



Istanbul Kemerburgaz University

Department of Computer Engineering

CE370 SOFTWARE ENGINEERING PROJECT

Space Wars Game

DESIGN REPORT

Abdurrahman HUT

Course Instructor: Assoc. Prof. Mert ÖZKAYA

Table of Content

1. Introduction.....	3
a. Purpose of the document.....	3
b. Purpose of the system.....	3
2. Detailed Design Class Diagram	5
3. Dynamic Models.....	6
a. State Diagrams.....	6
b. Activity Diagram.....	8
c. Sequence Diagrams.....	10
4. Software Architecture.....	17
a. UML Package Diagram.....	17
b. Multi-tier Design Pattern.....	18
5. References.....	21

1. Introduction

a. Purpose of the document

Design report is the route map of any software system to implement it. It gives in depth information about software system's structure. Design report shows system's detailed class diagram. It is more detailed compared to the class diagram in Analysis Report. Also, it shows the state, activity and sequence diagrams which shows systems behaviors in different scenarios. Also, software system's design pattern and package diagram are given in the design report.

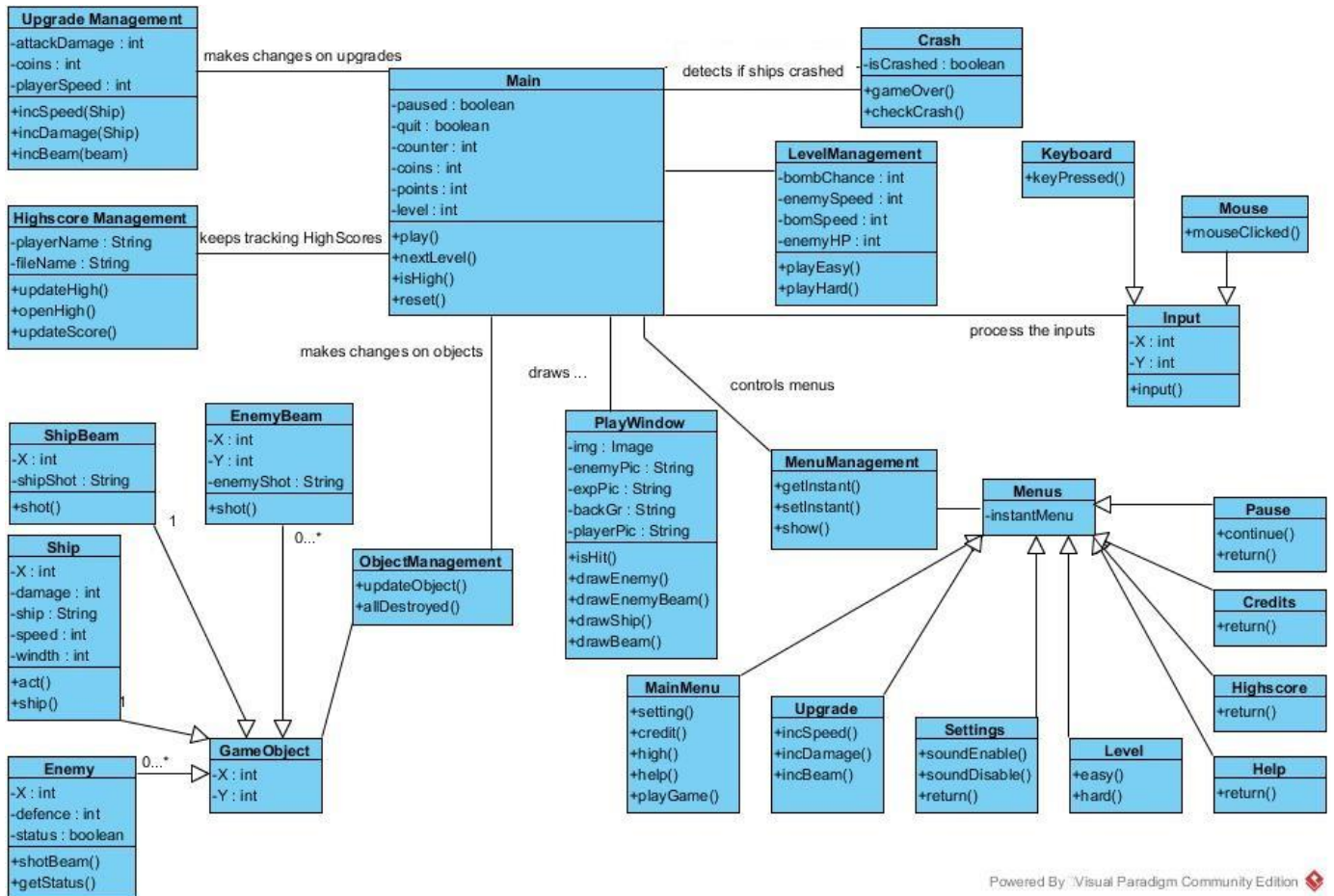
b. Purpose of the system

The users of this system cannot be limited since it is a game. There are different types of users for this system. However, the main possible users of this system are the children. The game has lots of features of visual effects which attract children mostly and that's why children are willingness to play the game more likely. For the other users of the system, people are bored sometimes after their daily works and want to escape from those duties with playing games. They give their selves

some time to have fun. All in all, children are most probably the main domain but that is not for all cases.

The new game provides lots of improves for its users compared to the old one. New game offers two different difficulty level to users which are easy mode and hard mode. In easy mode, there are less enemy spaceships and enemies' beams just move vertically, in hard mode there are more enemy spaceships and the beams of enemies follows the position of the player's spaceship. Addition to the old system, the bricks in front of the player are removed to make game more attractive. Also, enemies are separated from each other to increase the fire density for player to escape. In the new game player is able to upgrade his/her spaceship, to upgrade the ship player uses coins that he/she gained from after every successful level. Player can upgrade the spaceship's attack damage, speed as well as the firing speed.

2. Detailed Design Class Diagrams



Powered By Visual Paradigm Community Edition

Figure for Detailed Class Diagram

3. Dynamic Models

a. State Diagrams

States of the Player

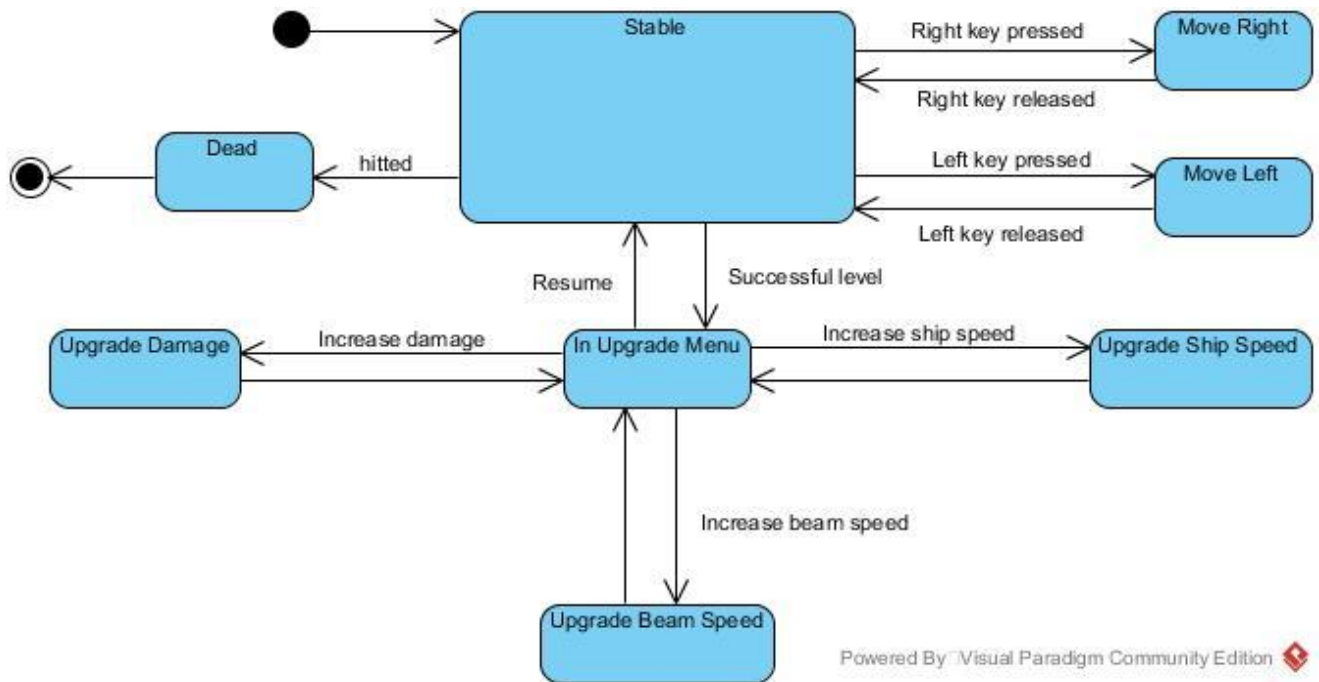


Figure for player's states

This state diagram shows the behavior of the player during playing. If the user uses keys that he/she can use, the player will move horizontally on the screen. The player is the opportunity to pass through the screen borders when player reached the left end of the screen, player will automatically pass to the right end. If there is no input came from user, it is initially at the state of "stable". After player succeed in a level, player is in inactive state named "unable". Player can purchase upgrade or

continue to the next level without spending his/her coins. If the player gets hit, game will be over.

State of the Enemy

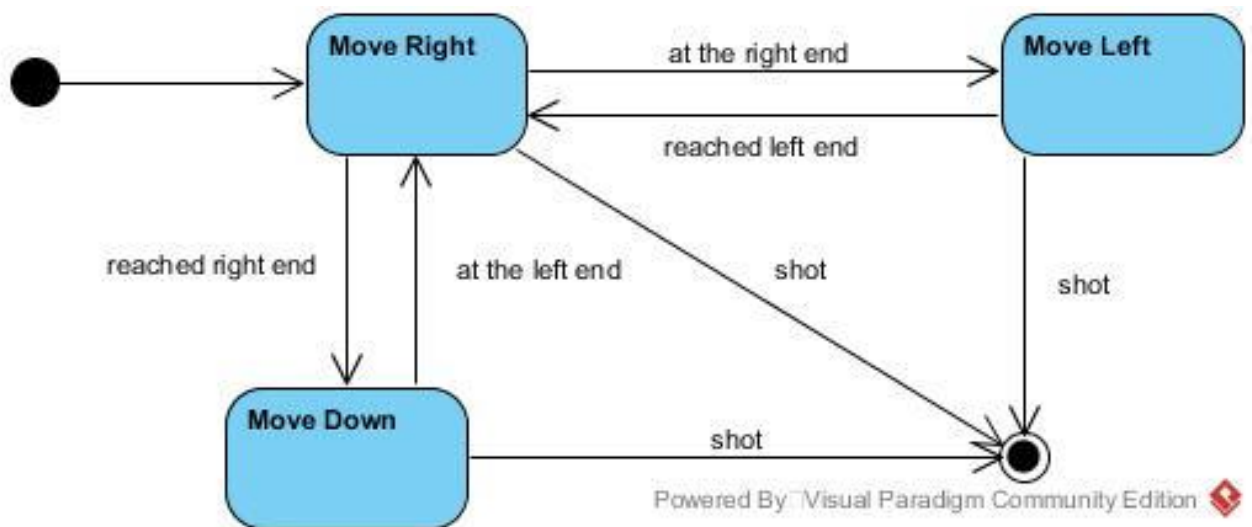


Figure for enemy's states

This figure shows the states of an enemy. Enemies keep moving to the right until they reach the right end of the screen. When they reach the right end, they will be vertically moved one line closer to the player's ship. Then, they continue to move to the left until they reach the left end of the screen. When they reach the left end, the same thing will be happened. Enemies can be shot during any of the states. This case is the end of the life of an enemy.

b. Activity Diagram

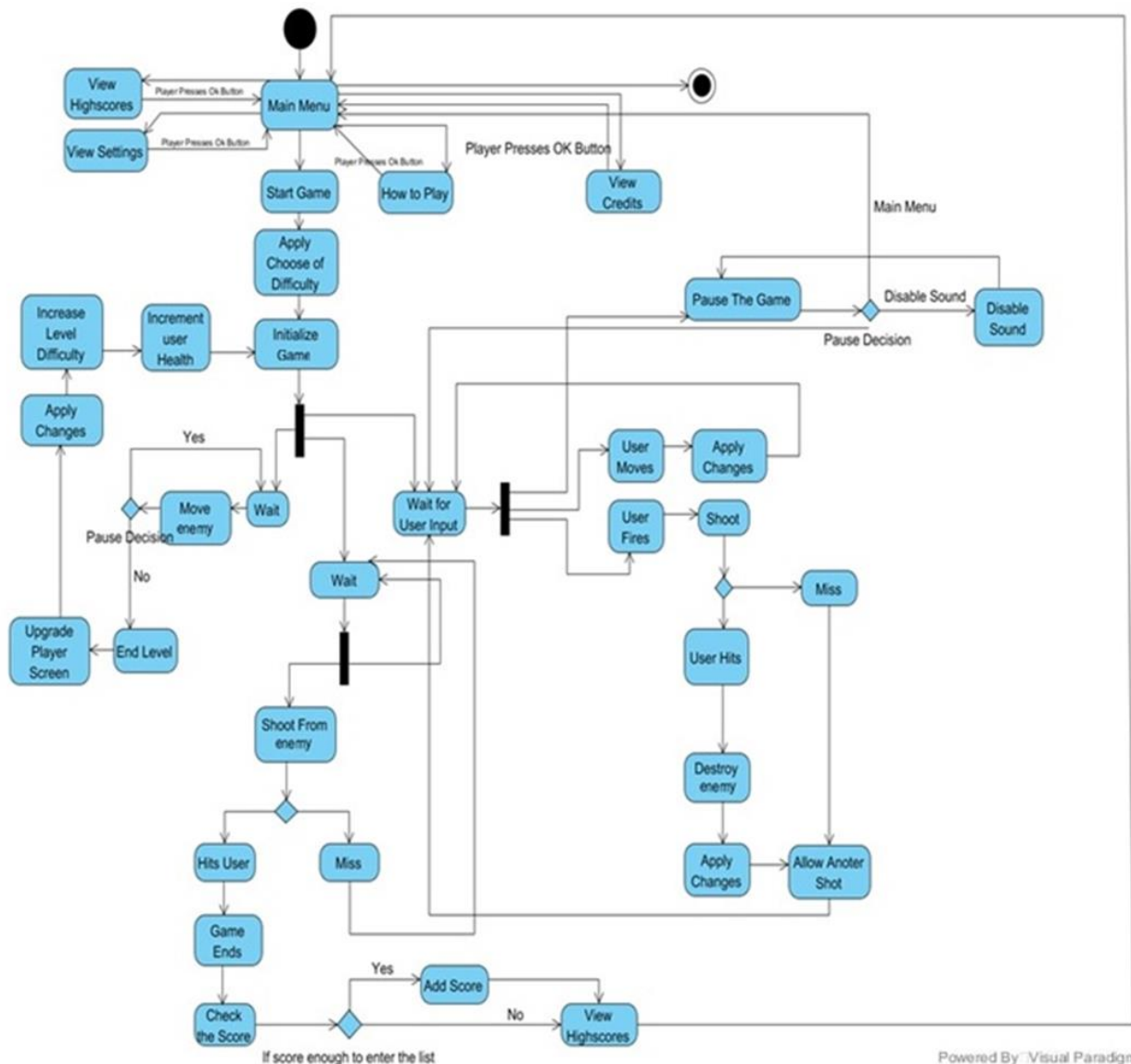


Figure for Activity Diagram

This diagram is showing the main flow of the game. Game is loaded from the main menu and game gives some options to the user such as viewing high scores, settings, credits or how to play. The system gives the opportunity to go back to the main menu when user is in the any of the option menus. The system is able to terminate with the quit option or start a new game. If a new game is selected, the system will ask to the

user for the difficulty level and it will apply the changes. The game is loaded from the first level and it is waiting for the input from the user. If user uses the arrow keys, the system allows the user to move horizontally along the screen. If the user chooses the space key the system fires a laser beam from player spaceship. If the beam hits an enemy, the system destroys that enemy but the laser beam can also miss the alien. System always allows player to shoot another shot if the game has not finished yet. Also, the game waits for random seconds according to the difficulty of that level and moves the enemy ships towards to player's ship and checks if the enemies have reached the end of the screen that causes the user to lose. Moreover, the system makes enemies to shoot beams randomly. If a bomb hits the player's spaceship, the player loses the game. Another possibility is pressing the esc key. If user presses esc key, the pause menu is displayed, system can terminate from this screen by pressing the esc button again.

At the end of the game, the system checks the score. If the score is greater than any high scores that recorded before, the system edits the scores and then shows it to the player. If the score is not able to get into the high score list, the system just shows the current high scores.

c. Sequence Diagrams

Scenario: Changing Settings

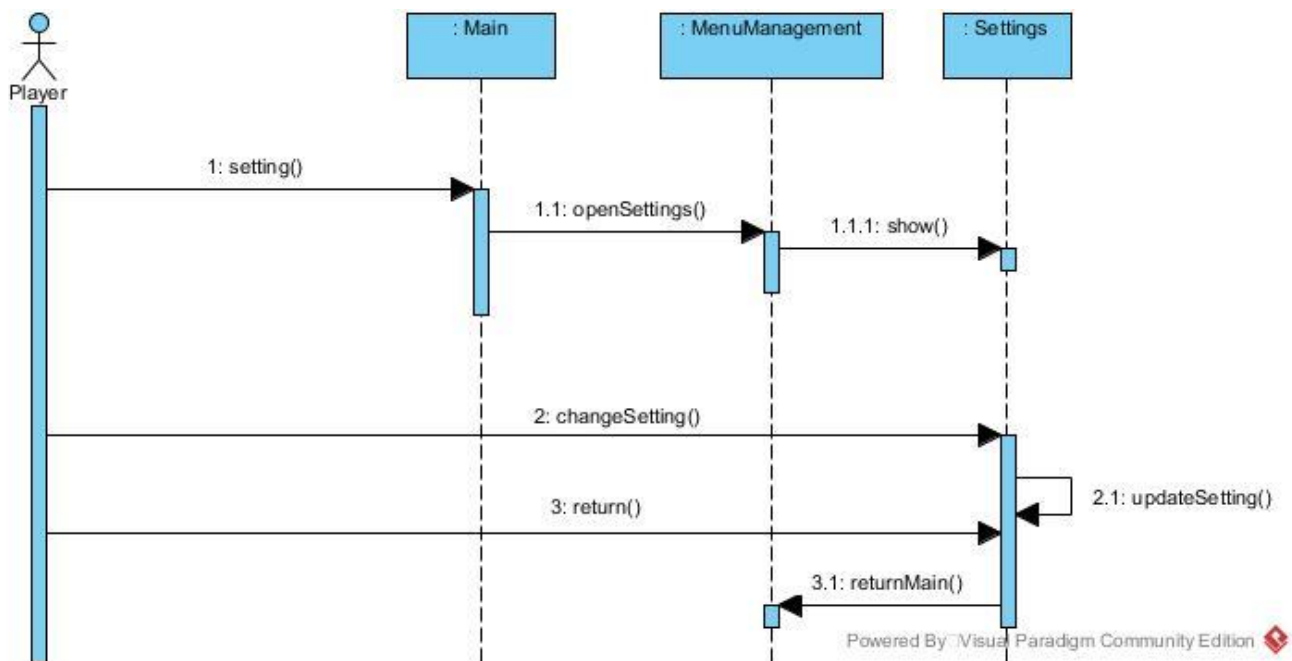


Figure for Sequence Diagram; Changing Settings

- Settings can be changed from the main menu or from the pause menu.
- Settings menu lets user to disable/enable the sound.
- Settings are updated as the user changed.
- User wants to return to main menu

Scenario: Getting Help

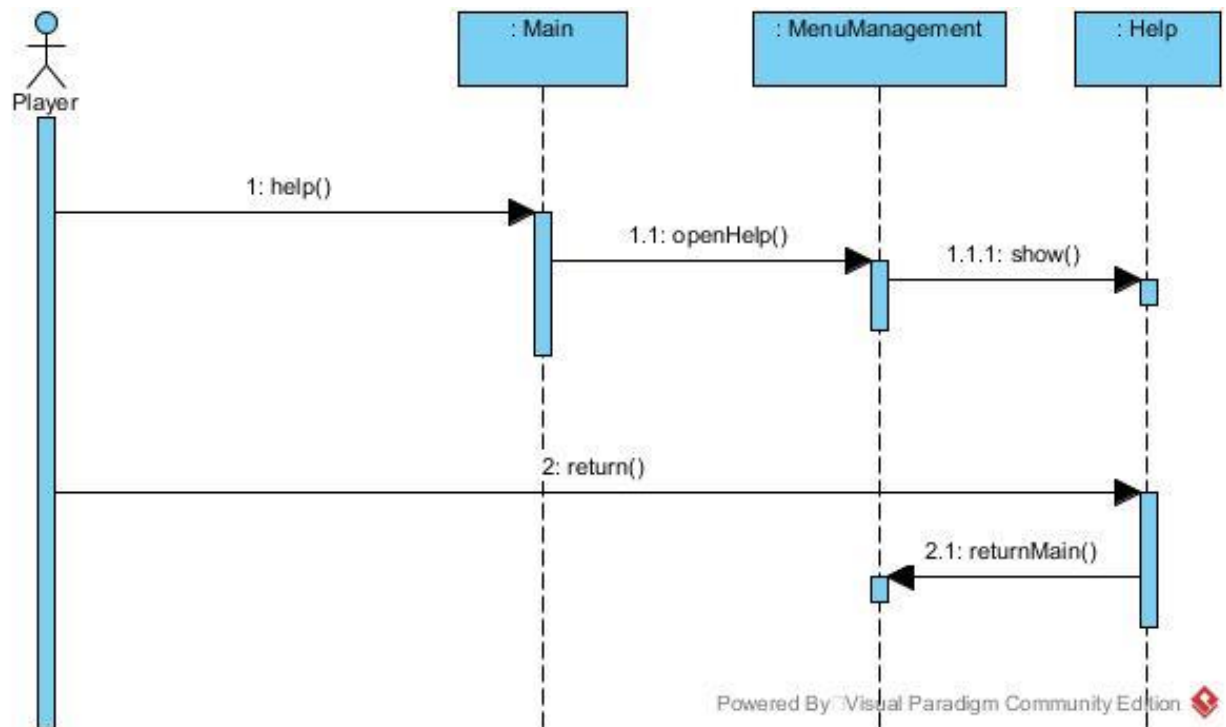


Figure for Sequence Diagram; Getting Help

- User chooses Help from Main Menu, to get information about game controllers and rules.
- Help is shown on the window
- User wants to return to main menu

Scenario: Viewing High Scores

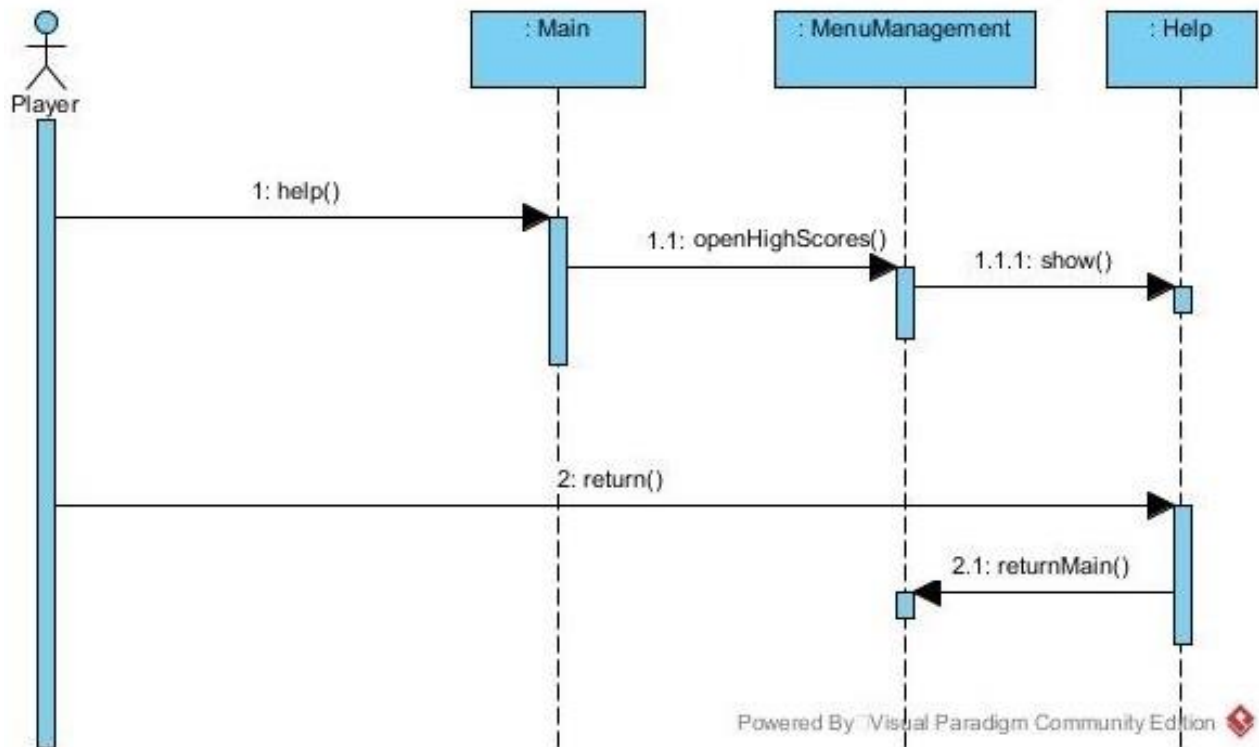


Figure for Sequence Diagram; Viewing High Scores

- To see the top ten scores, the user chooses “View High Scores” option.
- The system shows the highest ten scores.
- If the user wants to return back to the main menu, he/she chooses the “Back” option of the High Scores screen and exits from the high scores page.

Scenario: Viewing Credits

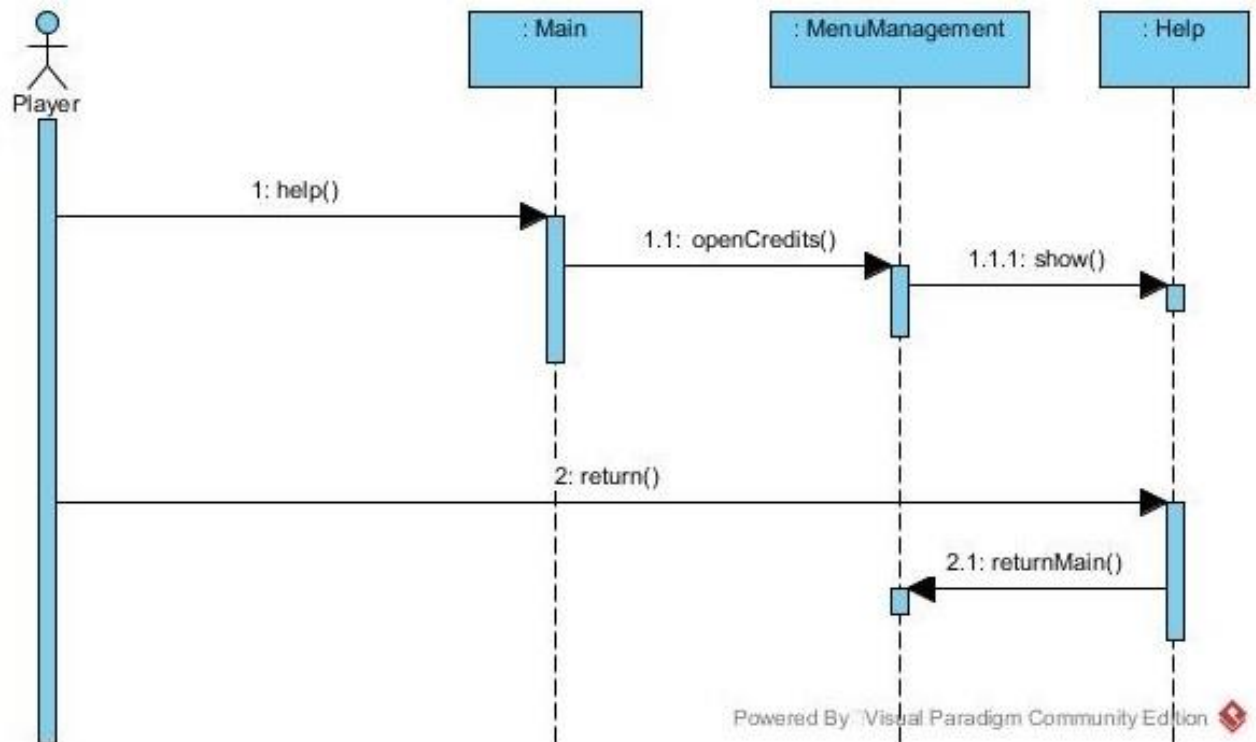


Figure for Sequence Diagram; Viewing Credits

- Player wants to show credits. View credits option is chosen from the main menu. The system displays the credits.
- Back option is used for going back to main menu from the view credits option.

Scenario: Upgrading User's Spaceship

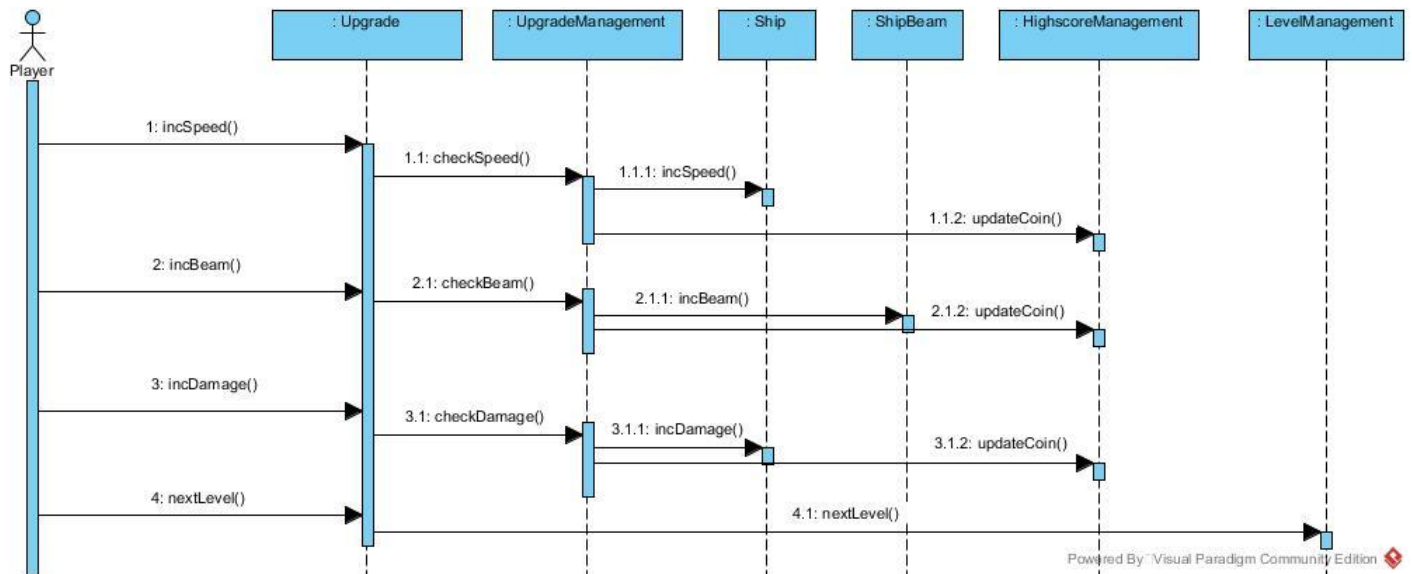


Figure for Sequence Diagram; Upgrading User's Spaceship

-At the end of the successful level, the system gives two options.

Player can continue game without spending any coins or player can upgrade the spaceship by spending earned coins.

-If coins are spent, system activates the remaining coin amount.

Scenario: Choosing Game Difficulty

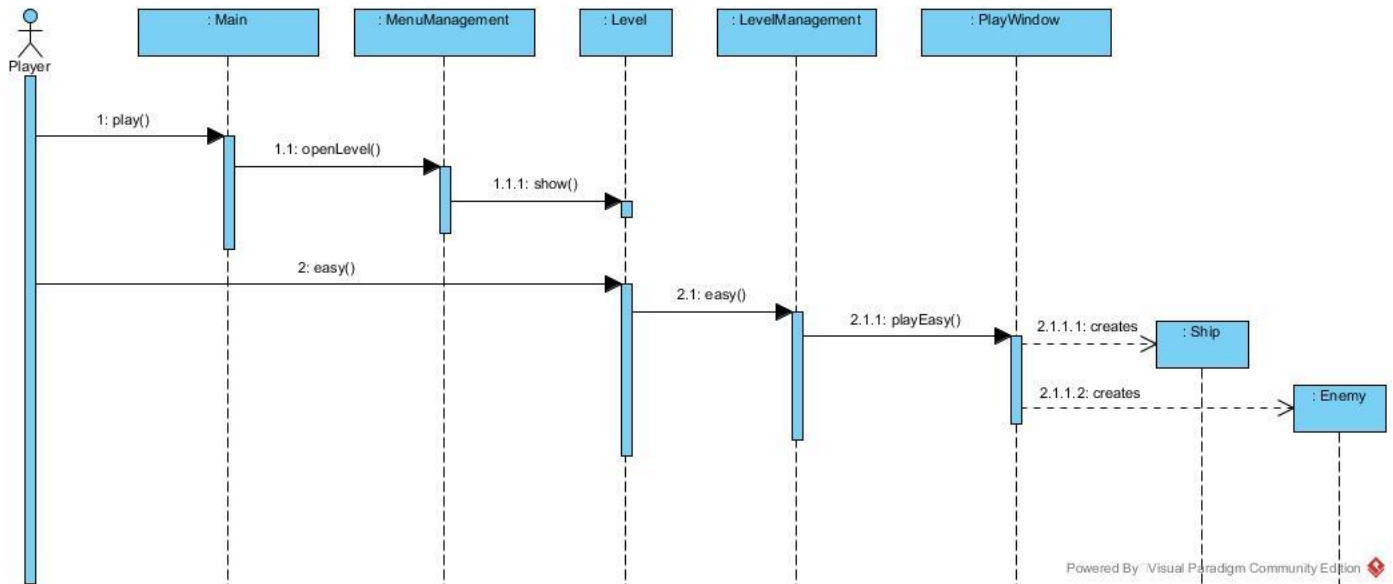


Figure for Sequence Diagram; Choosing Game Difficulty

- The user selects the play game from the main menu.
- The system asks user the game difficulty level either hard or easy.
- The game is loaded as user wanted difficulty.

Scenario: Shooting to an enemy

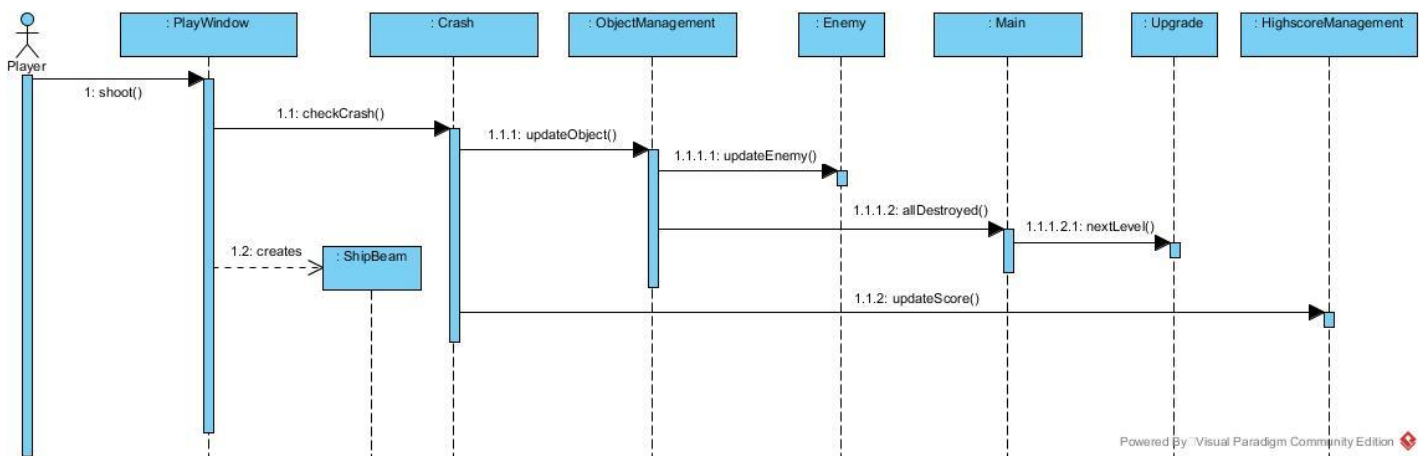


Figure for Sequence Diagram; Shooting an Enemy

- User tries to kill the enemies by shooting.
- When the user succeeds, status of enemy is updated.
- User earns points.
- When all of the enemies are killed, upgrade menu is displayed.

4. Software Architecture

For this software system, the multi-tier design pattern is the best architecture since a game has mainly 3 layers which are: Interface layer, processing layer and data layer.

a. UML Package Diagram

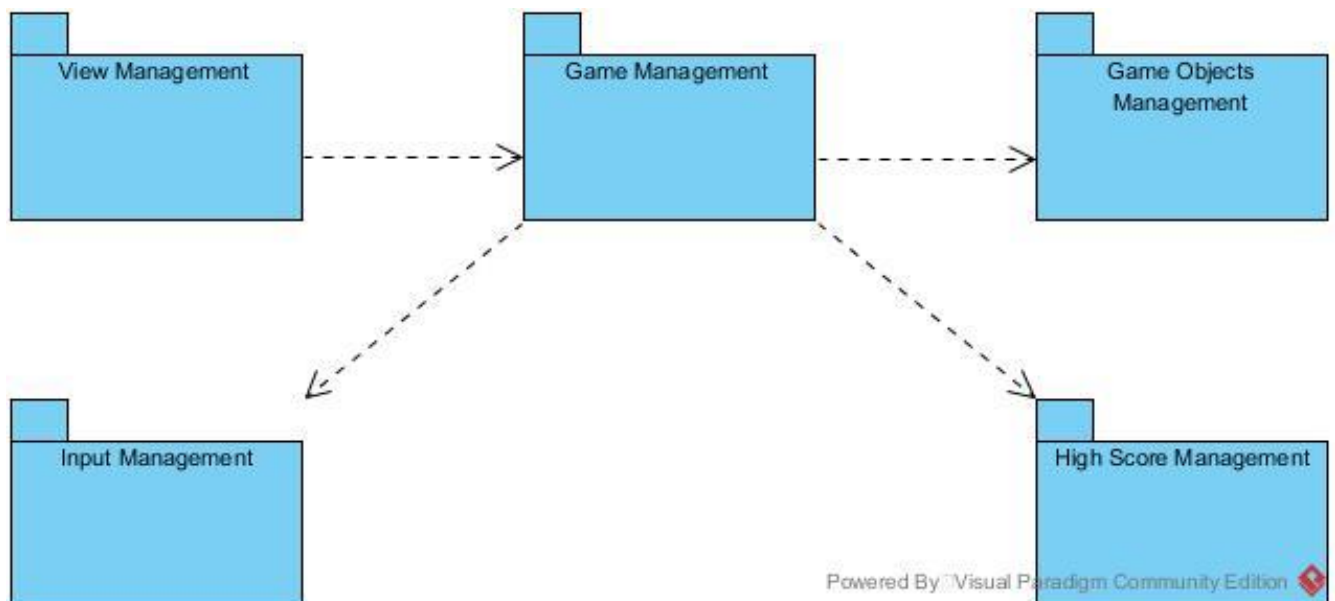


Figure for UML Package Diagram

b. Multi-tier Design Pattern

Interface Layer

For this system, the Interface layer is View Management which has all the menu objects and the play screen.

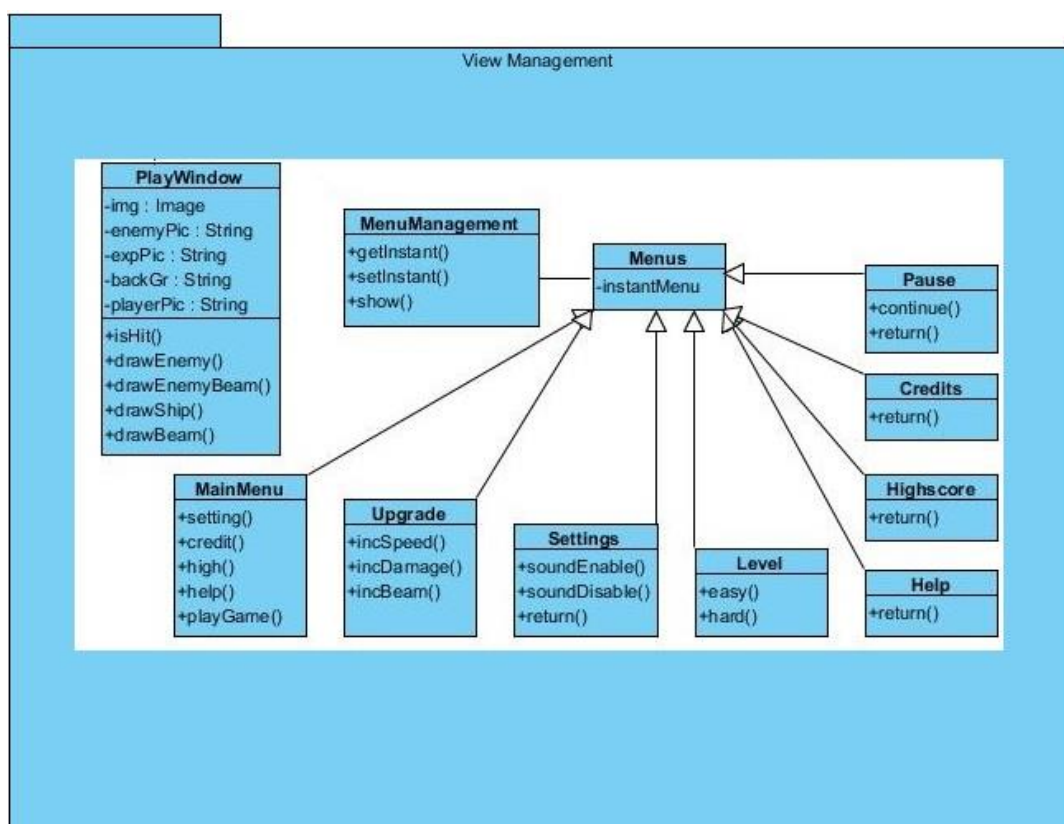


Figure for Layer 1 of Multi-Tier Design Pattern

Processing Layer

For this system, the processing layers are Game

Management and the Input Management which are making the processing part of the software system.

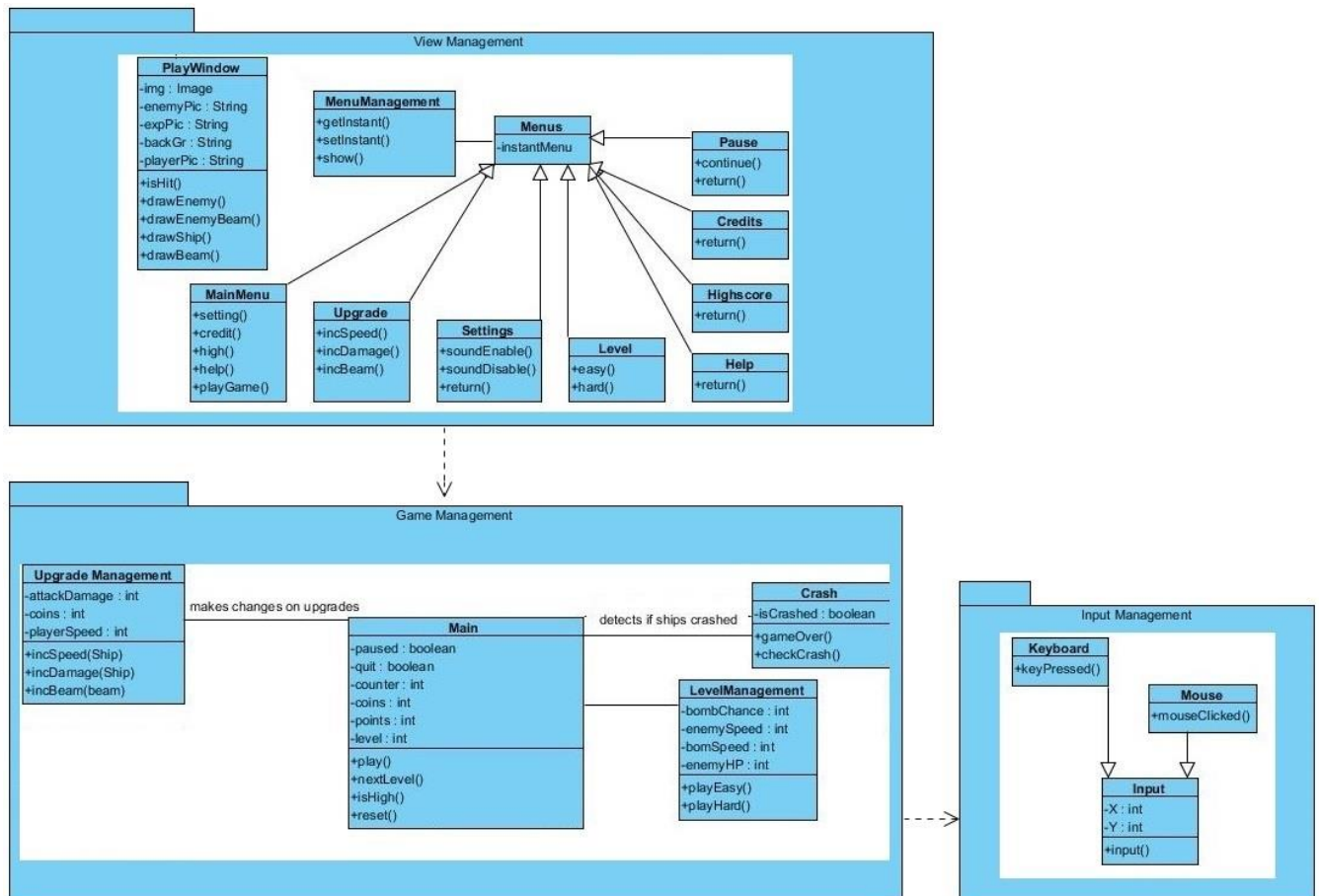


Figure: Layer 2 added to Design

Data Layer

For this system, the data layers are High Score Management and the Game Objects Management which are making the processing part of the software system.

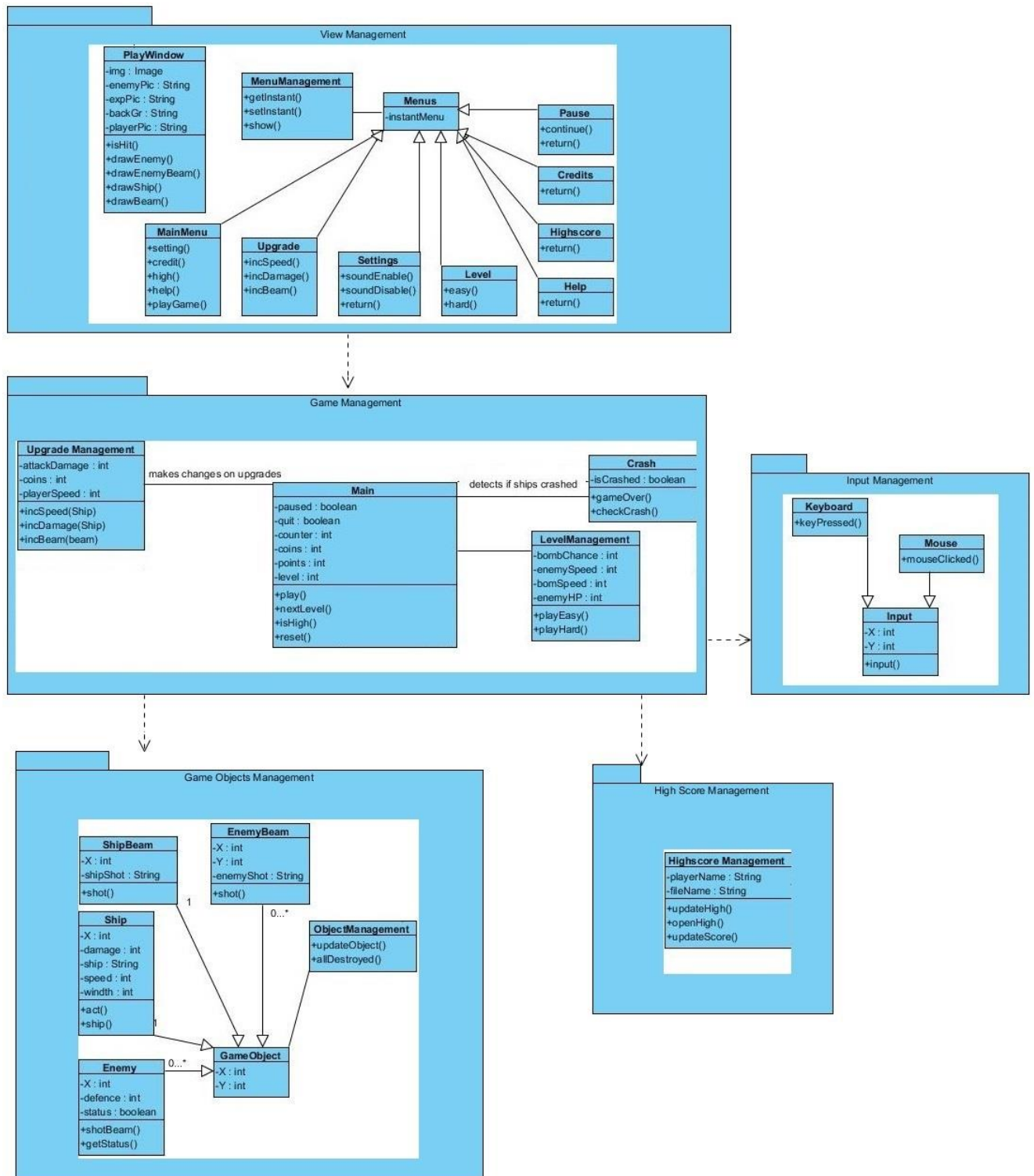


Figure: Layer 3 added to Design

6. References

- <http://www.uml-diagrams.org/package-diagrams.html>
- <http://www.win.tue.nl/~johanl/educ/2I45/ADS.03.ArchitectureStyles.pdf>