

# **COMP 302**

## **Project Report**

### **Group 9 - Mechatronic**

#### **Group Members:**

- Batuhan Yalçın
- Eray Sözer
- Eren Berke Demirbaş
- Gökhan Sarı
- Volkan Turgut

## Table of Contents

Introduction .....	3
Teamwork Organization.....	3
Use Case Diagrams.....	6
Use Case Narratives .....	6
1. Save Game .....	7
2. Give Extra Live As A Gift.....	8
3. Build The Game.....	9
4. Use A Taller Paddle Power-Up.....	11
5. Use Destructive Laser Gun Power-Up .....	12
System Sequence Diagrams .....	14
1. Save Game .....	14
2. Give Extra Live As A Gift.....	15
3. Build The Game.....	16
4. Use A Taller Paddle Power-Up.....	17
5. Use Destructive Laser Gun Power-Up .....	18
Operation Contracts .....	18
Sequence/Communication Diagrams .....	19
1. Aliens' strategy pattern sequence diagram: .....	19
2. GameView' singleton pattern sequence diagram:.....	20
3. Build Menu Sequence/Communication Diagram.....	20
Class Diagrams.....	22
Package Diagrams .....	22
Discussion of Design Alternatives and Principles.....	23
Supplementary Specifications.....	24
Glossary .....	27

## **Introduction**

We envision a next-generation game application, Asteroid Alien Crusher with the flexibility to support multiple players, advanced graphical user interfaces to extend the game experience and immersive game mechanics where you try to win the game by destroying all the obstacles while getting the highest score. We envision a next-generation game application, Asteroid Alien Crusher with the flexibility to support multiple players, advanced graphical user interfaces to extend the game experience and immersive game mechanics where you try to win the game by destroying all the obstacles while getting the highest score ranking in market.

## **Teamwork Organization**

### **1. Main**

Initially we didn't know that we were supposed to create new branches and merge them so everyone worked on this branch and pushed their work. After we learned it this branch was used to merge every other successful branch.

### **2. Synchronized-array**

gameObjects array is turned into synchronized array to get rid of errors, that occur when different parts of the program tries to modify the array at the same time. Gökhan worked on this branch.

### **3. GiftAsteroidAlien**

Before opening this branch we implemented the creation of aliens but they were being created randomly. In this branch we made the implementation of aliens being created after gift asteroids are destroyed. Eray and Eren worked on this branch.

### **4. NewFileSaveLoad**

Save-Load with a file is implemented in this branch. SaveLoadFileAdapter and SaveLoadFileService classes are implemented in this branch with adapter pattern. Save-load bugs are also fixed in this branch. Eray and Eren worked on this branch for saving file option.

## **5. All\_merged**

In this branch conflicts on the main branch solved. Gökhan, Eray, Batuhan and Eren testing branches merged here, Alien class with the strategy pattern also created here. Batuhan and Eren initialize Save/load adapter for save load database and file classes. Batuhan also adds some buttons for running panel logging panel in here. Also sum bugs are solved in this branch.

## **6. File\_LoadSave**

Initially, different method is tried on saving into a file (serialization). Then, it was seen that this method is not efficient way to save it. Then, we implemented file save option in NewFileSaveLoad branch. Eray and Eren worked on this branch.

## **7. SurprisingAlien**

This branch was created after the implementation of aliens. We created another alien type surprising alien which inherits the strategies of other aliens and has some other unique behaviors. It changes behaviors according to the number of asteroids.

## **8. updatedVisualElements**

Instead of using rectangles and circles for game objects, images for asteroids, the paddle, the ball and the background is added. Gökhan and Volkan worked on this branch.

## **9. Batuhan\_login**

Batuhan implemented login and signup screen in this branch with using different database then the save/load database. Save and Load Database also worked here.

## **10. Aliens**

In this branch we made the implementation aliens. We used strategy pattern to code the alien behaviors and hit behaviors of aliens. Hit behaviors activate when aliens are hit by the ball, alien behaviors are activated sometime after the alien is created. Eray and Eren worked on this branch together.

## **11. New\_main\_merge**

This branch helped to solve some conflicts on the main branch during merg.

## **12. Group-test**

This branch was used for class-testing, writing overview, abstract function, representation invariant, repOk method of paddle class. All group members are worked on this branch.

## **13. Eren\_testing**

This branch was used for testing intersects function of GameModel. Glass box and black box testing of this method is written in this branch. Function specifications is also written in this branch. Eren worked on this branch.

## **14. Volkan\_test**

This branch was used for testing isAsteroidTrue, and changeTargetLoc function of BuildModel. Glass box and black box testing of this method is written in this branch. Function specifications is also written in this branch. Volkan worked on this branch.

## **15. Eray\_Testing**

This branch was used for doing the testing of CorrectInputTest method of AsteroidButtons. Eray worked on this branch.

#### **16. Batuhan\_testing**

This branch was used for testing checkAsteroidAsteroidCollision function of GameModel. Glass box and black box testing of this method is written in this branch. Function specifications is also written in this branch. Eren worked on this branch.

#### **17. gokhanTest**

Includes tests for translateGameObjectForUI method in GameController class.

#### **18. eren\_final\_final\_branch**

Asteroid position change with mouse clicked implemented in this branch. The name of this branch is eren\_final\_final because we have a problem to create new branch and merge with others. That's why eren's branch and eren\_final\_branch is lost. This branch is created when we first started to work on our separate branches. Eren worked on this branch.

#### **19. Batuhan's\_Branch**

This Branch consist of some Batuhan' requirement in described week agendas. Inside each Asteroids, Asteroid conversion to game object, some ui prictures added in this branch. Some parts of the panel classes gameModel, buildModel, their controllers, and their Views changed fixed or implemented.

#### **20. Eray's\_branch**

Explosive asteroid implementation added. Eray worked on this branch.

#### **21. Moving-asteroids**

Added moving asteroids to the game.

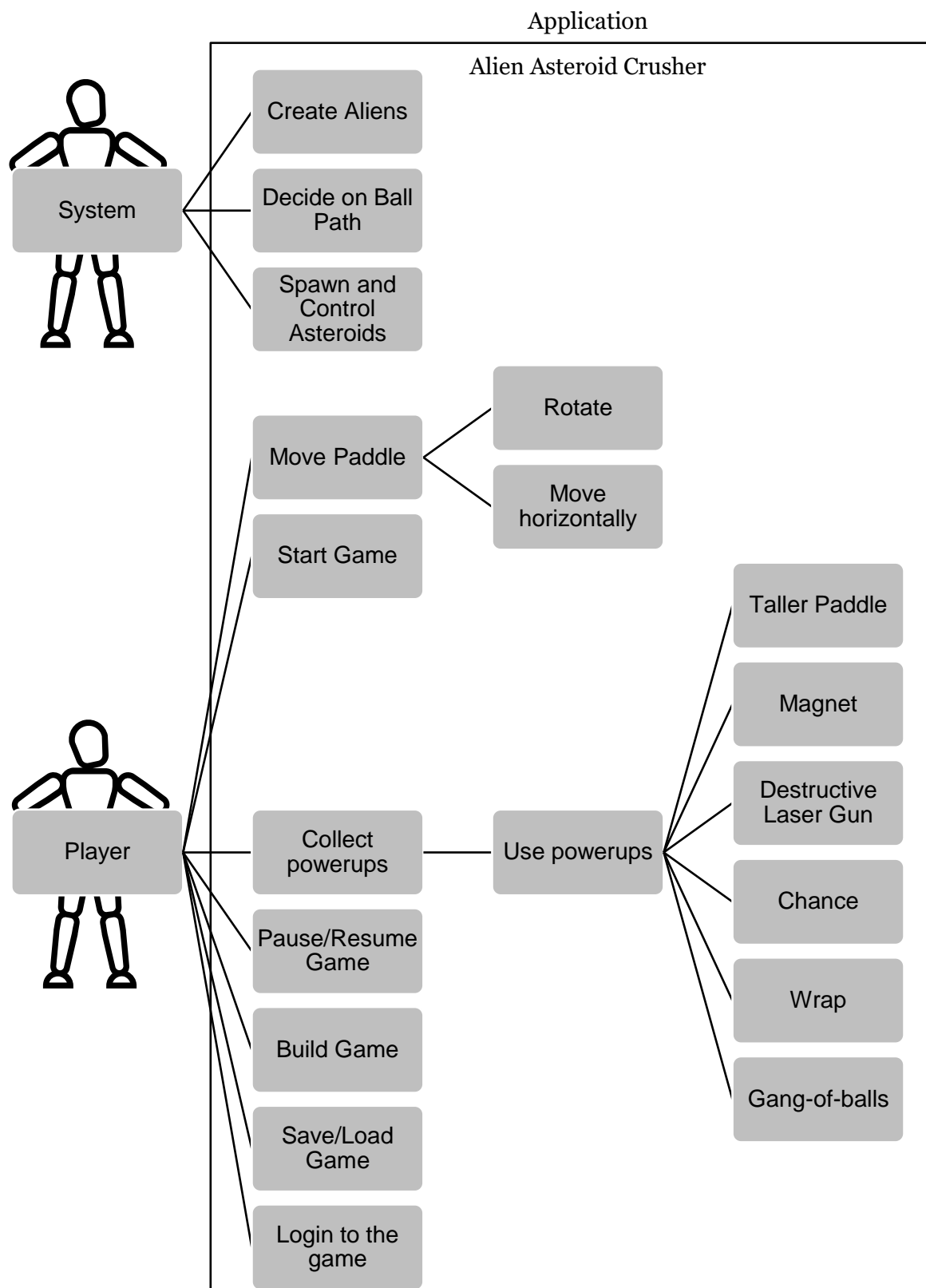
#### **22. controllerPattern**

Initial controller pattern of the gameModel-gameView-gameController, and buildModel-buildView-buildController constructed here. Eray and Gökhan worked on this branch.

#### **23. game-view-image**

Added image for the paddle. Implemented rotation, fixed some collision bugs. Gökhan worked on this branch.

## Use Case Diagrams



## Use Case Narratives

5 important/complex selected use case

## 1. Save Game

**Scope:** Alien Asteroid Crusher Application

**Level:** User-goal level

**Primary Actor:** Player

**Stakeholders and Interests Lists:**

- Player: wants to save the game into a file or to a database
- Database: stores the state of game with attributes such as username, types of asteroids and their numbers, their positions, player's scores, etc.

**Preconditions:**

- The player logs in to the game successfully.
- The game is successfully built in the building mode.
- If the game Running Mode is started, the game should be in pause mode.
- Application has necessary permissions to create files and access database management system.

**Success Guarantee (or Postconditions):**

- The state of game is saved into a file or database based on the player preference at any time.
- The file or database stored the information about the state of game properly.

**Main success scenario:**

1. The player logs in to the game.
2. The game is built successfully.
3. If the game is started, the game should be paused.
4. The player clicked the "Save" button in the panel or presses "Ctrl + S".
5. Then, a pop-up window emerges which player can choose either save into a file or to a database option.
6. The game state successfully saved into database or file directory.
7. The player either continues to play game or quits from the game.

**Extensions (Alternative Flow):**

2a. The game crashed after or during the player configuring the game in the building mode.

1. The build mode raises an error message which saying that player either has to give valid inputs or has to restart game again.
2. After the game is built successfully, the player can pause the game for saving.

3a. After building mode, the player needs to start button to play game.

1. Before pressing start button, player can save the existing game scheme built in the building mode.

6a. The player chooses to save into file option and chooses folder from the pop-up, but error occurs while saving into the file.

1. An error message emerges in the saving menu.
2. The game state automatically saved into predetermined path (for recovering purposes)

6b. The player chooses to save to a database option, but the database path is changed or not found in the computer.

1. The system throws an error message like “The database not found”
2. The system asks player to give correct database path or install database.

**Special Requirements:**

- A database created for the game with MongoDB or MySQL
- A backend code that interacts with the database
- A predetermined file directory for recovery purposes

**Technology and Data Variations List:**

- Mouse and keyboard buttons required to save game.
- A database

**Frequency of Occurrence:** Starting right after the building mode and continues until the end of the game (can occur at any time in this interval)

**2. Give Extra Live As A Gift**

**Scope:** Alien Asteroid Crusher Application

**Level:** Sub Function

**Primary Actor:** System

**Stakeholders and Interests Lists:**

- Player: would like to have an extra life for enjoyment of the game to last even more.
- System: Increases the life under two circumstances

**Preconditions:**

- The log in action completes successfully.
- The game is successfully built in the building mode.
- The game starts and the system gets into the running mode.

**Success Guarantee (or Postconditions):**

- The application is started.
- The game is built.
- The game has started.
- The player has less than 3 lives.
- The player has reached the threshold point for gaining an extra life.

**Main success scenario:**

1. The player logs in to the game.
2. The game is built in the building mode.
3. The game is started, and game gets into the state of running mode.
4. The player has lost at least one of his/her lives.
5. The player reaches the designated threshold point. (The threshold point will be determined later.)
6. The player gets their extra life successfully.

**Extensions (Alternative Flow):**



2a. The game crashed after or during the player configuring the game in the building mode.

1. The build mode raises an error message which saying that player either has to give valid inputs or has to restart game again.
2. After the game is built successfully, the player can pause the game for saving.

4a. The player plays the game until they have one remaining life and didn't reach the threshold point.

1. Player couldn't catch the ball and lose their one remaining life.
2. The game ends and a message pop up indicating that the game is over.
3. Player returns to the starting screen of the application.

**Frequency of Occurrence:** Starting right after the building mode and continues until the end of the game. It occurs one time high-most. It occurs when the player reaches the threshold point for gaining an extra life and has less or equal to two lives.

### 3. Build The Game

- **Scope:** Alien asteroid crusher application
- **Level:** User-goal
- **Primary actor:** Player
- **Stakeholders and Interests:**
  - Player: Creating obstacle and powerups layout. Choose the number of obstacles or places and type of the obstacles
- **Precondition:**
  - The game should start
- **Success Guarantee (or Postconditions):**
  - The game starts successfully with Chosen with a layout that consists of all the constrains the obstacles.
- **Main success scenario:**
  - 1- The Player chooses a layout or creates a new one
  - 2- If choose a layout it can go with step 5 otherwise continue the scenario
  - 3- Player defines the number of obstacles
  - 4- Player chooses the powerups
  - 5- The system put the obstacles randomly
  - 6- The player changes the position of the existing obstacle
  - 7- Player adds or removes obstacle type
  - 8- Player saves the layout
  - 9- The player runs the game
- **Extensions (or Alternative Flows):**
  - a\*. User (Player) related errors
  - b\*. System (Game) related errors

1b. Player select a layout, but the layout is not defined or defined in a wrong way.

- i. The system cannot upload the layout. The system throws the error message that says layout not defined
- ii. The layout is not as described. The player reports the issue

- iii. Return the map selection page
- 2b. Player enters invalid input to several objects
  - i. The system cannot carry all the number of objects if the user enters invalid number of objects, the system throws the error message and shows the valid condition to the player.
  - ii. The system returns success scenario step 2.
- 2b. Player gives empty the input.
  - i. The system throws a warning message to the player to inform empty selection.
  - ii. The system builds default obstacles
- 3b. Player choices invalid powerup choices
  - i. The system throws the error message and shows the valid powerups to the player.
  - ii. The system returns success scenario step 3.
- 4b. Player gives empty the input.
  - i. The system throws a warning message to the player to inform empty selection.
  - ii. The system builds default powerups.
- 5a. System cannot place all obstacles without collision.
  - i. The system tries to do randomization at most 5 times. If any of them succeed continue with the success scenario 5.
  - ii. If not succeed, return step 2.
- 6b. Player try to move obstacle to a new place. The player placed the obstacle invalid position.
  - i. The system throws an error message and the obstacle stay in the same position
- 7a. Player chooses a type of obstacle. However, the system couldn't find the obstacle feature.
  - i. The system gives the error message and returns the obstacle page.
- 7b. The Player chooses the invalid number of specific obstacle types.
  - i. The system throw exception. Says the conditions for the block types.
  - ii. The player cannot place the new obstacle and should continue with the next step.
- 7a. If the gift gives two aliens of the same type.
  - i. The system throws an error message two aliens of the same type cannot appear at once.
  - ii. The second alien will not appear.
- 8a. If the system couldn't save the layout.
  - i. The system throws an error message indicating that the layout couldn't save.
  - ii. Player can go run a step without saving the file.

- **Special requirement:**

- 1. Obstacle type and position are taken from the final version before the start.
  - 2. Players can change the difficulty of the game.
  - 3. Players can adjust the volume of sound

- **Technology and Data Variations List:**

- Obstacle List, game layout, Mouse button
- **Frequency of Occurrence:**
  - In the building step of the game

#### 4. Use A Taller Paddle Power-Up

**Scope:** Alien asteroid crusher application

**Level:** User-goal level

**Primary Actor:** Player

**Stakeholders and Interests Lists:**

- Player: wants to double the length of the paddle for 30 seconds

**Preconditions:**

- The player logs in to the game successfully.
- The game is built successfully (building mode).
- The game is started successfully (running mode).
- The player has at least one chance (live).
- The player has at least one “Taller Paddle” power-up.
- The player must press the “T” button or click the icon of power-up on the screen.

**Postconditions (Success Guarantee):**

- The length of the paddle has been doubled for 30 seconds.
- The double-length paddle is able to perform all actions (moving horizontally and rotating up to 45 degrees) of the normal paddle.

**Main Success Scenario:**

- 1) The player logs in to the game.
- 2) The new game is built, or the existing game scheme is loaded by the player.
- 3) The game starts.
- 4) The player collects at least one “Taller Paddle” power-up from gift asteroids.
- 5) The number of “Taller Paddle” power-ups is increased by 1 in the inventory and appears on the screen.
- 6) The player presses the “T” button or clicks the icon of power-up on the screen.
- 7) The number of “Taller Paddle” power-ups is decreased by 1 in the inventory and disappears from the screen.
- 8) The length of the paddle has been doubled for 30 seconds.

**Extensions:**

\*a At any time if the ball falls out of the game window:

1. The length of the paddle returns to its normal length.
2. The number of “Taller Paddle” power-ups is decreased by 1 in the inventory and disappears from the screen.

\*b At any time if the player has already used another power-up:

1. The game does not allow the player to use the “Taller Paddle” ability.

2. An indication occurs that showing player cannot use two abilities at the same time.
  3. If the player collects two “Taller Paddle” abilities and tries to use them before one is finished, the game does not allow players.
- 1.a The player enters the wrong login information.
    - 1) The system raises an error message until the player enters true login information.
  - 2.a System could not build a game successfully.
    - 1) The system raises an error message.

**Special Requirements:**

- The number of “Taller Paddle” power-ups is decremented and one of the icons of “Taller Paddle” disappears.
- A special timer to indicate the remaining time of power-up to the player

**Technology and Data Variations List:**

- Mouse and keyboard buttons are required to use power-up.
- Timer is required for measuring remaining time.

**Frequency of Occurrence:** Throughout the game

## 5. Use Destructive Laser Gun Power-Up

**Scope:** Alien asteroid crusher application

**Level:** User-goal level

**Primary Actor:** Player

**Stakeholders and Interests Lists:**

- Player: wants to use “Destructive Laser Gun” power-up that attaches guns to both ends of the paddle and destroys a full column of blocks, asteroids.

**Preconditions:**

- The player logs in to the game successfully.
- The game is built successfully (building mode).
- The game is started successfully (running mode).
- The player has at least one chance (live).
- The player has at least one “Destructive Laser Gun” power-up.
- The player must press the “D” button or click the icon of power-up on the screen to activate laser guns.

**Postconditions (Success Guarantee):**

- The destructive laser gun is attached to both ends of the paddle when the player activated this power-up.
- The destructive laser gun power-up allows the player to shoot five times with the click of the mouse or pressing the “D” button.

- The shots of a destructive laser gun are able to destroy the full column of blocks, even the firm asteroids.

**Main Success Scenario:**

- 1) The player logs in to the game.
- 2) The new game is built, or the existing game scheme is loaded by the player.
- 3) The game starts.
- 4) The player collects at least one “Destructive Laser Gun” power-up from gift asteroids.
- 5) The number of “Destructive Laser Gun” power-ups is increased by 1 in the inventory and appears on the screen.
- 6) The player presses the “D” button or clicks the icon of power-up on the screen.
- 7) The number of “Destructive Laser Gun” power-ups is decreased by 1 in the inventory and disappears from the screen.
- 8) The “Destructive Laser Gun” is activated and added to both ends of the paddle.
- 9) The player uses five shots of the “Destructive Laser Gun” power-up.

**Extensions:**

\*a At any time if the ball falls out of the game window:

1. If the player activated the “Destructive Laser Gun” ability and could not catch the ball, the ability is lost.
2. The number of “Destructive Laser Gun” power-ups is decreased by 1 in the inventory and disappears from the screen.

\*b At any time if the player has already used another power-up:

4. The game does not allow the player to use the “Destructive Laser Gun” ability.
5. An indication occurs that showing player cannot use two abilities at the same time.
6. If the player collects two “Destructive Laser Gun” abilities and tries to use them before one is finished, the game does not allow players.

1.a The player enters the wrong login information.

- 2) The system raises an error message until the player enters true login information.

2.a System could not build a game successfully.

- 2) The system raises an error message.

**Special Requirements:**

- The number of “Destructive Laser Gun” power-ups is decremented and one of the icons of “Destructive Laser Gun” disappears.

**Technology and Data Variations List:**

- Mouse and keyboard buttons are required to use power-up.

**Frequency of Occurrence:**

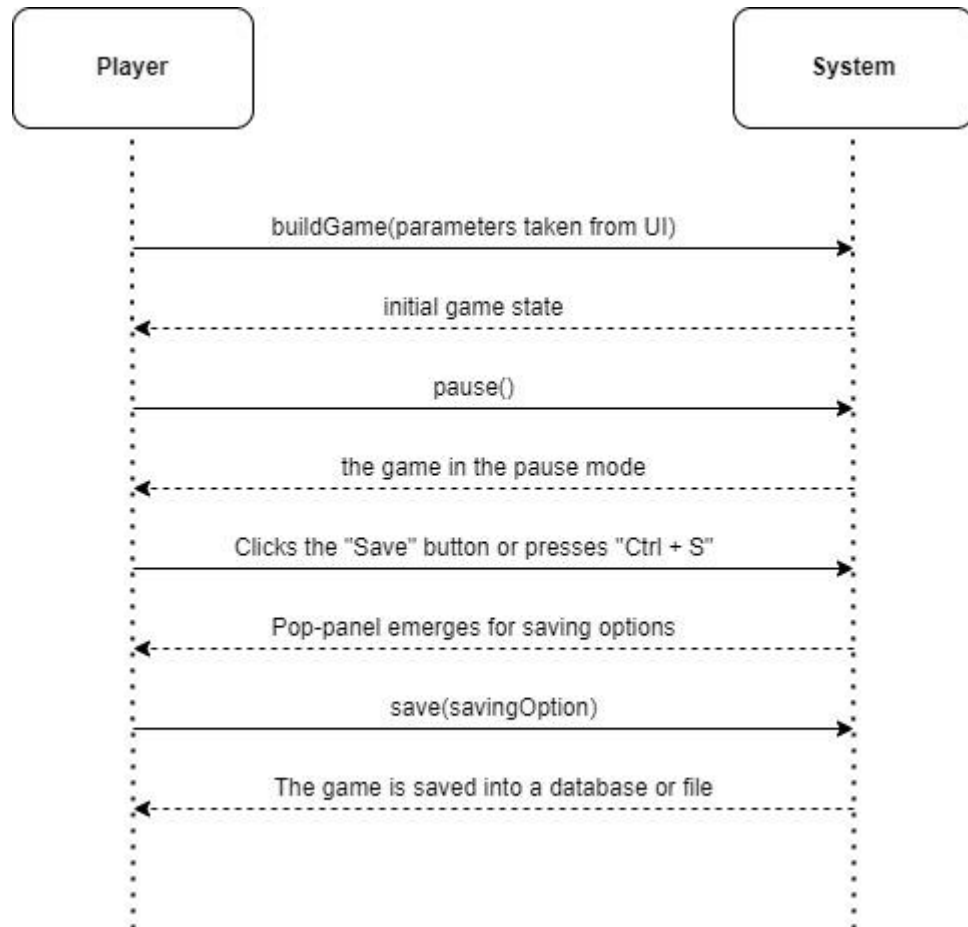
- Throughout the game “Destructive Laser Gun” can be collected and used.

- Each “Destructive Laser Gun” power-up provides five shots.

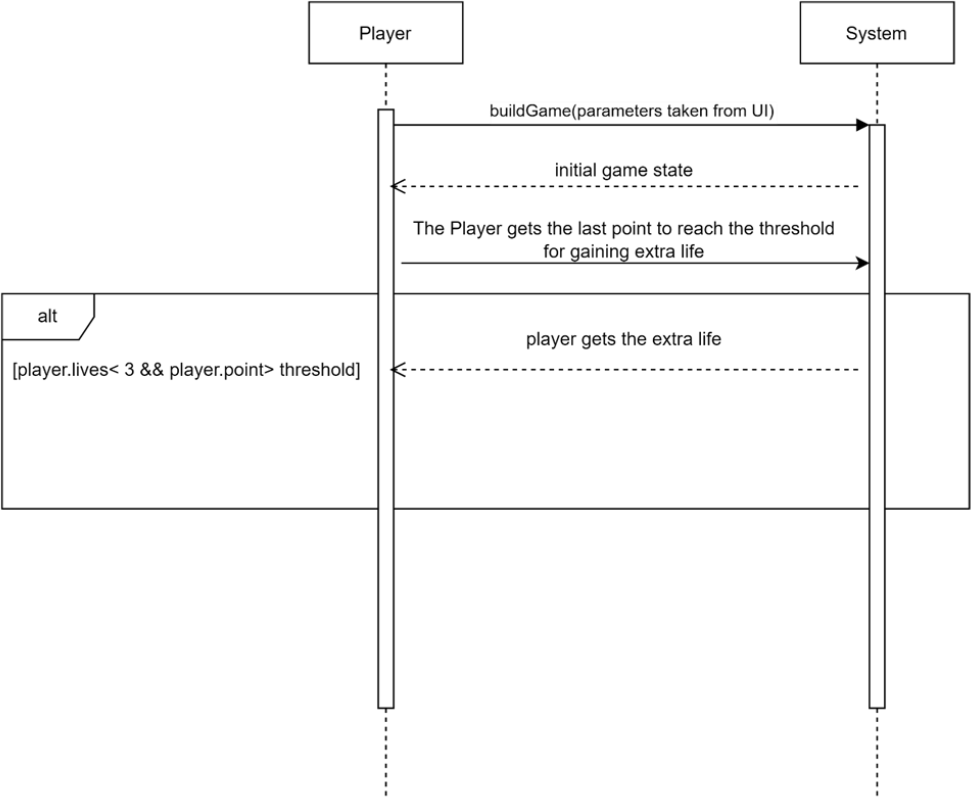
## System Sequence Diagrams

5 important/complex selected SSD's.

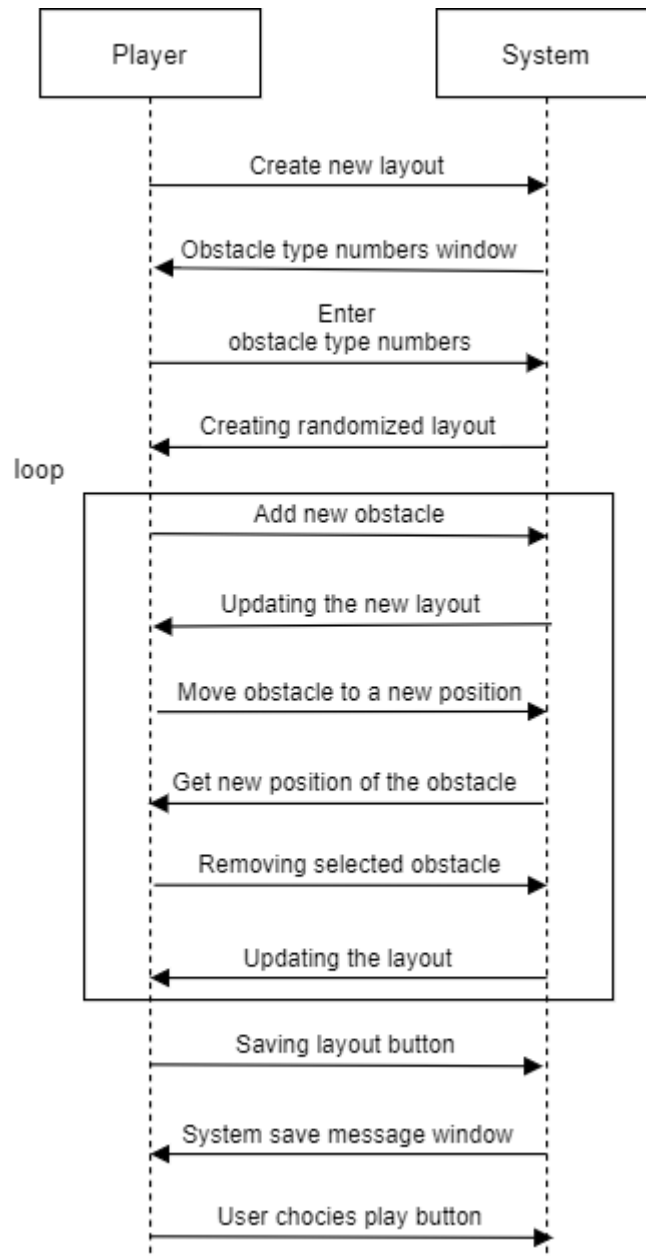
### 1. Save Game



2. Give Extra Live As A Gift

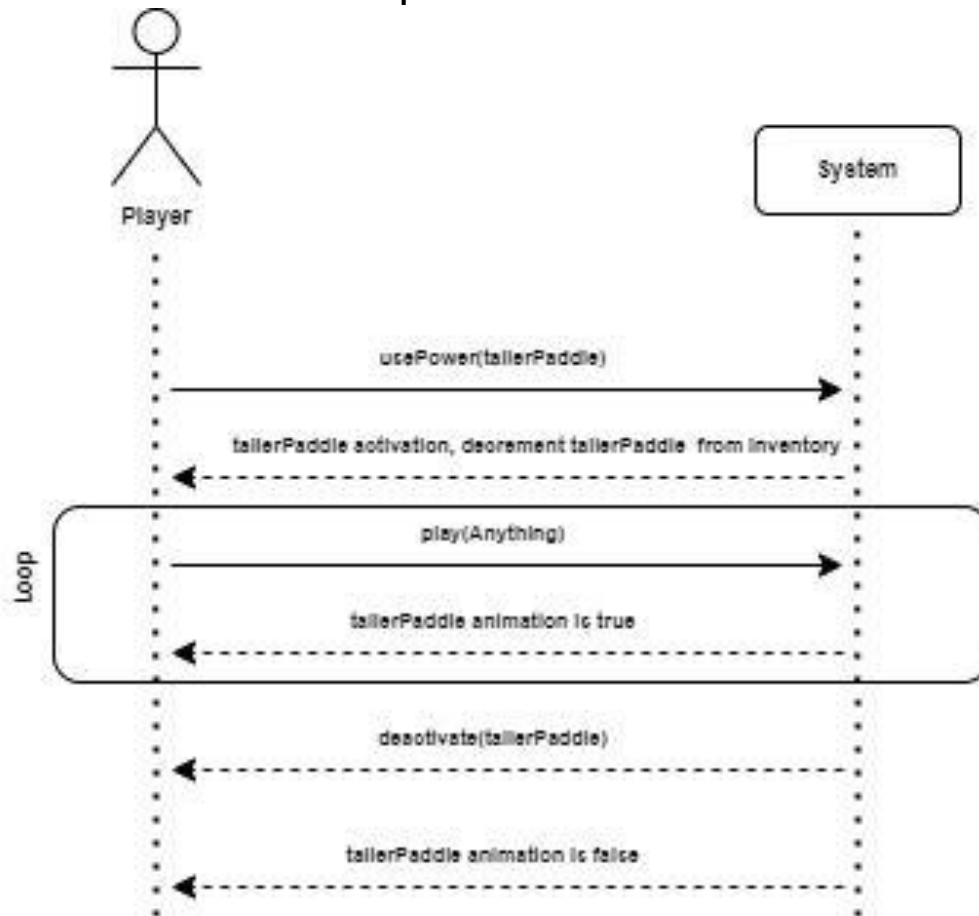


### 3. Build The Game

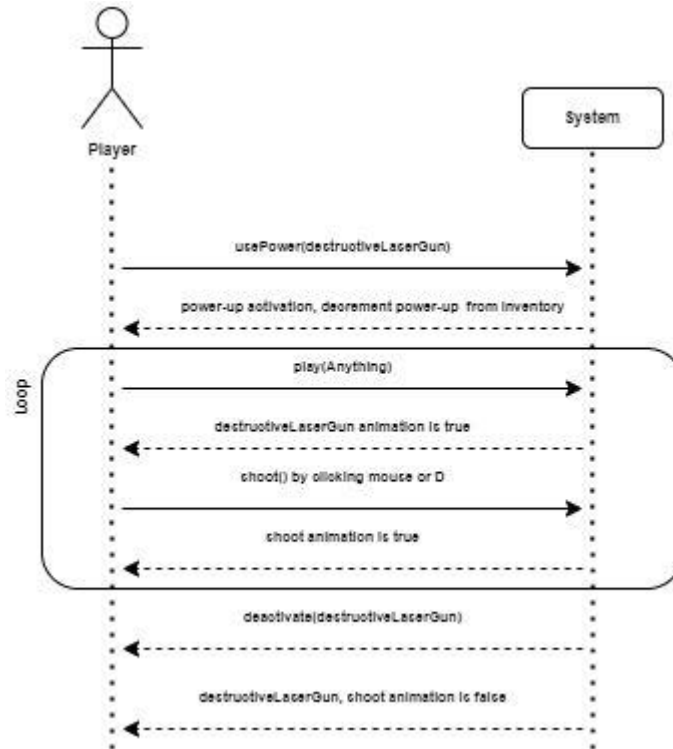




#### 4. Use A Taller Paddle Power-Up



## 5. Use Destructive Laser Gun Power-Up



### Operation Contracts

**Operation:** GainingExtraLife()

**Cross References:** Cases: The game gives one extra life as a gift for successful players.

**Preconditions:** Player should login to the game and the game is built and started to run.

The player has reached the critical point and has less than 3 lives at the time their score goes past the critical point.

**Postconditions:** The game increases the life of the player if the preconditions are met.

**Operation:** SaveGame()

**Cross References:** Cases: Saving the game

**Preconditions:** Player should login to the game and the game is not running  
The user has permissions to save a file or has a Database

**Postconditions:** The game is successfully saved into a file or database

LoadGame() option and button is activated

**Operation:** useTallerPaddle()

**References:** Cases: Use "Taller Paddle" power-up.

**Preconditions:** Player has "Taller Paddle" power-up and chance.

**Postconditions:** Length of the paddle is doubled for 30 seconds (attribute modification).

The number of "Taller Paddle" power-up is decremented from

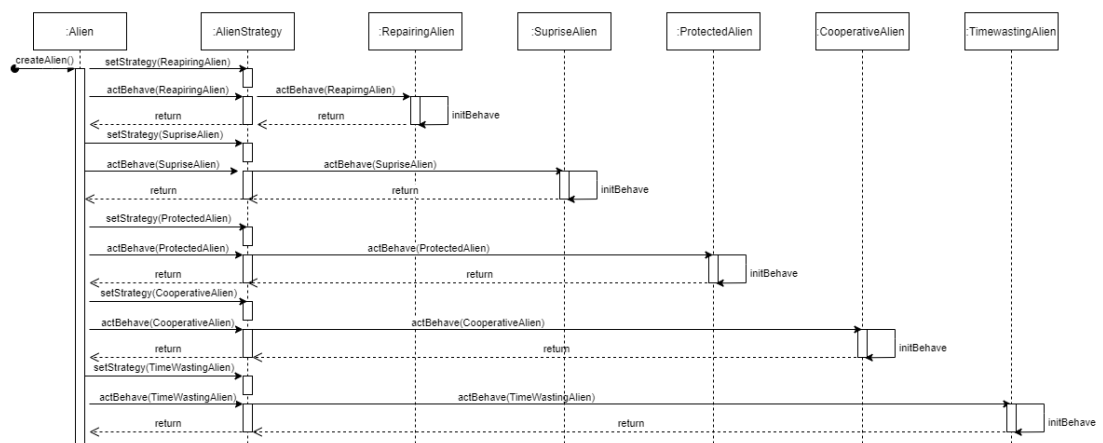
Inventory (attribute modification).  
30 seconds timer is activated to deactivate power-up.

**Operation:** useDestructiveLaserGun()  
**References:** Cases: Use “DestructiveLaserGun” power-up.  
**Preconditions:** Player has “DestructiveLaserGun” power-up and chance.  
**Postconditions:** Paddle’s destructiveLaserGun property is set true (attribute modification).  
 Number of “Destructive Laser Gun” power-up is decremented from Inventory (attribute modification).  
 When the player press D button or click mouse 5 times or time is finished, the paddle’s magnet property is set false (attribute modification).

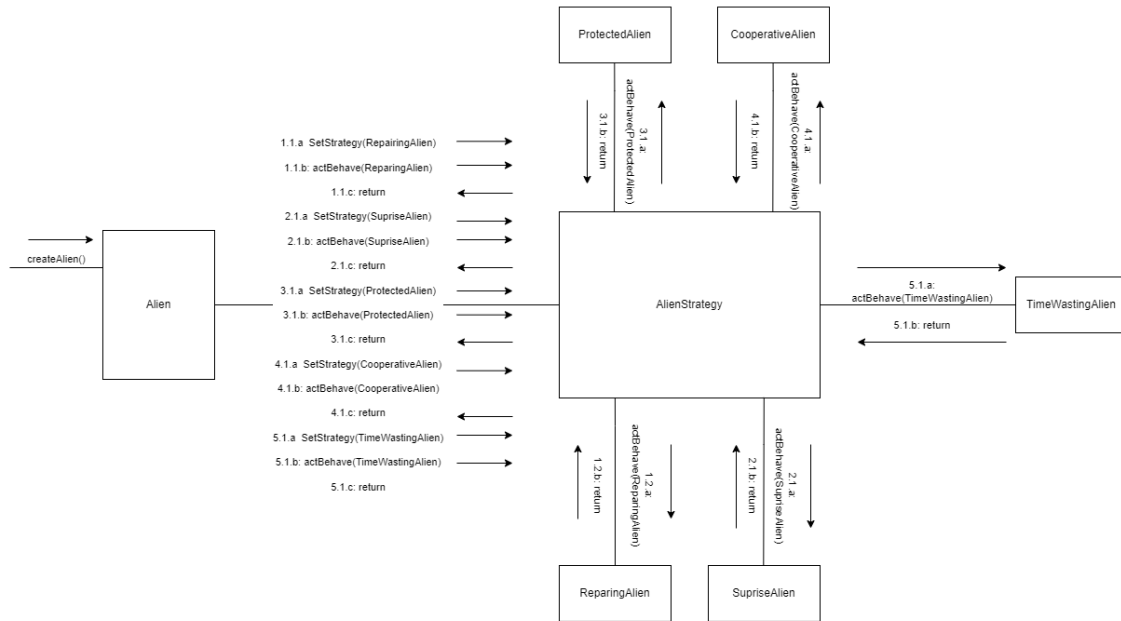
**Operation:** SigningUp()  
**Cross References:** Cases: Creating an account for the Alien Asteroid Crusher.  
**Preconditions:** The application must have been just started. (Right after starting the application.)  
 The player must designate a ID, Username and a password, and he/she must enter it to the appropriate form entries in the initial window of the program.  
**Postconditions:** After signing up successfully, the player remains on the same page.  
 Need to return login page to open the game.

## Sequence/Communication Diagrams

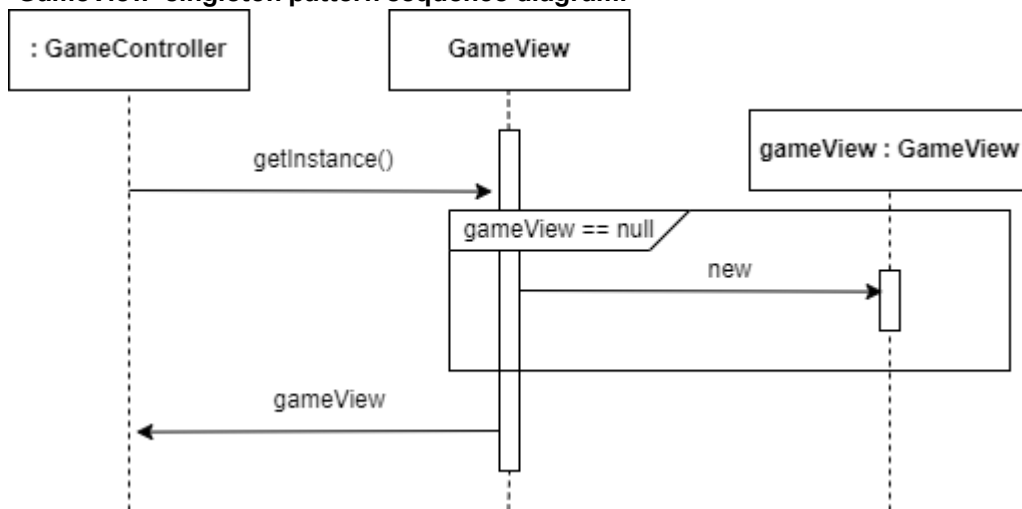
### 1. Aliens’ strategy pattern sequence diagram:



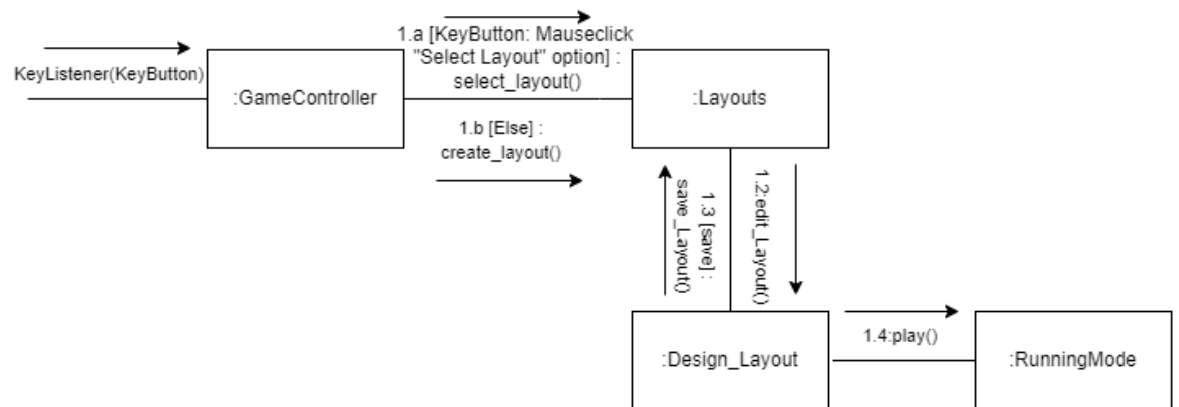
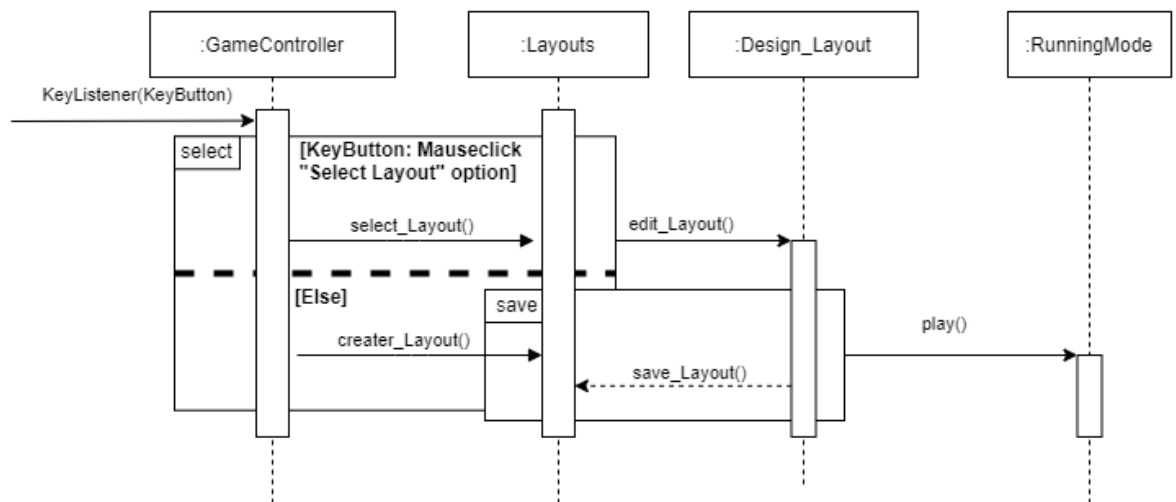
Aliens’ strategy pattern communication diagram



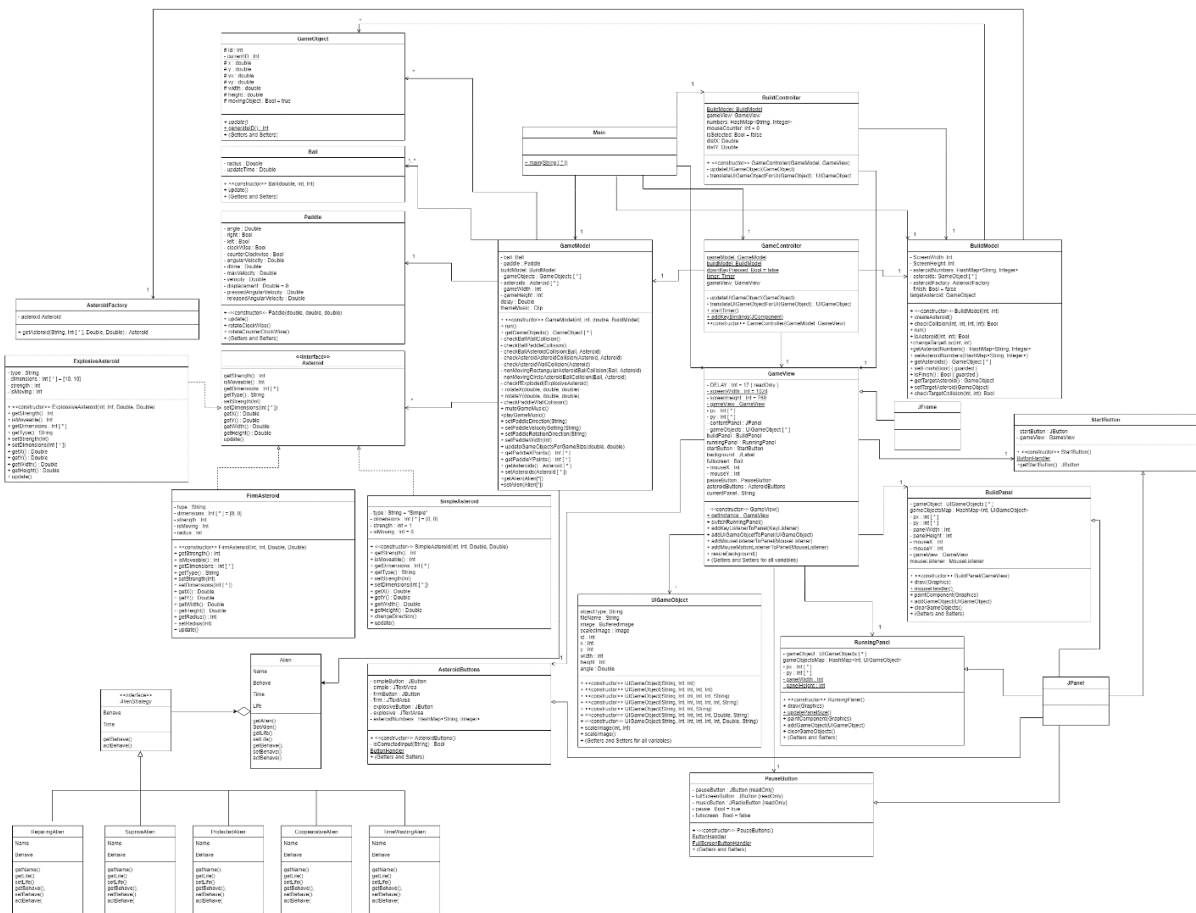
## 2. GameView' singleton pattern sequence diagram:



## 3. Build Menu Sequence/Communication Diagram



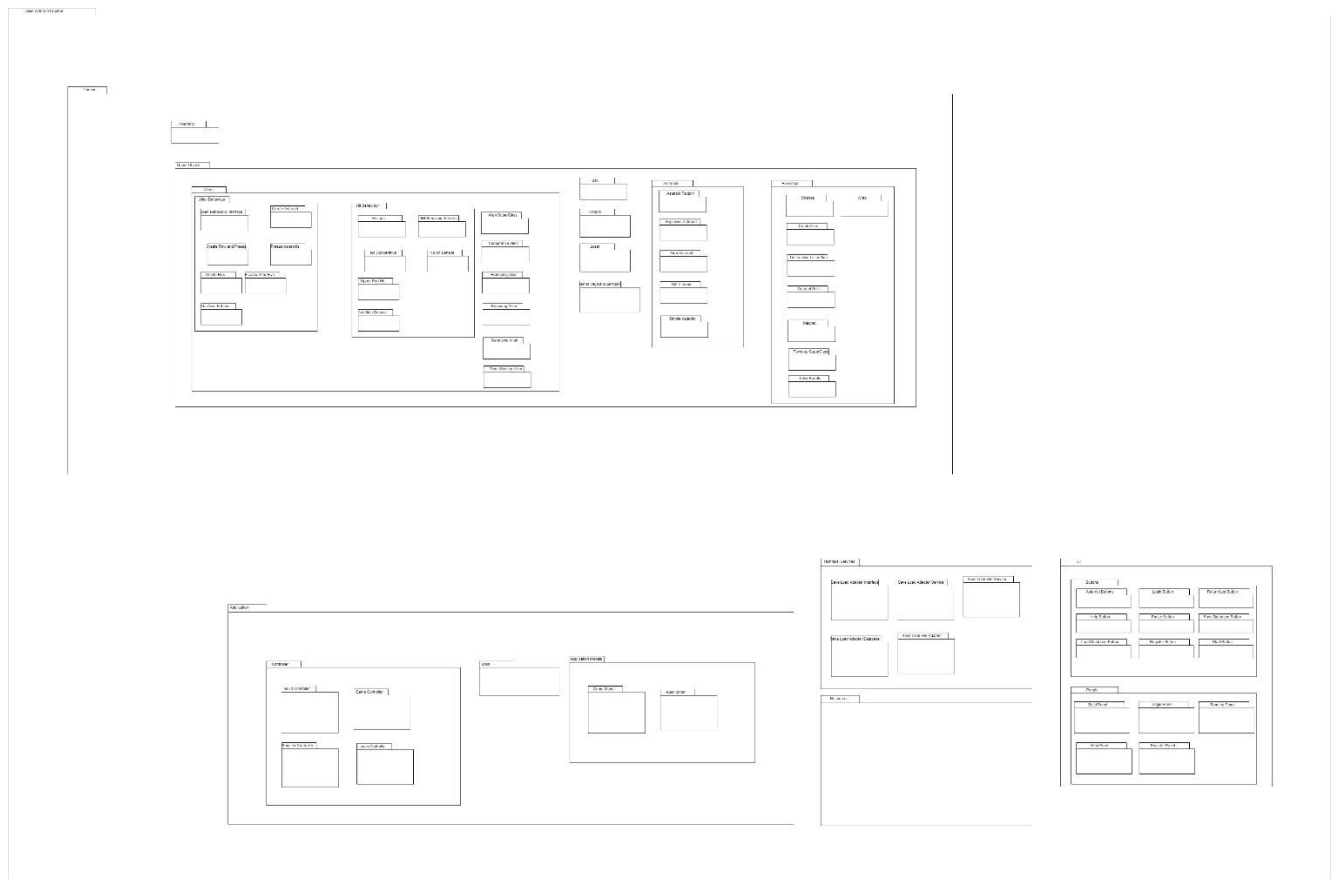
## Class Diagrams



Link: <https://app.diagrams.net/#G1O5dEiqMrA-zybKmoUTacIMbYE-1CMEv>

## Package Diagrams

## Layers and packages



Link: <https://app.diagrams.net/#G1O5dEiqMrA-zybKmoUTacIMbYE-1CMEv>

## Discussion of Design Alternatives and Principles

In our project we used many GRASP principles such as Controller for GameController class, Creator for AsteroidFactory, Information Expert where GameModel knows the state of Game Objects and to access Game Objects you need to ask Game Model etc.

We also used many GoF patterns such as Factory for Asteroid Factory, singleton for GameModel and GameView, Adapter for load and Save, Strategy for Aliens etc. However, after finishing the project and looking back at it we realized we could have used alternative patterns for some use cases which might have eased our troubles such as Observer and Visitor Patterns.

## Observer

In our game, there are elements whose state change depending on other elements of the game. For instance, the `UIGameObject` array in the `UI View` depends on `GameObject` array in the `Game Model`. If a game object falls out of the game borders, it is removed from both `GameObject` array and `UIGameObject` array. By using Observer pattern, we could have made `UIGameObject` array a subscriber of `GameObject` array, and any changes happened to the `GameObject` array would have affected the other array. This pattern could also be useful at other parts of the game as well.

## Visitor

We could have used visitor pattern for collision between ball and other game objects such as asteroid, aliens and paddle. Ball would have been the visitor and implement a HitVisitor interface and rest of the game objects would have been visitable. When ball hits the

game objects it visits that game object and accepts the ball as visitor. Ball gets the reference for that game object. Then we could have handled the collision in our HitVisitor class which implements Visitor class. This way we could have grouped the collision methods in one class and adding new collision methods would be much easier. For example, HitVisitor could visit moving asteroids which results in different kind of reflection or frozen asteroids which results in different kind of effect. Implementing all of them in one class would have been easier.

## Supplementary Specifications

### Revision History

Version	Date	Description	Authors
Final Draft.	June 7, 2022	Final Draft.	Batuhan Yalçın Eray Sözer Eren Berke Demirbaş Gökhan Sarı Volkan Turgut

### Introduction

This document is the repository of all Alien Asteroid Crasher game requirements which are not specified in the use cases.

### Functionality

Functionality is common across many use cases.

### Logging and Error Handling

- Log all errors to persistent storage.
- Errors should not crash the app. In case of an error, a pop-up screen should appear explaining the problem, if possible.

### Usability

- Java VM should be installed.
- An operating system that supports GUI.

### Human Factors:

- Objects on the screen should be big enough for the player to see clearly.
- Different types of asteroids and aliens should have distinct looks to distinguish each of them and not confuse them
- The ball and paddle shouldn't move very fast so that a human eye can track them.
- When the game is paused and then resumed there should be a countdown from 3 for the player to adjust to the scenario and shouldn't start straight away.
- The paddle should be responsive the player's inputs should be implemented straight away.
- When an asteroid is hit, and an alien is killed there should be a hit animation and sound to clearly show a hit occurred.
- Score and lives text should be big.

### Reliability

### Recoverability



- If there is an error causing a game crash, the program should try to handle the error in the background. This may cause a pause of the game or the necessity of restarting. Much more analysis is needed here (will be added later).

#### **Performance**

- The game should be able to handle multiple balls smoothly.
- Frame rate should not go below 30 fps.

#### **Supportability**

- Help screen (a separate JFrame) will support the player during the game. The player can find solutions to common problems and game instructions in the help screen.

#### **Adaptability**

#### **Configurability**

#### **Interfaces**

##### **1- Noteworthy Hardware and Interfaces**

- Screen
- Keyboard
- Mouse
- Speaker

##### **2- Software Interfaces**

- Java
- Database

#### **Special Requirements**

##### **1. Hardware and software constraints**

- The game should be playable in most hardware systems; therefore, the game should not require too much processor power.
- Any GUI based operating system is required.
- JVM support is required.
- Java Swing library will be used.

##### **2. Age limitations**

- The player should be older than 13+ years old.

##### **3. Physical environment concerns**

- Heat is likely to cause performance drops, therefore, the hardware should be well cooled.

#### **Application-Specific Domain (Business) Rules**

<b>ID</b>	<b>Rule</b>	<b>Changeability</b>	<b>Source</b>
<b>RULE 1</b>	<b>Building Mode Rules:</b> 1- At least 75 simple asteroids 2- At least 10 firm asteroids 3- At least 5 explosive asteroids 4- At least 10 gift asteroids	Low Every map should have at least this	Project Manual

		many asteroids	
RULE 2	<b>Ball Hits a Moving Object:</b> <ol style="list-style-type: none"> <li>1- If the object is moving parallel to the ball, it will reflect like the case of non-moving object but gain 5px/second</li> <li>2- If the object and ball is moving in opposite direction the ball will reflect with 180 degrees and its speed will stay the same</li> <li>3- If the object and ball is moving in perpendicular direction ball will reflect with an angle of 45 degrees</li> </ol>	Low These are the rules of physics of this game and these rules can't be changed	Project Manual
RULE 3	<b>Rotate Paddle Rule:</b> <ol style="list-style-type: none"> <li>1- The paddle can be rotated by up to 45 or 135 degrees.</li> <li>2- The rotation speed is 20 degrees/second until reaching maximum rotation angle.</li> <li>3- When the player releases the paddle, the rotation speed will be 45 degrees/second to turn back its horizontal state.</li> </ol>	Low These rotation degrees and rotation speed are given in the project manual. Therefore, there is no potential changes planned for this rule.	Project Manual
RULE 4	<b>Ball Hits a Non-Moving Object:</b> <ol style="list-style-type: none"> <li>1- If it hits a side of an object it will reflect with an angle equal to and symmetric around the norm of the hit surface</li> <li>2- If it hits a corner the same thing will happen but this the time the reflect surface will be an imaginary surface making 45-degree angle with corner.</li> </ol>	Low These are the rules of physics of this game and these rules can't be changed	Project Manual
RULE 5	<b>Move Paddle Rule:</b> <ol style="list-style-type: none"> <li>1- If player presses and releases the left or right arrow buttons, the paddle will move by an offset equal to <b>L/2</b> with a movement speed of <b>L/second</b>.</li> <li>2- If player does not release the button, the paddle will move with a movement speed of <b>2*L/second</b>.</li> </ol>	Low These movement offset and movement speeds are given in the project manual. Therefore, there is no potential changes planned for this rule.	Project Manual
RULE 6	<b>Score Calculations:</b> <ol style="list-style-type: none"> <li>1- When an asteroid is destroyed the score of the player will be updated using following formula:  <math display="block">NewScore = OldScore + \frac{300}{CurrentTime - GameStarting}</math> </li> </ol>	Low This score calculation should always stay the same however some	Project Manual

		extra bonus points that increase the score can later be added.	
RULE 7	<b>Size of Game Objects Rule:</b> <ol style="list-style-type: none"> <li>1- The paddle length <b>L</b> will be the <b>10%</b> of the game screen's width.</li> <li>2- The radius of explosive asteroids is <b>10px</b>. These asteroids are circular.</li> <li>3- The firm asteroids are also circular with a radius equal to <b>10px + (number of hits required to destroy * 1px)</b>.</li> <li>4- The remaining asteroids are rectangles with dimensions <b>L/5 and 20px</b>.</li> <li>5- The ball is a <b>17×17</b> pixels object.</li> <li>6- The paddle thickness <b>T</b> is <b>20px</b>.</li> </ol>	Low These sizes of objects are given in the project manual. Therefore, there is no potential changes planned for this rule.	Project Manual
RULE 8	<b>Controls:</b> <ol style="list-style-type: none"> <li>1- Move paddle left horizontally: arrow left key</li> <li>2- Move paddle right horizontally: arrow right key</li> <li>3- Rotate paddle left: A button</li> <li>4- Rotate paddle left: D button</li> <li>5- Throw ball: W key or left mouse button</li> <li>6- Use magnet: M key</li> <li>7- Use taller paddle: T key</li> <li>8- Use chance: C key</li> <li>9- Use wrap: V key</li> </ol>	Low These buttons are written in manual	Project Manual

## Glossary

Glossary

## Revision History

Version	Date	Description	Authors
Final Draft.	June 7, 2022	Final Draft.	Batuhan Yalçın Eray Sözer Eren Berke Demirbaş Gökhan Sarı Volkan Turgut

## Definitions

Term	Definition and Information	Format	Validation Rules	Aliases
Abyss	A place located underneath the paddle where when the ball goes	A concept		Out of Screen

	there player loses the ball and loses a life			
Alien Asteroid Crusher	An entire game application consisting of several pages, modes, etc. (login page, building mode, running mode)	An executable		Application, game
Ball	A circular object that can break asteroids and kill aliens that also reflects and bounces every time it hits an object such as the paddle, wall, asteroids, and aliens.	PNG		Ball
Building Mode	A game mode that the player needs to complete before going into a running mode in which he creates a layout of different kinds of asteroids.	A concept		Building Mode
Chance	A power-up that increases player's chances by 1.	A concept		Extra live
Cooperative Alien	An alien helps the player by destroying all asteroids in one row	PNG		Cooperative Alien
Destructive Laser Gun	A power-up that adds two laser guns to both ends of the paddle which can be fired by the player (5 times).	A concept		Gun
Explosive-asteroids	A rectangular obstacle that explodes after it is hit. The neighbor asteroids within 2*L pixels radius also get after the explosion.	PNG		Explosive-asteroids
Firm-asteroid	A circular obstacle that requires multiple hits to break, depends on the radius of the obstacle. After each hit, it gets smaller.	PNG		Firm-asteroid
Gang-of-balls	A power up that replicate ball 10 times and release them into the game.	A concept		Gang-of-balls
Gift Asteroids	A rectangular obstacle that can be broken with one hit. After broken, either a power-up will reveal, or aliens will be triggered.	PNG		Gift Asteroids
Help screen	A pop-up screen that shows the instructions	JFrame		Help screen

	and explanations about the game.			
Lives	The total number of players' rights to play the game.	An integer		Chances
Login screen	A screen that appears when the game is started. The user enters his/her unique username to log in to the game.	JFrame		Login screen
Magnet	A power-up that allows the paddle to catch the ball and release it whenever the player activates.	A concept		
Paddle	A rectangular object controlled by the player, located at the bottom of the screen which can move horizontally and rotate	PNG		Rectangular reflector
Pause/Resume Button	A button located on the top left of the screen pauses the game when it is running and resumes the game when paused	JButton		Pause/Resume Button
Protecting Alien	An alien is located underneath asteroids to protect them from getting hit by the ball but vulnerable against hits on their top side.	PNG		
Repairing Alien	An alien that builds back the broken simple asteroids every 5 seconds and can get destroyed when hit by the ball	PNG		Repairing Alien
Running Mode	A game mode which starts when player starts the game and ends when player is out of lives in which he can control the paddle, break asteroids, kill aliens and gain score.	A concept		Running Mode
Score	The total number of points gained by destroying asteroids with respect to rules.	An integer		Points
Simple Asteroid	A rectangular obstacle that can be broken in one hit without giving any power-ups.	PNG		Simple asteroid
Taller Paddle	A power-up that doubles the length of the paddle for 30 seconds.	A concept		Expansion of paddle

Time Wasting Alien	An alien freezes asteroids for 15 seconds to protect them against the ball	PNG		Time Wasting Alien
Wall	Edges of the screen from which the ball can bounce.	A concept		
Wrap	A power-up that allows player to move along the entire screen (make transitions between both ends of the screen)	A concept		Wrap