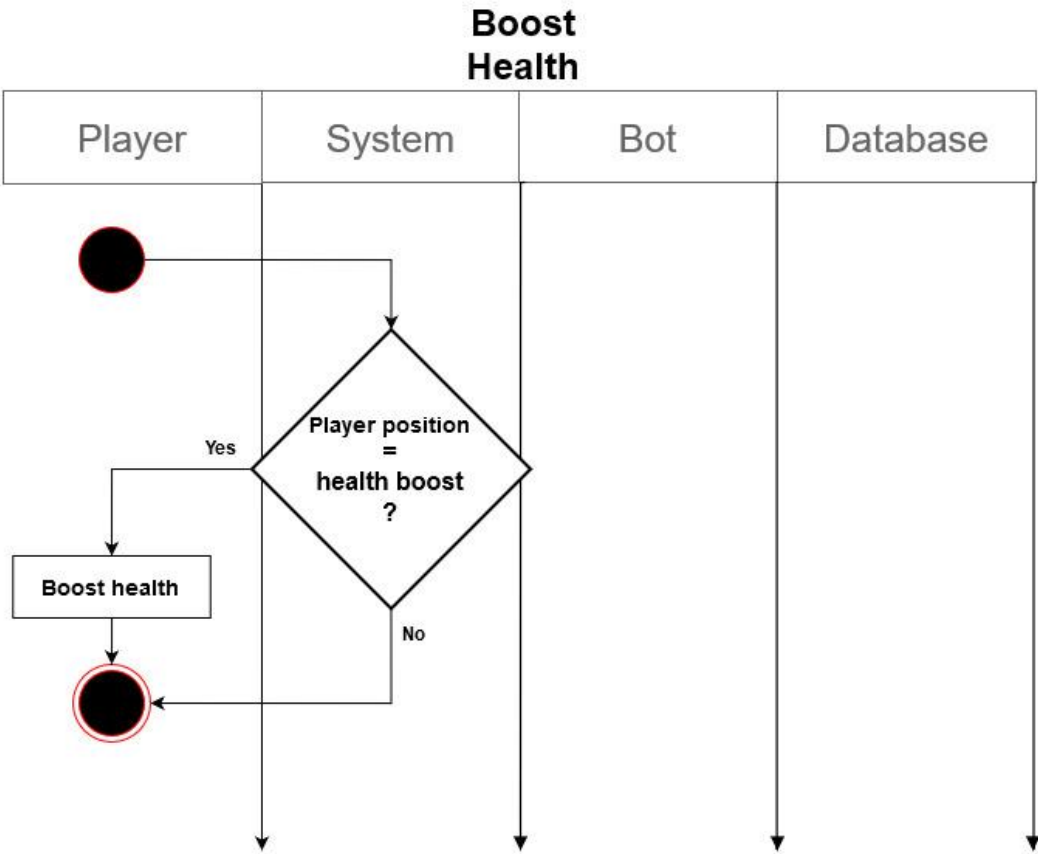
Name: Damage Health



Description: This is the swimlane diagram of the corresponding activity diagram ID: 26. It shows how different actors are involved with each activities of the module.

8.27 Swimlane Diagram ID: 27

Name: Boost Health



Description: This is the swimlane diagram of the corresponding activity diagram ID: 27. It shows how different actors are involved with each activities of the module.

9. Database modeling

There is not much use of database for this game. We are going to save the game state in the local database while pausing in the single player mode and we will also save the player profiles in the local database. Database used for this project is not a relational one, rather it is a database provided by the game engine to save data as a map object of attributes and values.

9.1 Data Object Identification

ID	Noun	Attributes
1	Player	48,49
2	Player location	X
3	Multiplayer	X
4	Game	X
5	Player game state	2,6,31,34,37
6	Number of kills	X
7	server	X
8	client	X
9	IP	X
10	match	X
11	Character	X
12	options	X
13	AI Bot	X
14	System	X
15	Mode	X
16	Graphics Scalability Settings	X
17	specifications	X
18	Control	X
19	shadow	X

20	quality	X
21	aliasing	X
22	FPS	X
23	Post processing	X
24	User Profile	48,49
25	database	X
26	Group	X
27	Text message	X
28	Lobby	X
29	communication	X
30	object	X
31	health	X
32	damage	X
33	aid	X
34	Ammo	X
35	reload	X
36	system	X
37	Weapon	X
38	shop	X
39	time	X
40	Advanced locomotion	X
41	Photorealistic environment	X
42	Frame rate	X
43	camera	X
44	distance	X
45	texture	X

46	Localization	X
47	Game Environment	X
48	Name	X
49	Avatar	X

9.2 Data Objects

Noun	Attributes
Player	48,49
Player Game State	2,6,31,34,37
User Profile	48,49

9.3 Analysis:

- Player and User Profile are identical. So, we take Player and discard User Profile.

9.4 Final Data Objects

Noun	Attributes
Player	48,49
Player Game State	2,6,31,34,37

9.5 Schema

Player Game State		
Attribute	type	Size
PlayerID	number	10
Player Location	varchar	30
Number of Kills	number	100
Health	number	100
Weapon	varchar	20
Ammo	number	500

Player		
Attribute	Type	Size
<u>PlayerID</u>	number	10
Name	varchar	50
Avatar	image	null

10. Class Based Modeling

10.1 General classification

General classification criteria:

Candidate classes are characterized in seven general classification. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Classification of nouns:

Player - 1, 4, 7
Map - 2, 6
Player location - 2, 6
Name - 2
Avatar - 2
Game State - 2, 7
Server - 2
Client - 2
IP - 2
Match - 3
Character - 2
Options - 2
AI Bot - 2, 4, 7
System - 2

Graphics Scalability Settings - 2, 3, 7
Specifications - 2
Shadow - 2
Aliasing – 2
System -
FPS - 2
Post processing - 2
Database - 1, 2, 7
Text message - 2
Lobby - 2, 6
Communication - 3
Object - 2
health system - 2, 7
reload system - 2, 3, 7
damage - 3
aid - 2
Ammo - 2
Weapon - 2, 4, 7
Shop - 6
Advanced locomotion - 3
Photorealistic environment - 2
Frame rate - 2
Camera - 2
Texture - 2
Game Environment – 2
Movement - 3, 7
Group - 5, 7

10.2 Selection criteria

The candidate classes are then selected as classes by six Selection Criteria. A candidate class generally becomes a class when it fulfills around three characteristics.

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Selection criteria of candidate classes:

Player - 1-5
 Map - 1-5
 Player location - 1, 4
 Game State - 1-5
 AI Bot - 1-5
 Graphics Scalability Settings - 1, 2, 3
 Database - 1, 2, 6
 Lobby - 1-5
 Health system - 2
 Reload system – 2
 Weapon - 1-5
 Movement - 1-5
 Group - 1-5

10.3 List of verbs

Verb	-	Description
Move	-	Move the player in all directions
Host	-	Host a session to play multiplayer
Join	-	Join an already hosted multiplayer session
Kick	-	Kick a player from a multiplayer session
Select	-	Select character
Play	-	Play a match
Change	-	Change graphics settings
Create	-	Create profile
Modify	-	Modify profile
Chat	-	Chat with other players in the same multiplayer session
Send	-	Send message
Receive	-	Receive message
Interact	-	Interact with interactive objects in the game environment
Shoot	-	Shoot the enemy
Reload	-	Reload weapon with ammo
Drop	-	Drop current weapon
Pick	-	Pick new weapon
Swap	-	Swap weapon
Walk	-	Walk in the game environment
Run	-	Run in the game environment
Jump	-	Jump in the game environment
Crouch	-	Crouch in the game environment
Combat	-	Combat in the game environment
Pause	-	Pause game
Resume	-	Resume game
Show	-	Show visuals
Spawn	-	Spawn players and bots
Trigger	-	Trigger bot behavior/response
Damage	-	Damage health
Boost	-	Boost health
Shake	-	Shake screen

10.4 Potential Classes

Class Name	Attributes	Methods
Player	<ul style="list-style-type: none"> - name - movement - avatarImage - characterName - characterImage - 3dAnimation - playerGameState - ip 	<ul style="list-style-type: none"> + changeName() + changeAvatar() + changeCharacter() + spawn() + isAlive() + combat() + getsShot() + createProfile()
Lobby/Menu	<ul style="list-style-type: none"> - listOfPlayers - listOfKickedPlayers - graphicsSettings - serverIP - graphicsSettings 	<ul style="list-style-type: none"> + sendMessage() + receiveMessage() + kickPlayer() + hostMatch() + joinMatch()
Group	<ul style="list-style-type: none"> - listOfPlayers - listOfKickedPlayers - graphicsSettings - serverIP 	<ul style="list-style-type: none"> + sendMessage() + receiveMessage() + kickPlayer() + hostMatch() + joinMatch()
Map/Visual	<ul style="list-style-type: none"> - 3dObjects 	<ul style="list-style-type: none"> + showObject() + showMap() + showAnimation() + shakeScreen()
Player Game State	<ul style="list-style-type: none"> - currentLocation - currentWeapon - health 	<ul style="list-style-type: none"> + pauseGame() + resumeGame() + saveGameState()

	<ul style="list-style-type: none"> - numberOfKills - ammo 	<ul style="list-style-type: none"> + damageHealth() + boostHealth()
AI Bot	<ul style="list-style-type: none"> - name - movement - avatarImage - characterName - characterImage - 3dAnimation - playerGameState 	<ul style="list-style-type: none"> + spawn() + isAlive() + onHunt() + patrolling()
Graphics Settings	<ul style="list-style-type: none"> - postProcessing - antiAliasing - shadowQuality - frameRate 	<ul style="list-style-type: none"> + changeAntiAliasing() + changeShadowQuality() + changeFrameRate + changePostProcessing()
Database	<ul style="list-style-type: none"> - database 	<ul style="list-style-type: none"> + store() + retrieve()
Weapon	<ul style="list-style-type: none"> - weaponName - damagePerBullet - highestAmmo - 3dAnimation 	<ul style="list-style-type: none"> + shoot() + ammoAvailable() + automaticReload() + manualReload() + pick() + equip() + swap() + drop()
Movement	<ul style="list-style-type: none"> - 3dAnimation 	<ul style="list-style-type: none"> + walk() + run() + goRight() + goLeft()

		+ goFront() + goBack() + crouch() + interact() + jump()
--	--	---

10.5 Analysis:

- Player and AI Bot have same attributes and some common functionality too. We can extend them from a new class BasicPlayer.
- Group and Lobby/Menu are identical. So we keep Lobby and discard Group.

10.6 Class Cards

BasicPlayer	
Attributes: - name - movement - avatarImage - characterName - characterImage - 3dAnimation - playerGameState - weapon	Methods: + spawn() + isAlive() + getsShot() + createProfile()

Responsibilities: <ul style="list-style-type: none"> • Storing player profile • Spawning • Player Animation 	Collaborators: <ul style="list-style-type: none"> • Player • AIBot • Visual • PlayerGameState • Weapon • Movement
--	---

Player (inherits BasicPlayer)	
Attributes: <p>- ip</p>	Methods: <p>+ changeName() + changeAvatar() + changeCharacter()</p>
Responsibilities: <ul style="list-style-type: none"> • Modify player profile • Player ip credentials 	Collaborators: <ul style="list-style-type: none"> • Lobby • BasicPlayer • Database

AIBot (inherits BasicPlayer)	
Attributes:	Methods: + onHunt() + patrolling()
Responsibilities: <ul style="list-style-type: none">• Patrolling Logic• Hunting Logic• Shooting Logic	Collaborators: <ul style="list-style-type: none">• BasicPlayer

Lobby	
Attributes: - listOfPlayers - listOfKickedPlayers - graphicsSettings - serverIP - graphicsSettings	Methods: + sendMessage() + receiveMessage() + kickPlayer() + hostMatch() + joinMatch()
Responsibilities: <ul style="list-style-type: none">• Chatting• Kicking player by server• Hosting match• Joining Match	Collaborators: <ul style="list-style-type: none">• Player• GraphicsSettings• Visual

Visual	
Attributes: <ul style="list-style-type: none">- 3dObjects	Methods: <ul style="list-style-type: none">+ showObject()+ showMap()+ showAnimation()
Responsibilities: <ul style="list-style-type: none">• Show all the visuals including animations	Collaborators: <ul style="list-style-type: none">• BasicPlayer• Lobby• Weapon• Movement

PlayerGameState	
Attributes: <ul style="list-style-type: none">- currentLocation- currentWeapon- health- numberOfKills- ammo	Methods: <ul style="list-style-type: none">+ pauseGame()+ resumeGame()+ saveGameState()+ damageHealth()+ boostHealth()
Responsibilities: <ul style="list-style-type: none">• Store player data during match	Collaborators: <ul style="list-style-type: none">• BasicPlayer• Database

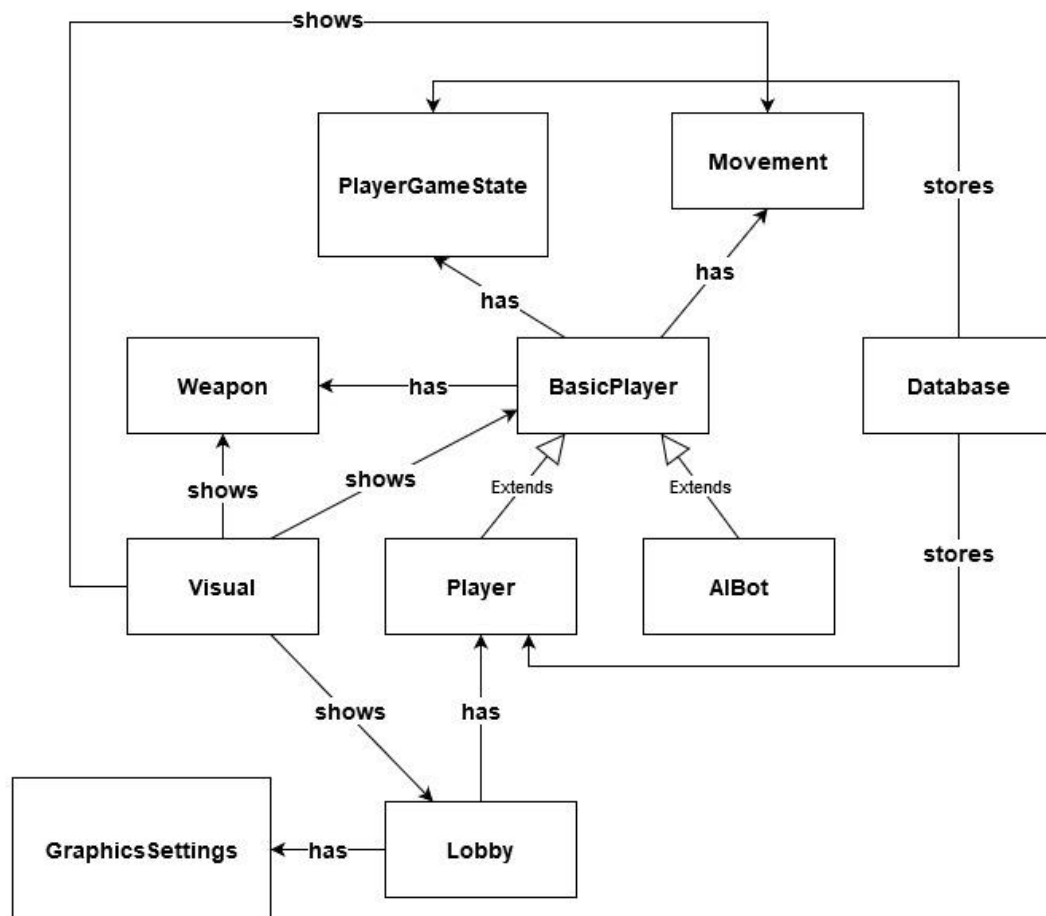
GraphicsSettings	
Attributes: <ul style="list-style-type: none">- postProcessing- antiAliasing- shadowQuality- frameRate	Methods: <ul style="list-style-type: none">+ changeAntiAliasing()+ changeShadowQuality()+ changeFrameRate+ changePostProcessing()
Responsibilities: <ul style="list-style-type: none">• Modify all graphics scalability settings	Collaborators: <ul style="list-style-type: none">• Lobby

Database	
Attributes: <ul style="list-style-type: none">- database	Methods: <ul style="list-style-type: none">+ store()+ retrieve()
Responsibilities: <ul style="list-style-type: none">• Storing profile and game state data• Retrieving stored data	Collaborator: <ul style="list-style-type: none">• Player• PlayerGameState

Weapon	
<p>Attributes:</p> <ul style="list-style-type: none">- weaponName- damagePerBullet- highestAmmo- 3dAnimation	<p>Methods:</p> <ul style="list-style-type: none">+ shoot()+ ammoAvailable()+ automaticReload()+ manualReload()+ pick()+ equip()+ swap()+ drop()
<p>Responsibilities:</p> <ul style="list-style-type: none">• Reloading• Shooting• Pick, swap, equip, drop weapon• Weapon info	<p>Collaborators:</p> <ul style="list-style-type: none">• BasicPlayer• Visual

Movement	
<p>Attributes:</p> <ul style="list-style-type: none">- 3dAnimation	<p>Methods:</p> <ul style="list-style-type: none">+ walk()+ run()+ goRight()+ goLeft()+ goFront()+ goBack()+ crouch()+ jump()
<p>Responsibilities:</p> <ul style="list-style-type: none">• All movements	<p>Collaborators:</p> <ul style="list-style-type: none">• BasicPlayer• Visual

10.7 CRC Diagram:



11. Behavioral Modeling

11.1 Event Identification Table:

No.	Event	Initiator	Collaborator
1.	Giving Direction	Movement	Visual
2.	Crouching	Movement	Visual
3.	Walking	Movement	Visual
4.	Jumping	Movement	Visual
5.	Running	Movement	Visual

6.	Shooting	Weapon	Visual
7.	Aiming	Weapon	Visual
8.	Picking Weapon	Weapon	
9.	Snatching Weapon	Weapon	
10	Dropping Weapon	Weapon	
11.	Getting Shot	BasicPlayer	PlayerGameState
12.	Health Damage	PlayerGameState	BasicPlayer

13.	Health Boost	PlayerGameState	Movement
14	Create Profile	BasicPlayer	Database Player
15.	Create Session	Lobby	
16.	Join Session	Lobby	
17.	View Game World	Visual	
18.	Change Graphics Settings	GraphicsSettings	Visual
19.	AI Bot Response	AlBot	Visual

20.	Send Message	Lobby	
21.	Receive Message	Lobby	
22.	Interaction with In-game Objects	Visual	
23.	Interactive Object Response	Visual	
24.	Kick Player	Lobby	Player
25	Character Selection	Lobby	Player
26.	Modify Profile	Player	Database

27.	Store in Database	Database	BasicPlayer Player PlayerGameState
28.	Start Match	Lobby	Player
29.	End Match	Lobby	
30.	Player Ready	Player	Lobby
31.	Manually Reload Weapon	Weapon	
32.	Automatically Reload Weapon	Weapon	
33.	Show Static visual	Visual	

34.	Show Dynamic Visual	Visual	Movement BasicPlayer Weapon Lobby AlBot
35.	Show Animation	Visual	Weapon Movement AlBot
36.	Pause Game	PlayerGameState	Database
37.	Resume Game	PlayerGameState	Database
38.	Scope Weapon	Weapon	
39.	Pick Ammo	Weapon	
40.	Unscope Weapon	Weapon	

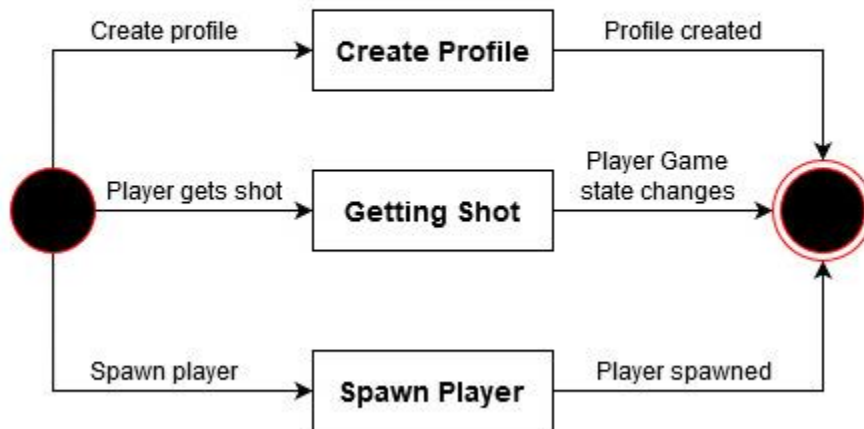
41.	Leave Game	Player	Lobby
42.	Retrieve from Database	Database	Player PlayerGameState
43.	Spawn Player	BasicPlayer	Visual

11.2 State transition diagrams

11.2.1 Class name: BasicPlayer

ID: C_01

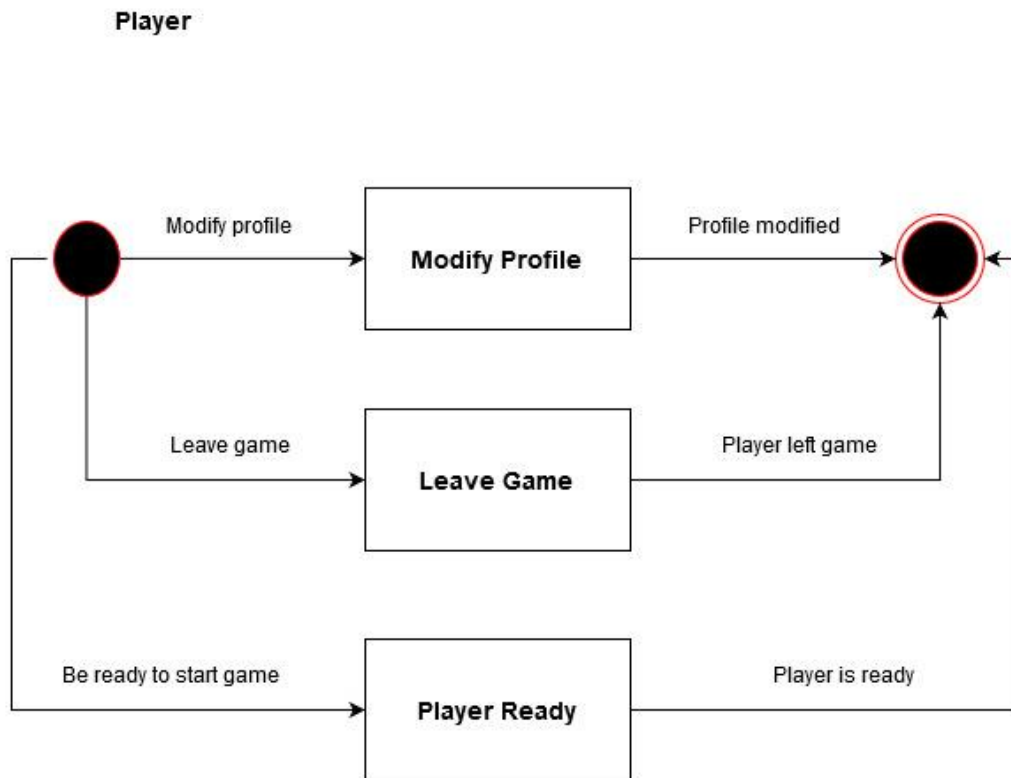
BasicPlayer



Description: The state diagram of BasicPlayer shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to profile creation, spawning players and shot detection.

11.2.2 Class name: Player

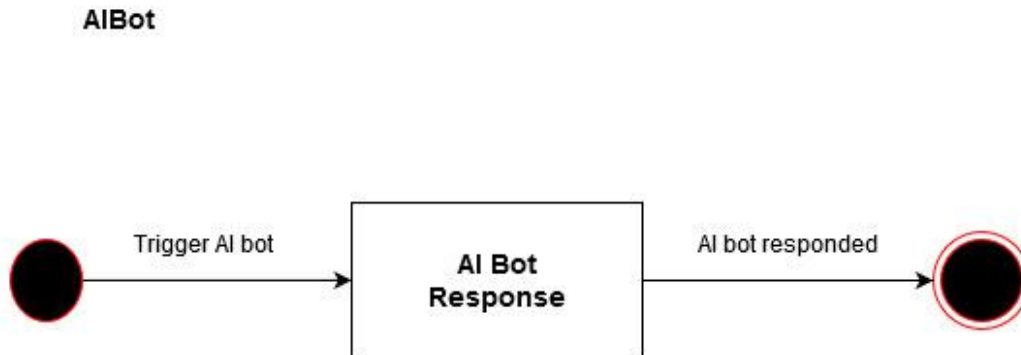
ID: C_02



Description: The state diagram of Player shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to profile modification and leaving an ongoing game.

11.2.3 Class name: AIBot

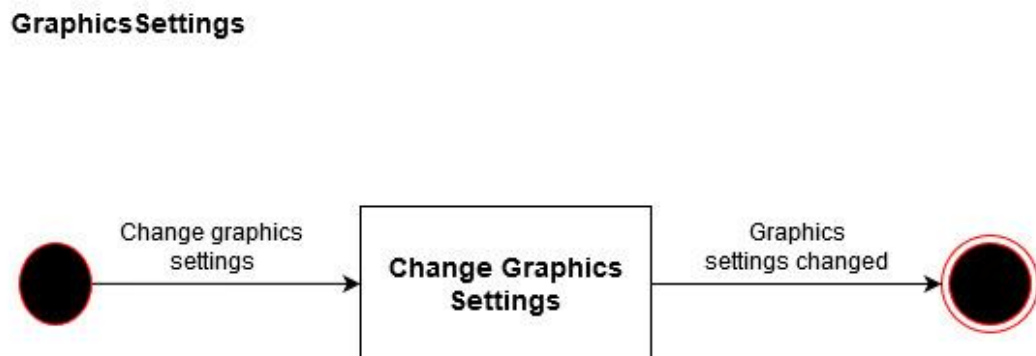
ID: C_03



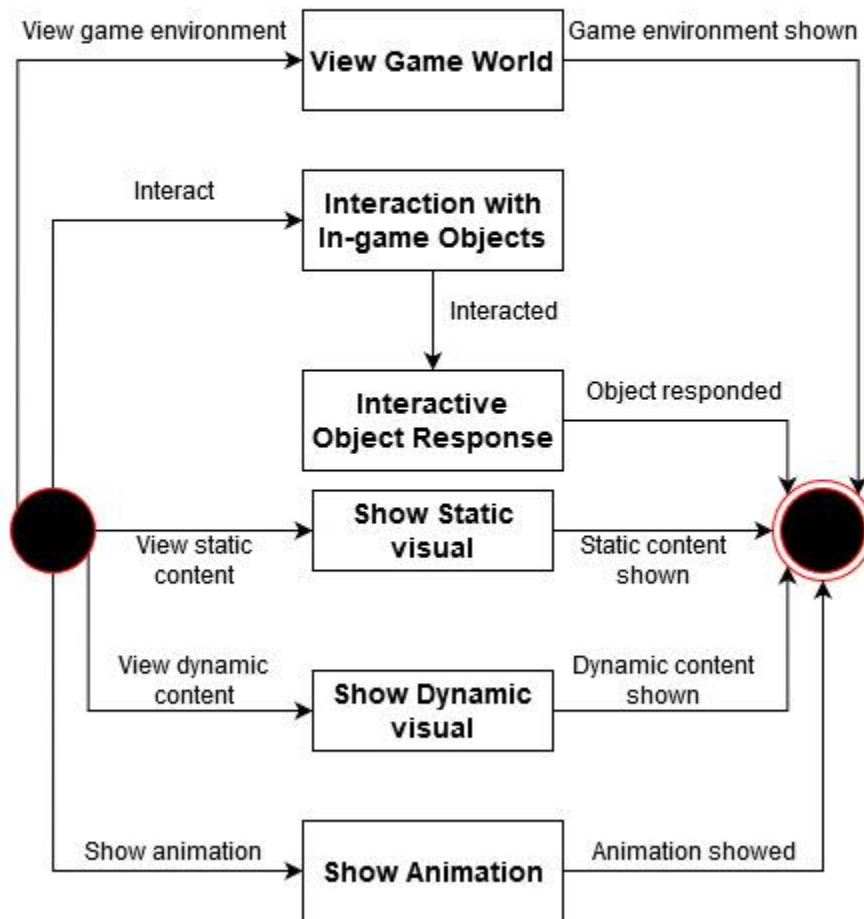
Description: The state diagram of AIBot shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another.

11.2.4 Class name: GraphicsSettings

ID: C_04



Description: The state diagram of GraphicsSettings shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another.

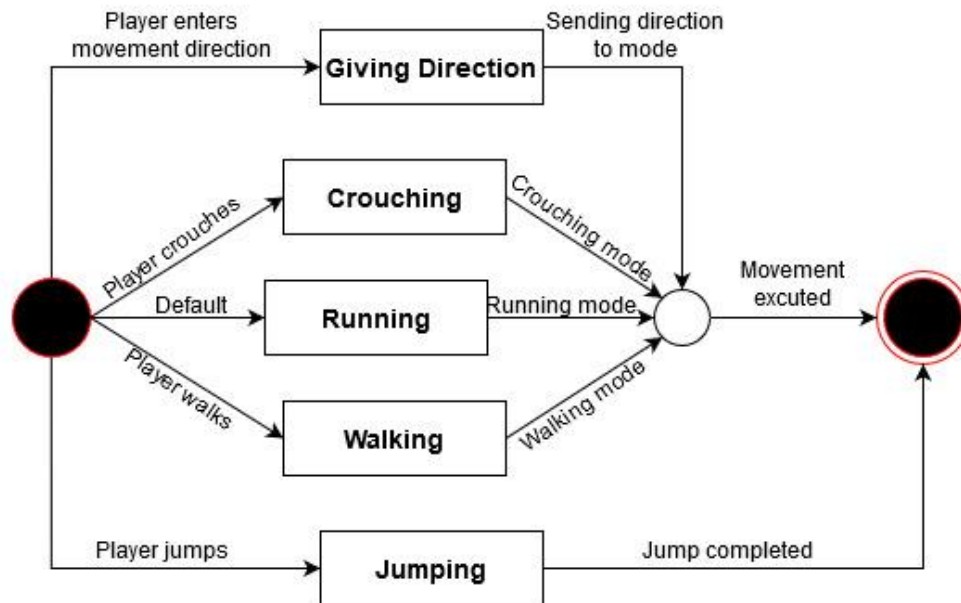
11.2.5 Class name: Visual**ID: C_05****Visual**

Description: The state diagram of Visual shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to interaction with game world objects and viewing of the game world through static and dynamic visuals.

11.2.6 Class name: Movement

ID: C_06

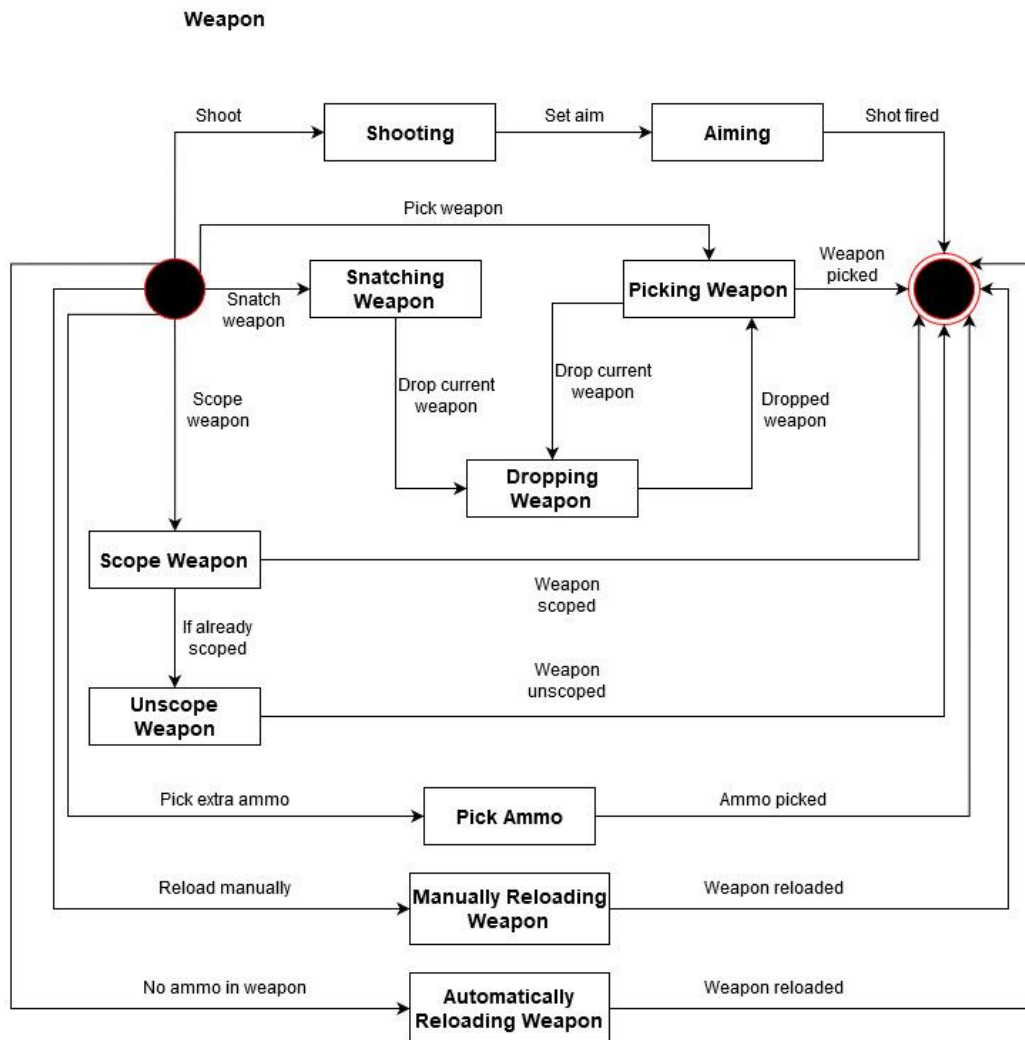
Movement



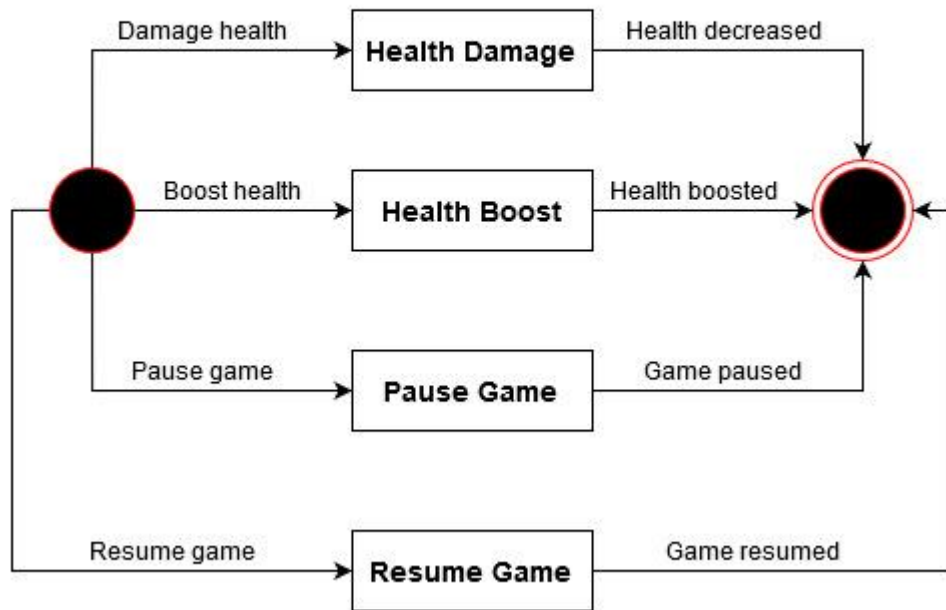
Description: The state diagram of Movement shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to movement directions, walking, running, jumping and crouching.

11.2.7 Class name: Weapon

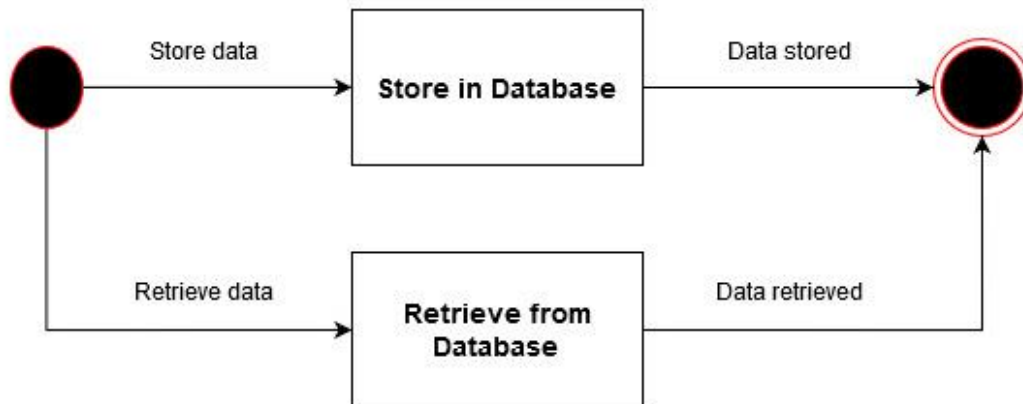
ID: C_07



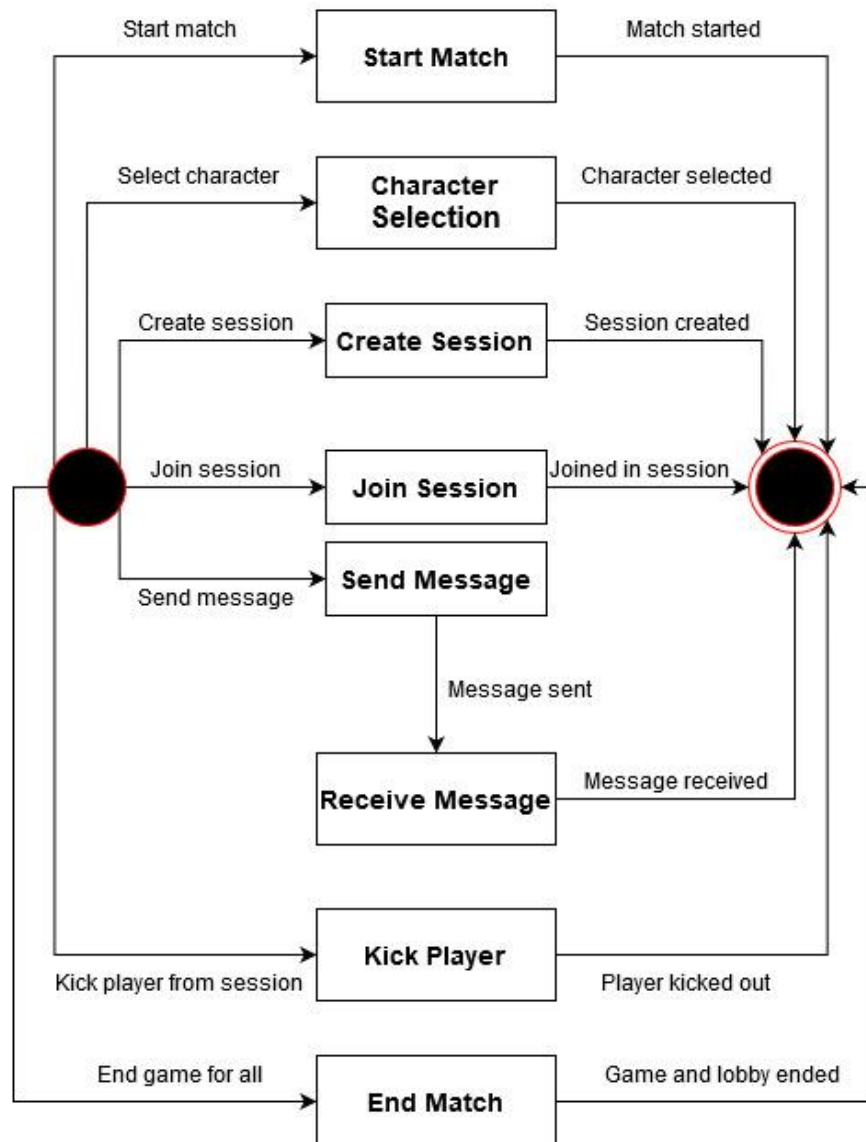
Description: The state diagram of Weapon shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to picking, snatching, dropping, reloading and scoping weapons.

11.2.8 Class name: PlayerGameState**ID: C_08****PlayerGame State**

Description: The state diagram of PlayerGameState shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to health damage and boost, pausing and resuming game.

11.2.9 Class name: Database**ID:** C_09**Database**

Description: The state diagram of Database shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to storing and retrieving data.

11.2.10 Class name: Lobby**ID: C_10****Lobby**

Description: The state diagram of Lobby shows the number of different states that the class can take over its lifetime and the events that cause it to transform from one state to another. The states are in this case, mainly related to sessions, chatting and kicking players.

11.3 Sequence diagram

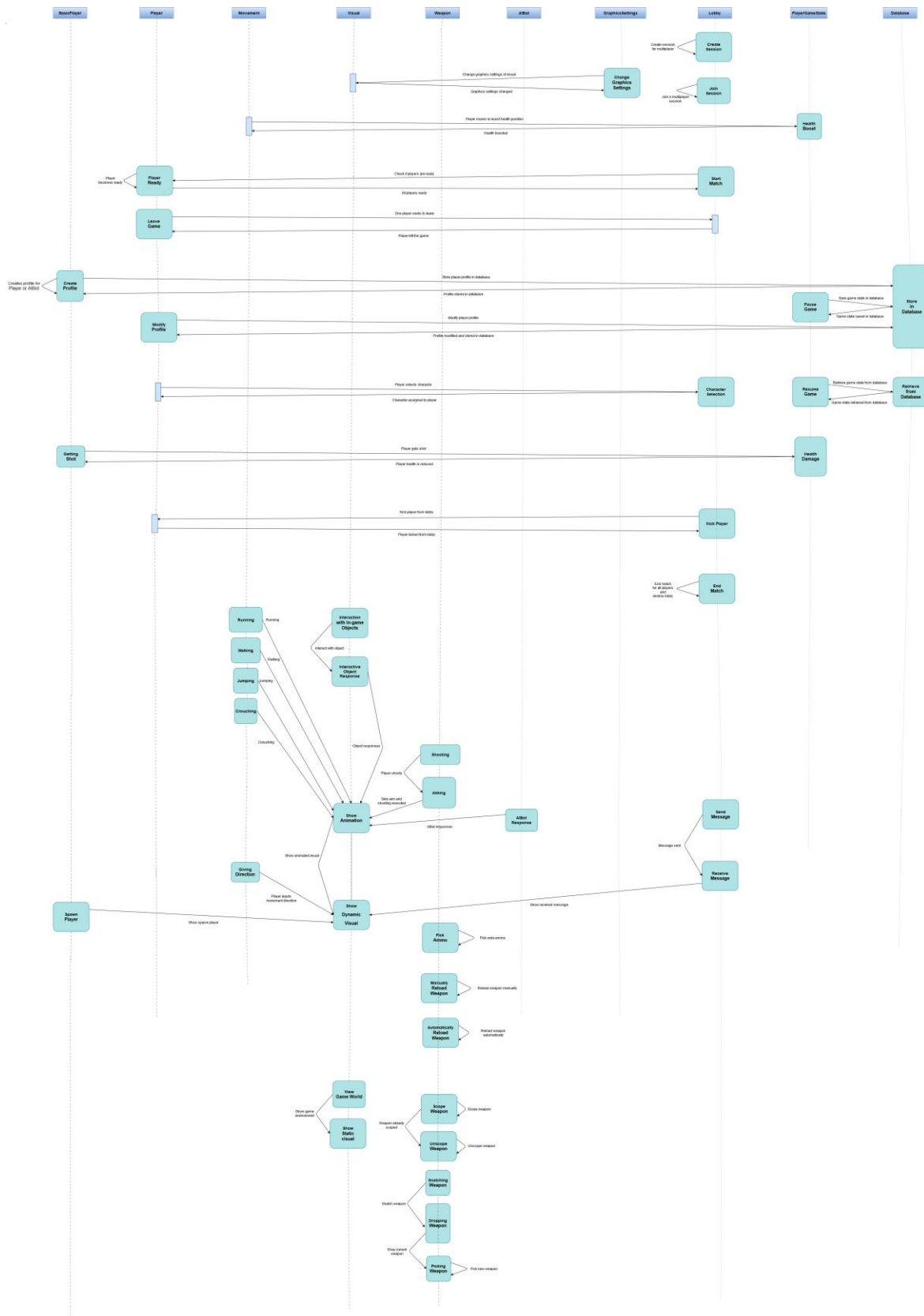


Figure: Sequence Diagram. (Please refer to the separate sequence diagram for this)

12. Appendix

12.1 Glossary

1. Moitrigor	-	Name of the game
2. Scoping	-	Zooming in
3. Lobby	-	Menu
4. Reload	-	Add ammo in the weapon