

COMP-302: SOFTWARE ENGINEERING

NEED FOR SPEAR

Phase1: REQUIREMENTS AND MODELING

29/10/2021

SyntacticSugar:

İrem Demir

Atakan Kara

Asu Tutku Gökçek

Farrin Marouf Sofian

Eren Yenigül

Table of Contents

[UML Use Case Diagram](#)

[Use Case Narratives](#)

- [Use Case 1: Play the Game](#)
- [Use Case 2: Save The Game](#)
- [Use Case 3: Build\(Setup\) the Game](#)
- [Use Case 4: Load the Game](#)
- [Use Case 5: Pause the Game](#)
- [Use Case 6: Resume the Game](#)
- [Use Case 7: Move the Phantasm](#)
- [Use Case 8: Rotate the Phantasm](#)
- [Use Case 9: Hit the Simple Obstacle \(Wall Maria\)](#)
- [Use Case 10: Hit the Firm Obstacle \(Steins Gate\)](#)
- [Use Case 11: Hit the Explosive Obstacle \(Pandora's Box\)](#)
- [Use Case 12: Hit the Gift Box \(Gift of Uranus\)](#)
- [Use Case 13: Equip Magical Hex](#)

[Domain Classes](#)

[System Sequence Diagrams](#)

[Operation Contracts:](#)

- [Contract CO1: Open Game:](#)
- [Contract CO2: Play Game:](#)
- [Contract CO3: Setup Game:](#)
- [Contract CO4: Shoot:](#)
- [Contract CO5: Pick Magical Ability:](#)
- [Contract CO6: Move Paddle:](#)
- [Contract CO7: Rotate Paddle:](#)
- [Contract CO8: Pause Game:](#)
- [Contract CO9: Resume Game:](#)
- [Contract CO10: Save Game:](#)
- [Contract CO11: Load Game:](#)

[Supplementary Specification](#)

- [Revision History](#)
- [Introduction:](#)
- [Functionality:](#)
- [Usability:](#)
- [Reliability:](#)
- [Performance:](#)

[Supportability:](#)

[Purchased Components:](#)

[Implementation Constraints:](#)

[Interfaces:](#)

[Vision](#)

[Introduction:](#)

[Positioning:](#)

[Business Opportunity:](#)

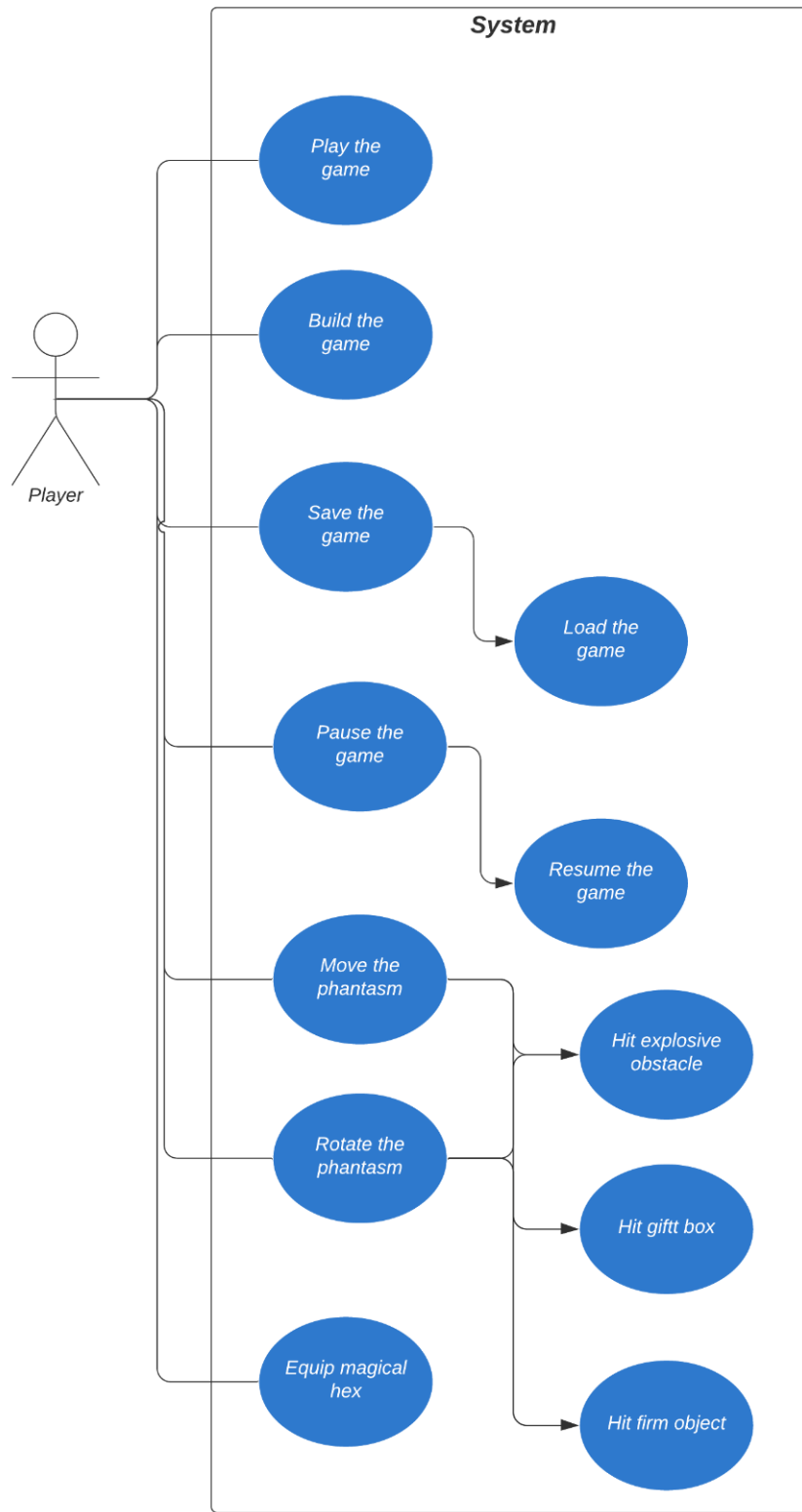
[Problem Statement:](#)

[Product Position Statement:](#)

[Alternatives and Competition:](#)

[Glossary](#)

UML Use Case Diagram



Use Case Narratives

Use Case 1: Play the Game

Use Case Name: Play the game

Scope: Game

Level: user-goal

Primary Actor: Player

Stakeholders and Interests: Player / wants to play Need for Spear's game

Preconditions: Player has a computer in which required softwares is installed.

Success Guarantee: Game opened and processed without errors.

Main Success Scenario:

1. Player opens the game.
2. Build mode shown to the user.
3. Player performs [Build\(setup\) Game](#).
4. Game starts with the selected layout.
5. Player controls a noble phantasm to protect the enchanted sphere.
6. Enchanted sphere interacts with obstacles in the game.
7. Game state updated according to the result of the interaction.
8. System repeats between 5-7 until all obstacles are cleaned.
9. Player wins the game.

Extensions:

*a At anytime, player performs [Pause the Game](#)

1. Player performs [Resume the Game](#)
 2. Player performs [Save the Game](#)
 3. Player goes back to start
 4. Player exits the game
- 5.a Player can't protect the enchanted sphere.
5.b Player loses live.
5c. If the player is out of lives, the game's lost screen appears.

Special Requirements:Game is installed on the machine.

Technology and Data Variations List: A database

Frequency of Occurrence: Every time player wants to play a game.

Miscellaneous:

Use Case 2: Save The Game

Use Case Name: Save the Game

Scope: Game

Level: user-goal

Primary Actor: Player

Stakeholders and Interests: Player/ Don't want to lose progress

Preconditions: Game starting successfully, The game should be paused

Success Guarantee: Save file is created and data is saved properly

Main Success Scenario:

1. Player clicks on the save button
2. A pop-up window shown asking for confirmation
3. Player clicks on Yes
4. Save the required information to the database
5. Feedback pop-up is shown to player for a successful save

Extensions:

- 3a. Player clicks No
 - 3a.1 pop-up disappears (nothing saved)
 - 3a.2 Game goes back to paused state
- 4a. Game couldn't be saved.
 - 4a.1 A pop-up shown to the player

Special Requirements:

Technology and Data Variations List: The data should be stored in a Database

Frequency of Occurrence: Every time player wants to save the game

Miscellaneous:

Use Case 3: Build(Setup) the Game

Use Case Name: Build(Setup) the Game

Scope: Game*

Level: user-goal

Primary Actor: Player

Stakeholders and Interests: Player / wants to load or create a layout

Preconditions: Game has been started.

Success Guarantee: A functional layout is shown and the game is ready to start.

Main Success Scenario:

1. "Create a new layout" and "Load a layout" buttons are shown to the user.
2. User clicks on the "create a new layout" button.
3. Obstacles are given to the user to edit on an empty screen.
4. User creates the layout with provided obstacles.
5. User saves the layout.
6. Relevant information is saved to the database.
7. Game has started with the created layout.

Extensions:

- 2.a. Users click on the "Load a layout" button.
- 2.b. A pop-up window for file selection shown to the user.
- 2.c User selects a layout.
- 2.d Layout is successfully loaded.
- 2.e Game has started with the loaded layout.
- 4.a User didn't satisfy the minimum requirements of the layout.
- 4.b A pop-up shown to warn the user about the requirements.

Special Requirements:

Technology and Data Variations List: There should be a database for either saving or loading a layout.

Frequency of Occurrence: Every time player opens the game.

Miscellaneous:

Use Case 4: Load the Game

Use Case Name: Load the Game

Scope: Game

Level: user-goal

Primary Actor: Player

Stakeholders and Interests: Player wants to start playing the saved game

Preconditions: The saved file should exist, Game should be open

Success Guarantee: Game is successfully loaded. Player can start playing

Main Success Scenario:

1. Player opens the game
2. Player successfully loads the game,
3. Player starts playing the saved game when loaded.

Extensions:

- 2.a. The game is not saved and the player cant load the game
- 2.b. A pop-up is shown to the player

Special Requirements:Non

Technology and Data Variations List: The player should use a keyboard and/or a mouse to select the game and navigate to it

Frequency of Occurrence:Whenever the game is opened or restarted

Miscellaneous:

Use Case 5: Pause the Game

Use Case Name: Pause the game

Scope: Game

Level: user-goal

Primary Actor: Player

Stakeholders and Interests: Player wants to pause the running game.

Preconditions: Game starting successfully, The game should not be paused.

Success Guarantee: The game is paused. Nothing is moving and the game objects are not interactable.

Main Success Scenario:

1. Player presses the pause button.
2. Game state is updated to non-interactable.
3. A pop-up is shown displaying the message: "The game is paused." and has a resume button and a save button.

Extensions:

Special Requirements:

Technology and Data Variations List:

1. The button click is done by a mouse or pressing 'p' on keyboard

Frequency of Occurrence: Whenever the Player clicks the pause button

Miscellaneous:

Use Case 6: Resume the Game

Use Case Name: Resume the game

Scope: Game

Level: user-goal

Primary Actor: Player

Stakeholders and Interests: Player wants to resume the game after the game being paused

Preconditions: The game has been paused.

Success Guarantee: The game continues from the point the player paused the game.

Main Success Scenario:

1. Player presses the Resume button which is under the "Paused" text when the game is paused
2. The Paused pop-up will disappear from the screen
3. The game will continue from the point the game was paused

Extensions: None

Special Requirements: None

Technology and Data Variations List: The player should click the Resume button using a mouse.

Frequency of Occurrence: Whenever the Resume button is pressed

Miscellaneous: None

Use Case 7: Move the Phantasm

Use Case Name: Move the Phantasm

Scope: Game

Level: Subfunction

Primary Actor: Player

Stakeholders and Interests: Player wants to move the Phantasm horizontally in between the boundaries of the screen.

Preconditions: The game has started.

Success Guarantee: The Phantasm is moved from left to right on one line which is close to the bottom of the game screen.

Main Success Scenario:

1. The Player presses left or right arrow keys to move the Phantasm.
2. The Phantasm is moved in the direction of the arrow key either until the key is released or the Phantasm has reached the boundary of the game screen.

Extensions: None

Special Requirements: None

Technology and Data Variations List: The player should use the left or right arrow keys on the keyboard of the computer/laptop.

Frequency of Occurrence: Whenever the player presses the arrow keys.

Miscellaneous: None

Use Case 8: Rotate the Phantasm

Use Case Name: Rotate the Phantasm

Scope: Game

Level: Subfunction

Primary Actor: Player

Stakeholders and Interests: Player wants to rotate the Phantasm

Preconditions: The game has started. The game is resuming.

Success Guarantee: The Phantasm is rotated in the intended direction and angle.

Main Success Scenario:

1. The Player presses A or D buttons.
2. When the A button is clicked, the Phantasm starts to rotate until it reaches 45 degrees, if the D button is being pressed the Phantasm starts to rotate until it reaches 135 degrees.
3. The user releases the key.
4. The Phantasm goes back to its original state, with the same rotation speed.

Extensions: None

Special Requirements: None

Technology and Data Variations List: The player should use the A or D keys on the keyboard of the computer/laptop.

Frequency of Occurrence: Whenever the player presses the A/D keys.

Miscellaneous: None

Use Case 9: Hit the Simple Obstacle (Wall Maria)

Use Case Name: Hit the Simple Obstacle (Wall Maria)

Scope: Game

Level: Subfunction

Primary Actor: Enchanted Sphere which bounces off the Phantasm

Stakeholders and Interests:

Preconditions: The game is running and the Enchanted Sphere is moving towards the obstacle.

Success Guarantee: The Enchanted Sphere hits Wall Maria.

Main Success Scenario:

1. Enchanted Sphere bounces off of the Phantasm.
2. Enchanted Sphere hits Wall Maria.
3. Wall Maria obstacle gets destroyed because it is the type of obstacle that can be broken with one hit only.
4. The obstacle disappears from the game scene.
5. Enchanted Sphere bounces back from the obstacle right before it gets destroyed.

Extensions: None

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Often (depends on the number of the simple obstacles)

Miscellaneous: None

Use Case 10: Hit the Firm Obstacle (Steins Gate)

Use Case Name: Hit the Firm Obstacle (Steins Gate)

Scope: Game

Level: Subfunction

Primary Actor: Enchanted Sphere which bounces off the Phantasm

Stakeholders and Interests:

Preconditions: The game is running and the Enchanted Sphere is moving towards the obstacle.

Success Guarantee: The Enchanted Sphere hits Steins Gate.

Main Success Scenario:

1. Enchanted Sphere bounces off of the Phantasm.
2. Enchanted Sphere hits Steins Gate.
3. The number on the obstacle decreases by 1.
4. Enchanted Sphere bounces back from the obstacle.

Extensions:

3a. The obstacle has a number bigger than 1. Therefore, it doesn't get destroyed when the Enchanted Sphere hits it.

1. The durability of the obstacle will be decreased by 1.

3b. The obstacle has a durability of 1.

1. The obstacle gets destroyed.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Sometimes (depends on the number of the firm obstacles)

Miscellaneous: None

Use Case 11: Hit the Explosive Obstacle (Pandora's Box)

Use Case Name: Hit the Explosive Obstacle (Pandora's Box)

Scope: Game

Level: Subfunction

Primary Actor: Enchanted Sphere which bounces off the Phantasm

Stakeholders and Interests:

Preconditions: The game is running and the Enchanted Sphere is moving towards the obstacle.

Success Guarantee: The Enchanted Sphere hits Pandora's Box.

Main Success Scenario:

1. Enchanted Sphere bounces off of the Phantasm.
2. Enchanted Sphere hits the obstacle
3. Obstacle explodes and its remains fall off.

Extensions:

- 3a. Remains of the obstacle hit the Noble Phantasm,
 - 3a.1 The Player loses one chance.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Rarely (depends on the number of the explosive obstacles)

Miscellaneous: None

Use Case 12: Hit the Gift Box (Gift of Uranus)

Use Case Name: Hit the Gift Box (Gift of Uranus)

Scope: Game

Level: Subfunction

Primary Actor: Enchanted Sphere which bounces off the Phantasm

Stakeholders and Interests:**Preconditions:** The game is running and the Enchanted Sphere is moving towards the obstacle.

Success Guarantee: The Enchanted Sphere hits Gift Box

Main Success Scenario:

1. Enchanted Sphere bounces off of the Phantasm.
2. Enchanted Sphere hits the obstacle
3. Obstacle gets destroyed
4. The destroyed obstacle drops a box downwards towards the noble phantasm.
5. The Noble Phantasm touches the box

6. The box opens and rewards the warrior with a magical ability that can be either used to support the warrior, or to create more challenges and obstacles for the other player.

Extensions:

5a. The Noble Phantasm does not touch the box.

5a.1 Nothing happens

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Sometimes (depends on the number of the gift box)

Miscellaneous: None

Use Case 13: Equip Magical Hex

Use Case Name: Equip Magical Hex

Scope: Game

Level: Subfunction

Primary Actor: Player

Stakeholders and Interests: Player wants to fire magical hexes that hit obstacles

Preconditions: The existence of noble phantasm

Success Guarantee: The noble phantasm is equipped with the magical hex to be able to fire magical hexes

Main Success Scenario:

1. The magical hex is equipped when the player presses H or clicks on the icon on the screen
2. The player shoots the hexes by pressing S on keyboard and if the noble phantasm rotates, the magical canons also rotate
3. The magical hex is functioning for 30 seconds before it disappears

Extensions:

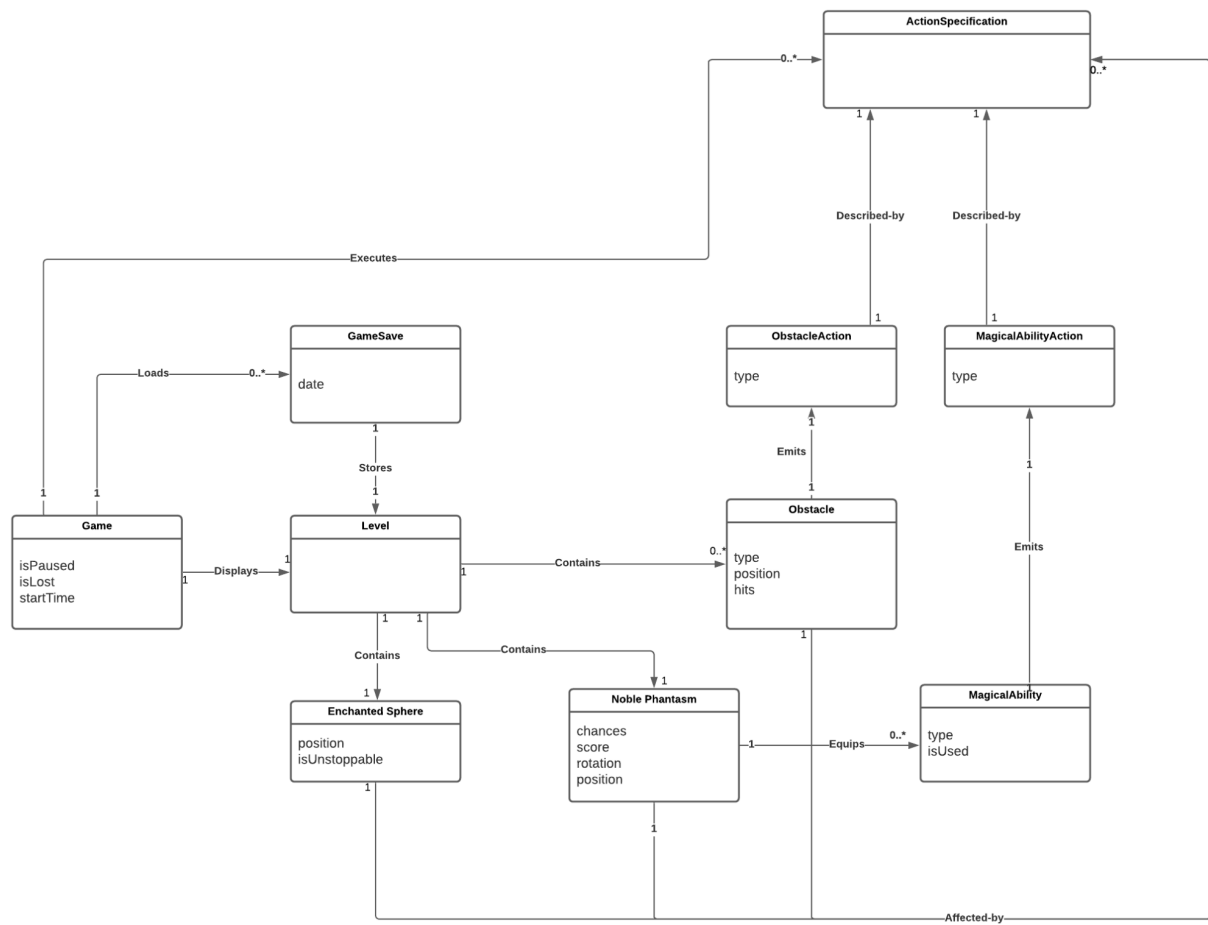
Special Requirements: Non

Technology and Data Variations List: The player should press H or use a mouse to click the icon on the screen

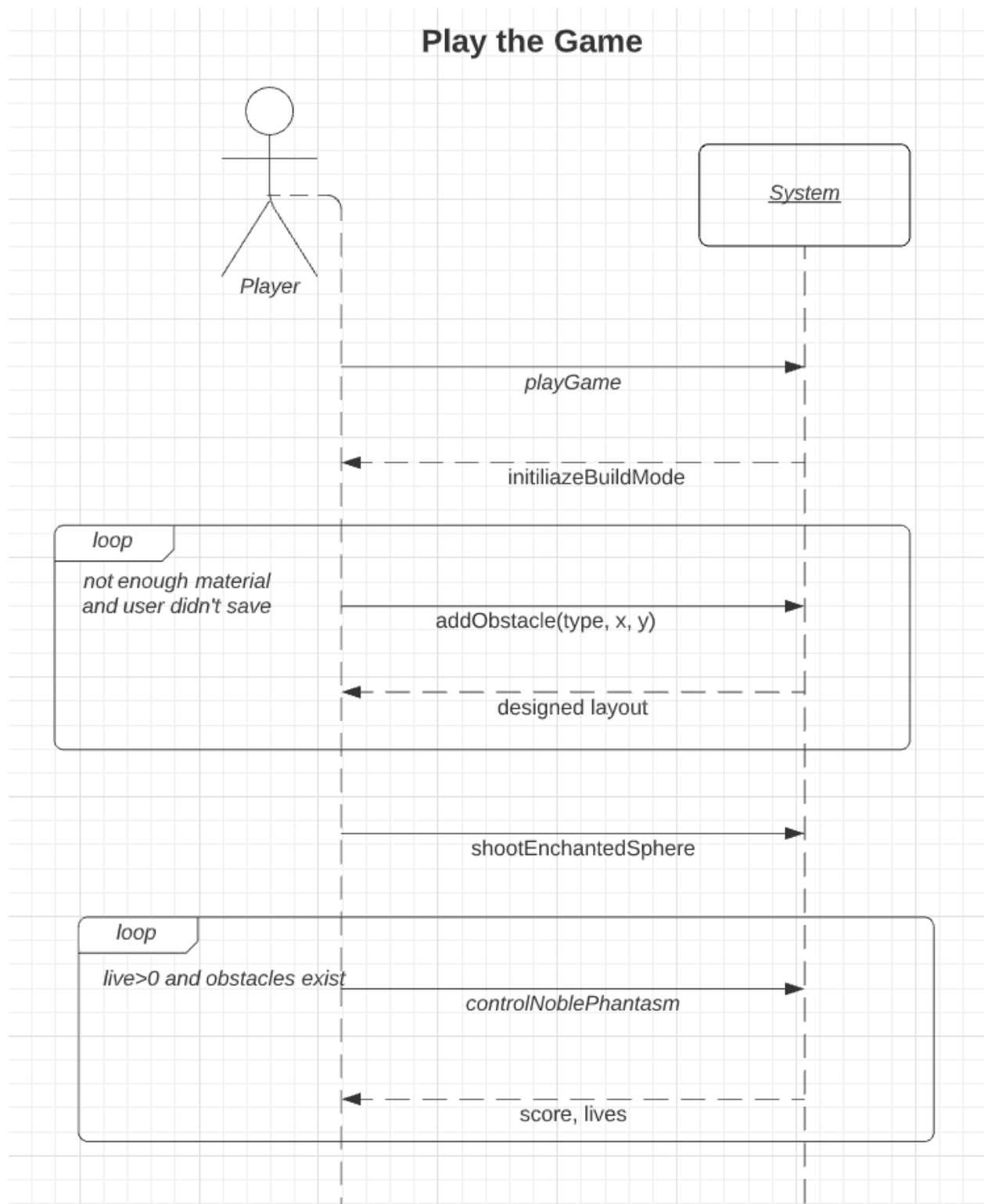
Frequency of Occurrence: Rarely

Miscellaneous:

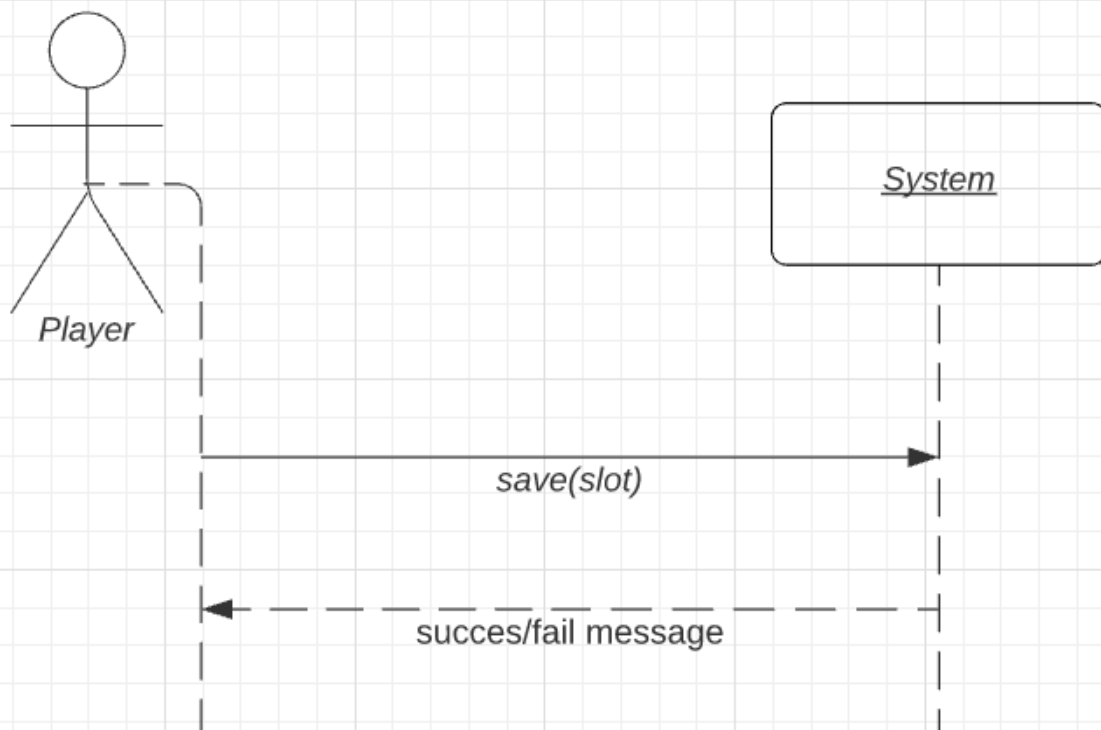
Domain Classes



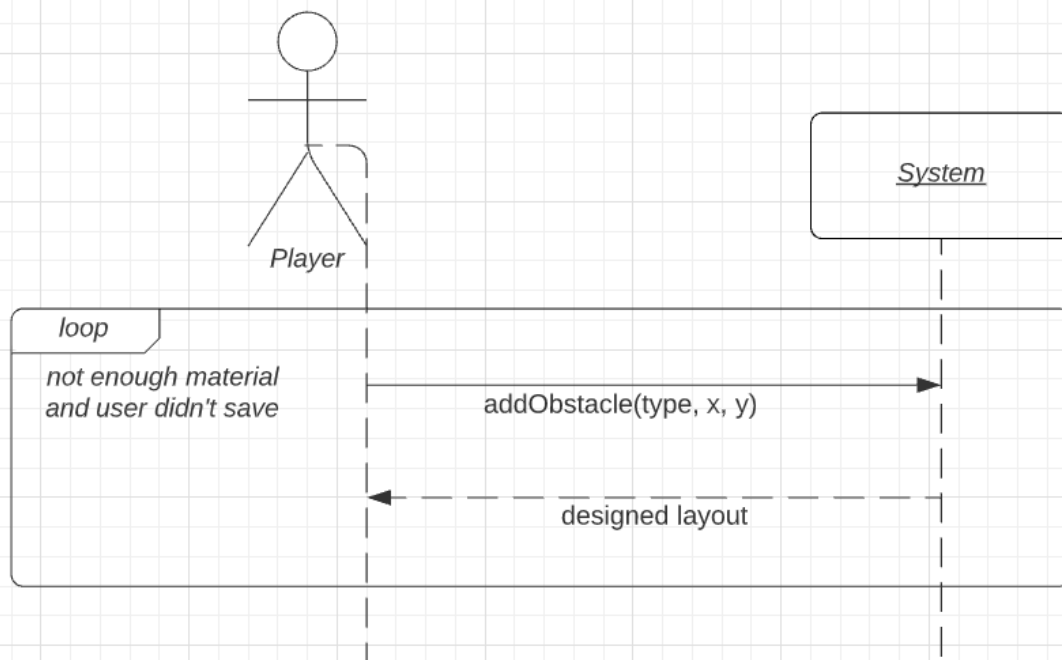
System Sequence Diagrams



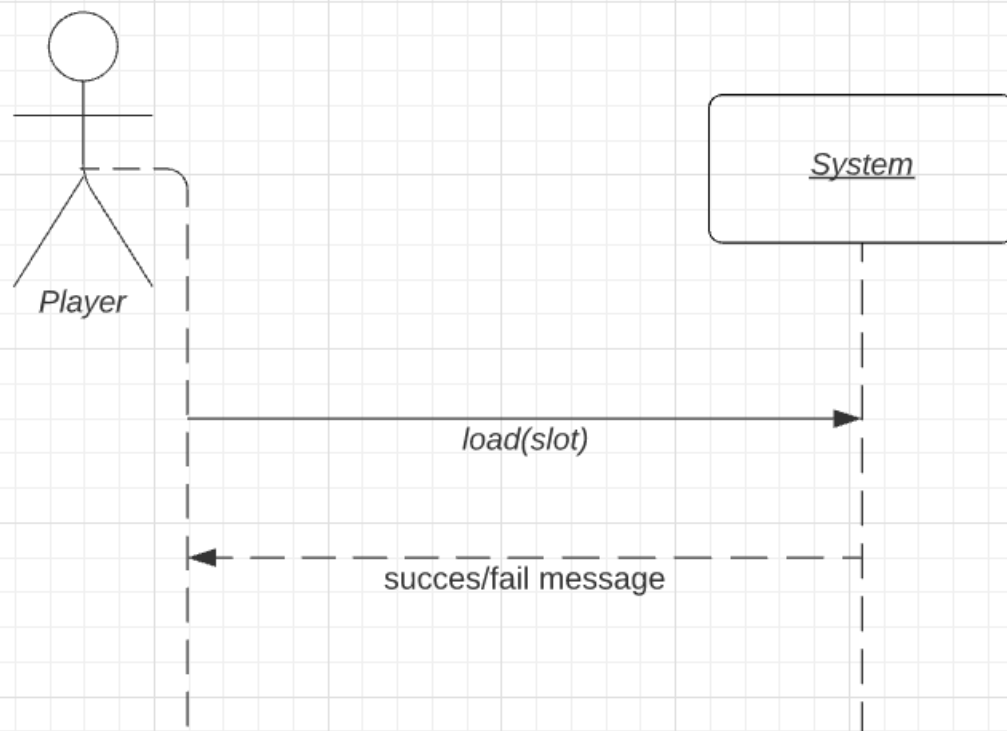
Save the Game



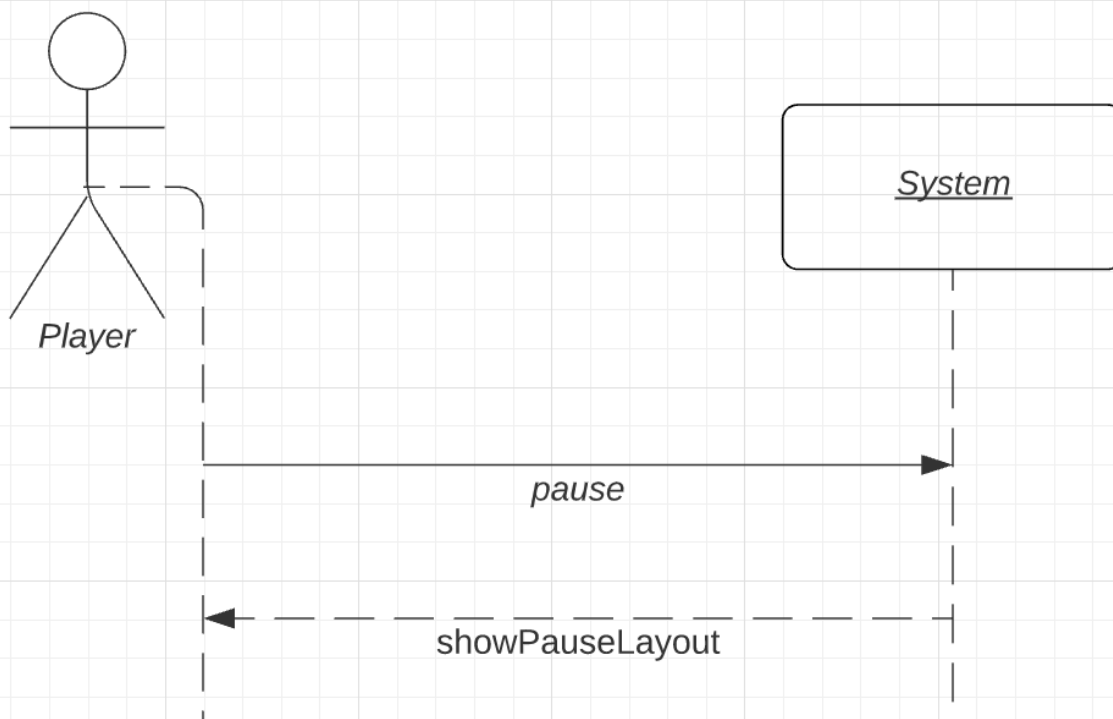
Build the Game



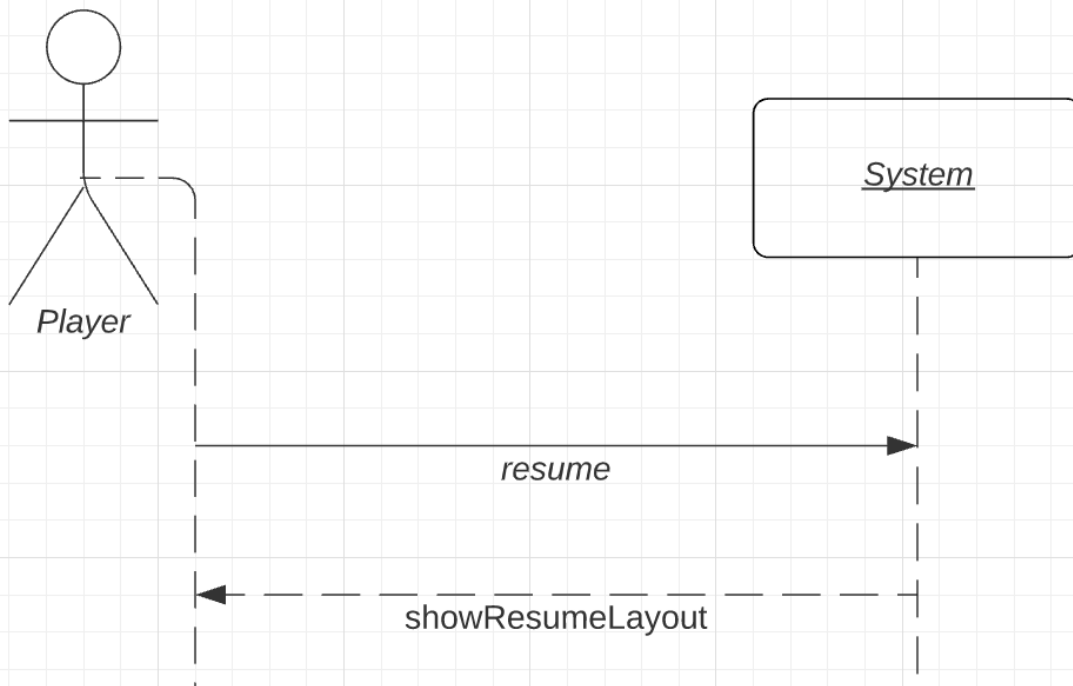
Load the Game



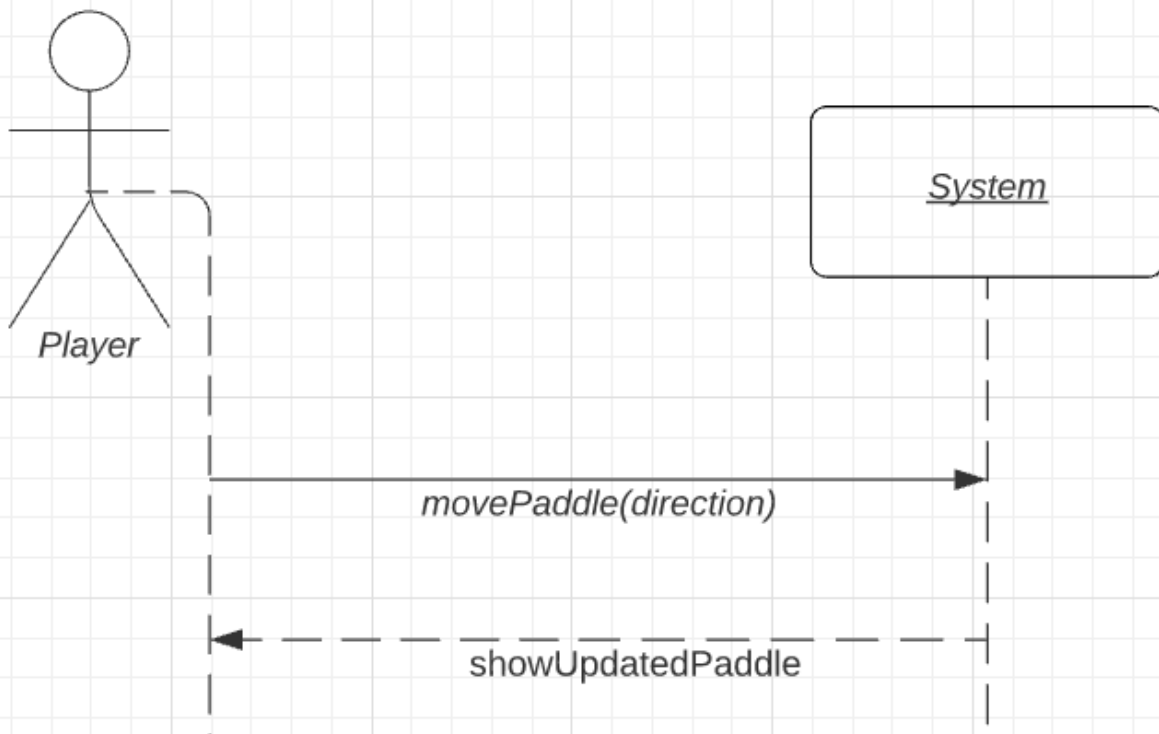
Pause the Game



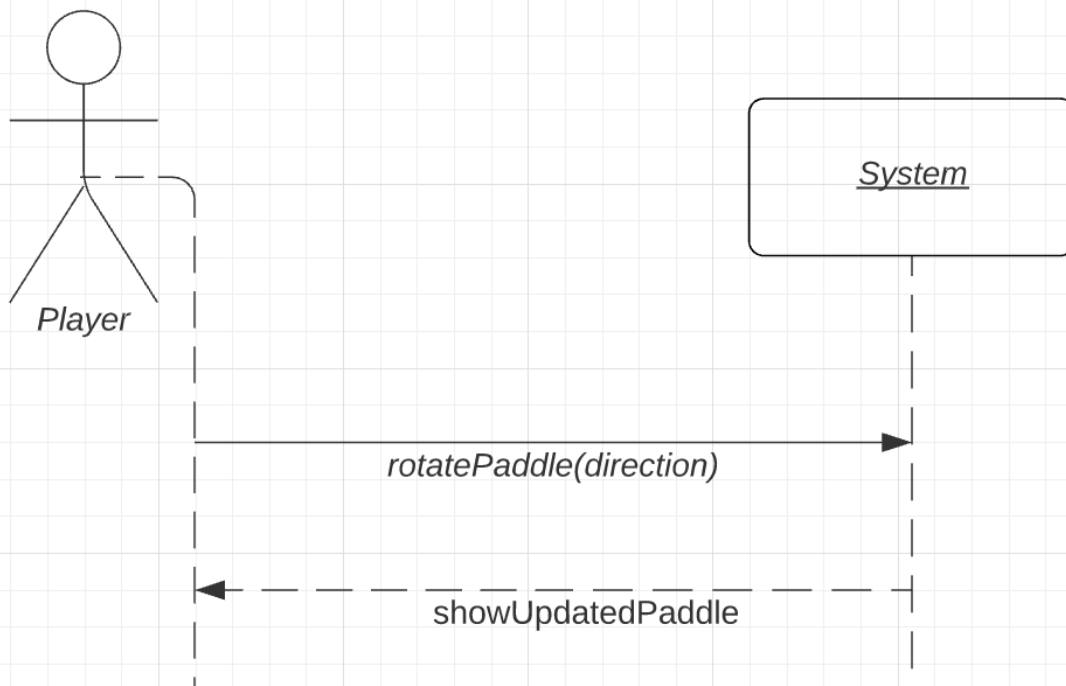
Resume the Game



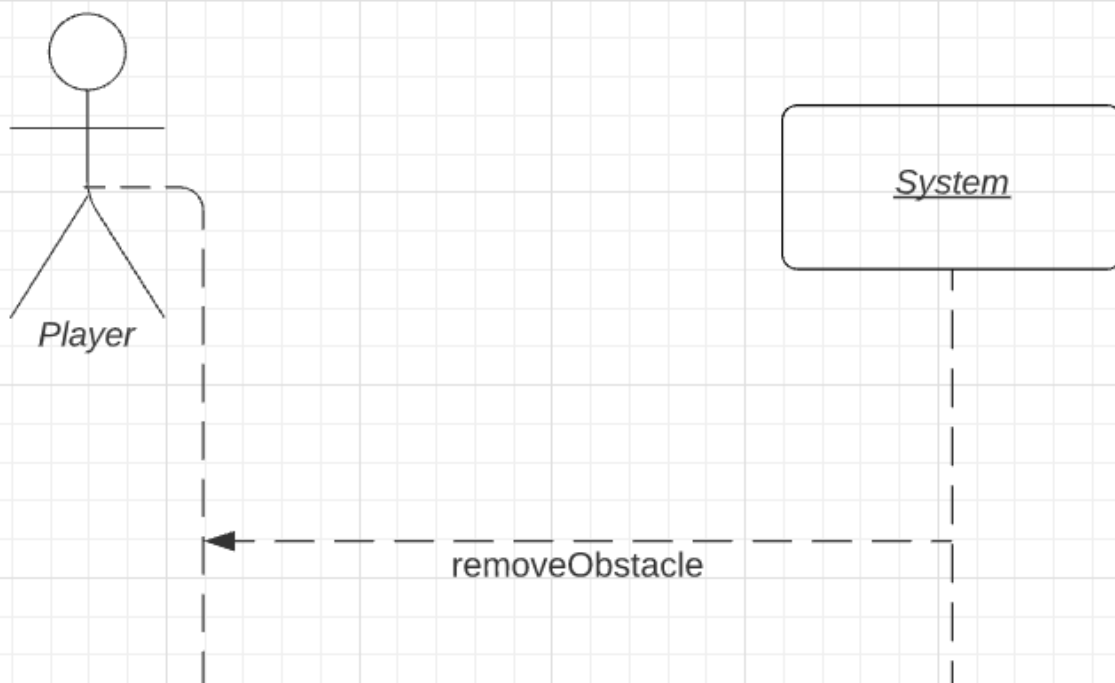
Move the Phantasm



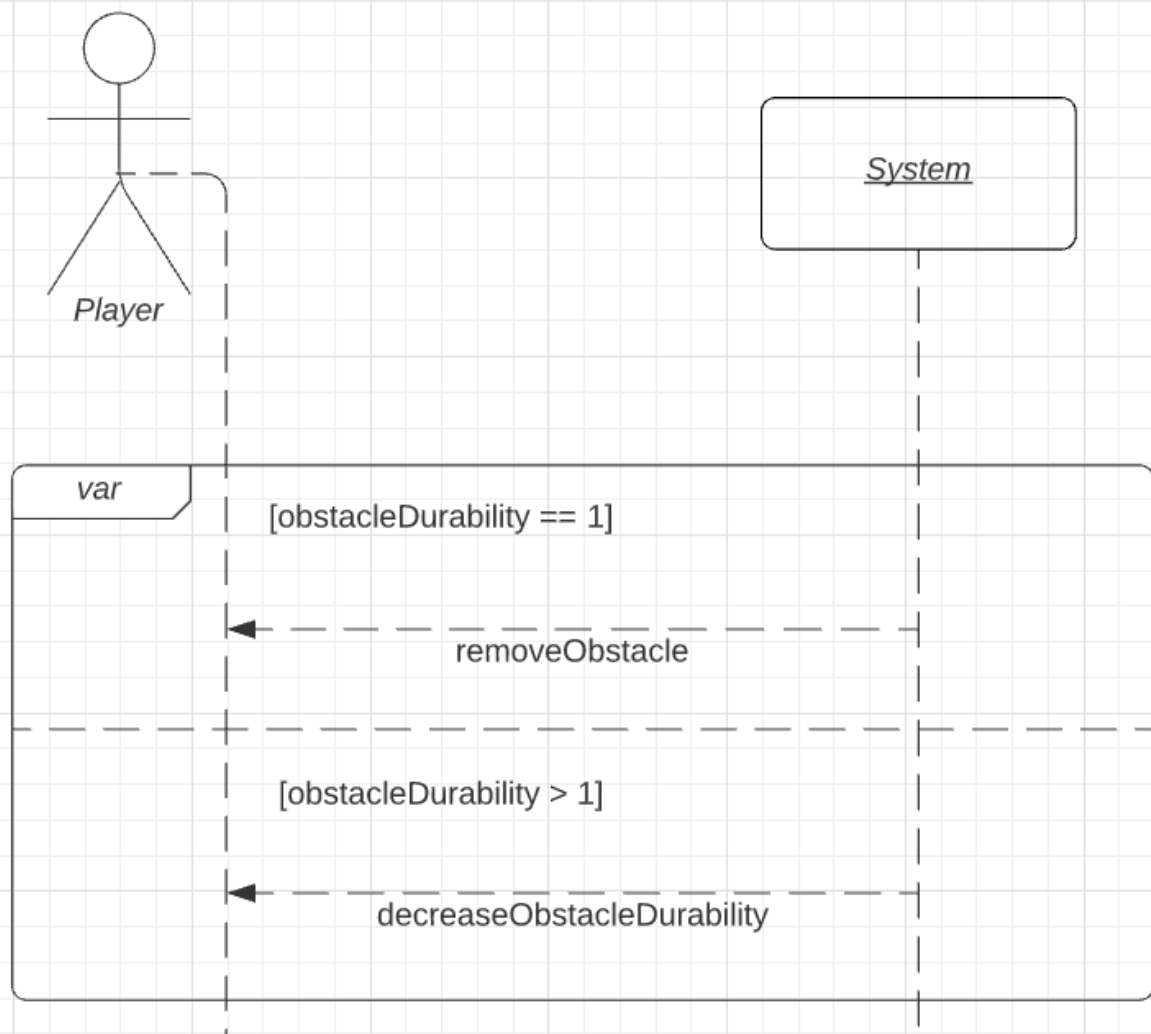
Rotate the Phantasm



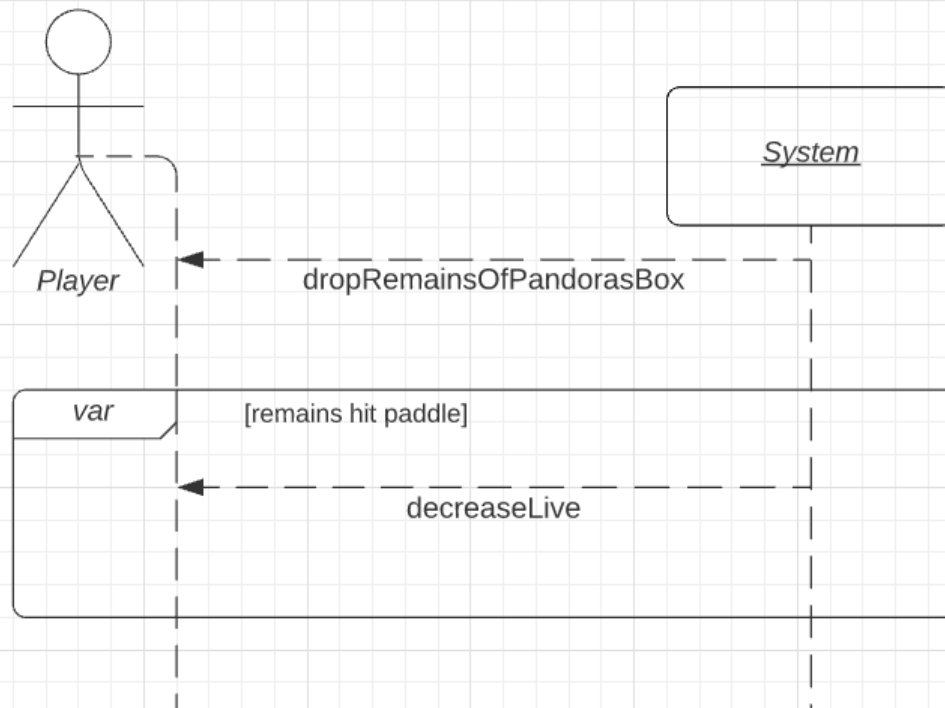
Hit a simple obstacle (Wall Maria)



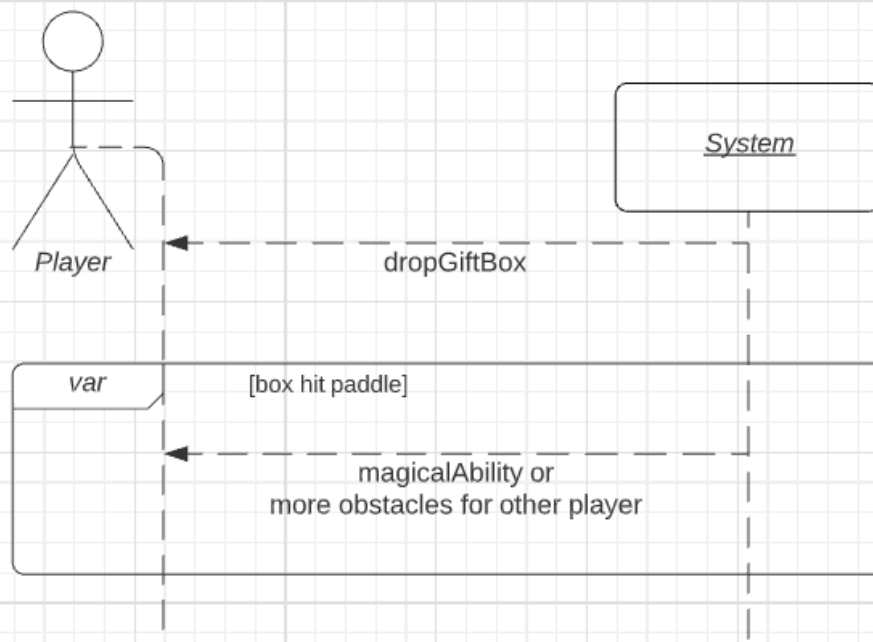
Hit a firm obstacle (Steins Gate)



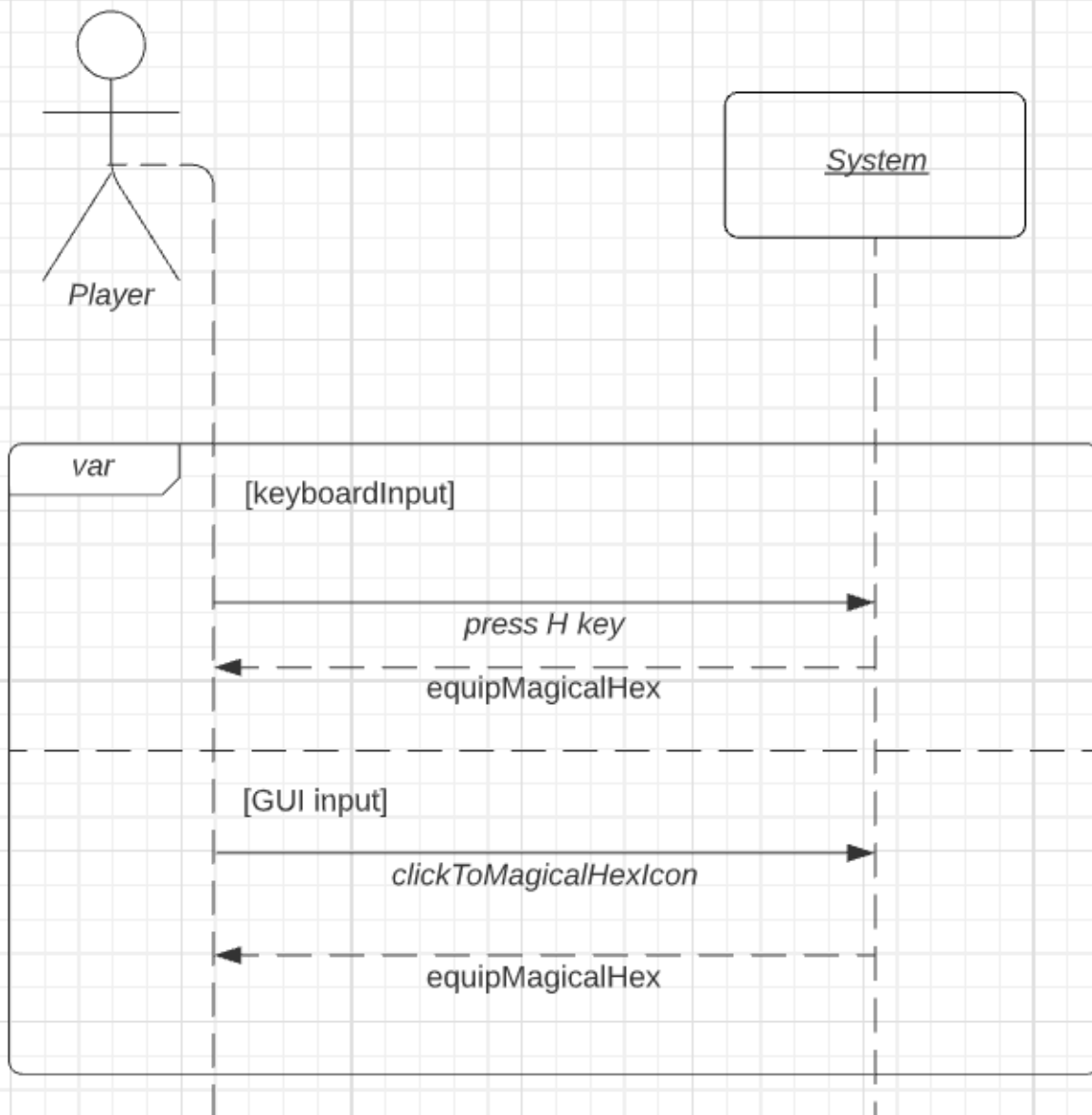
Hit an explosive obstacle (Pandora's Box)



Hit a gift box (Gift of Uranus)



Equip Magical Hex



Operation Contracts:

Contract CO1: Open Game:

Operation:	openGame()
Cross References:	Use Case 1: Play the Game Use Case
Preconditions:	Existence of the executable
Postconditions:	Game window was opened

Contract CO2: Play Game:

Operation:	playGame()
Cross References:	Use Case 1: Play the Game Use Case Use Case 3: Build(Setup) the Game Use Case 4: Load the Game
Preconditions:	Game window is open
Postconditions:	An instance of the Game was created with the desired settings of the player. addObstacles (type, x, y) was called. An instance of the Noble Phantasm and Enchanted Sphere was created.

Contract CO3: Setup Game:

Operation:	setupGame(type, x, y)
Cross References:	Use Case 1: Play the Game Use Case 3: Build(Setup) the Game Use Case 4: Load the Game
Preconditions:	Game window is open in build mode

Postconditions: An instance of the Game was created with the desired settings of the player: game.simpleObstacles was set to simpleObstacles, game.firmObstacles was set to firmObstacles, game.explosiveObstacles was set to explosiveObstacles and game.giftObstacles was set to giftObstacles.

Contract CO4: Shoot:

Operation: shoot()

Cross References: Use Case 7: Moving the Phantasm
Use Case 8: Rotating the Phantasm
Use Case 9: Hitting the Simple Obstacle (Wall Maria)
Use Case 10: Hitting the Firm Obstacle (Steins Gate)
Use Case 11: Hitting the Explosive Obstacle (Pandora's Box)
Use Case 12: Hitting the Gift Box (Gift of Uranus)

Preconditions: Noble Phantasm and Enchanted Sphere are present in the Game Window
Game is in the running mode

Postconditions: enchantedSphere.location = location

Contract CO5: Pick Magical Ability:

Operation: pickMagicalAbility(type)

Cross References: Use Case 9: Hitting the Simple Obstacle (Wall Maria)
Use Case 10: Hitting the Firm Obstacle (Steins Gate)
Use Case 11: Hitting the Explosive Obstacle (Pandora's Box)
Use Case 12: Hitting the Gift Box (Gift of Uranus)

Preconditions: A list of Magical Abilities are available on the game window

Postconditions: Depending on the type of the magical ability, an instance of that type was created and the type of the object was set to type
magicalAbility.type = type

Contract CO6: Move Paddle:

Operation: movePaddle(direction)

Cross References: Use Case 1: Play the Game
Use Case 7: Move the Phantasm
Use Case 9: Hit the Simple Obstacle
Use Case 10: Hit the Firm Obstacle
Use Case 11: Hit the Explosive Obstacle
Use Case 12: Hit the Gift Box
Use Case 13: Equip Magical Hex

Preconditions: Game is started
Paddle can be moved in the specified direction

Postconditions: paddle.location was updated in accordance with
paddle.speed and direction given

Contract CO7: Rotate Paddle:

Operation: rotatePaddle(direction)

Cross References: Use Case 1: Play the Game
Use Case 8: Rotate the Phantasm
Use Case 9: Hit the Simple Obstacle
Use Case 10: Hit the Firm Obstacle
Use Case 11: Hit the Explosive Obstacle
Use Case 12: Hit the Gift Box

Preconditions: Game is started

Postconditions: paddle.angle was updated in accordance with
paddle.rotationSpeed and rotation given

Contract CO8: Pause Game:

Operation: pauseGame()

Cross References: Use Case 1: Play the Game
Use Case 5: Pause the Game

Preconditions: Game is started

Postconditions: game.Status was updated to Pause

Contract CO9: Resume Game:

Operation:	resumeGame()
Cross References:	Use Case 1: Play the Game Use Case 5: Pause the Game Use Case 6: Resume the Game
Preconditions:	Game is paused
Postconditions:	game.Status was updated to Running

Contract CO10: Save Game:

Operation:	save(slot)
Cross References:	Use Case 1: Play the Game Use Case 2: Save the Game Use Case 5: Pause the Game
Preconditions:	Game is paused Connection to database is established
Postconditions:	Needed information was written into the database

Contract CO11: Load Game:

Operation:	load(slot)
Cross References:	Use Case 1: Play the Game Use Case 3: Build(Setup) the Game Use Case 4: Load the Game
Preconditions:	Game is started and in the build mode There exists a pre saved game
Postconditions:	game.Layout was set up in accordance with the pre-saved conditions.

Supplementary Specification

Revision History

Version	Date	Description	Author
Inception Draft	Oct. 29, 2021	First Draft. Will be defined primarily during elaboration.	SyntacticSugar

Introduction:

This document is the repository for all Need For Spear requirements not captured in the use case narratives.

Functionality:

Logging and error handling: Log all errors to persistent storage.

Environment: The game should be platform independent.

Capacity: Single player.

Pluggable Rules:

Usability:

Human Factors: Players play the game on computer screen

1. All game components can be seen from 1 meter
2. Colors should be appropriate both for differentiating objects and following the ball easily.
3. Avoid colors related with color blindness
4. Statistics and equipped features should be easily seen and followed

Reliability:

Recoverability:

If there is a failure due to the hardware or connection, the player may try to save the game and reload the game / restart the computer.

Performance:

The goal of our project is to present a game that can be completed, saved or loaded without any errors. Our aim is to build a game which can easily be played and that provides quick responses for all user inputs. Therefore, we will aim to achieve a simple but efficient UI for this game project.

Supportability:

Adaptability: It should work on the recent versions of supported OSs.

Purchased Components:

None.

Implementation Constraints:

Software will be built over JAVA using JAVA libraries.

Interfaces:

Hardware Interfaces: None.

Software Interfaces:

Vision

Introduction:

Need for Spear is an enhanced and computer-based version of a very popular arcade game called Breakout. It consists of a customizable game set up, different magical features. It serves its players a good enjoyable time with the opportunity of playing solo or with a friend.

Positioning:

Business Opportunity:

Different versions of the game are already available in the market. Different versions present the same brick breaking idea with different features. Since it is a well-known game, people will be interested if better graphics and different features are introduced.

Problem Statement:

Versions that are available on the market are popular and widely played. However, playing the same game with the same features becomes boring. Thus, a new version of the game with new changes are needed.

Product Position Statement:

The Need for Spear is for people who like to play games in any age following the logic of an Atari game with modern graphics and features like 2-player compiate.

Alternatives and Competition:

There are many versions of the game in the market(especially for Android& iOS), however it is not really common for computer games.

Glossary

Term	Definition and Information	Format	Validation Rules	Aliases
Warrior	The user			
Enchanted Sphere	The ball that player directs using paddle to hit obstacles	Graphical Object		Ball
L	Noble Phantasm's length	A rational number that is equal to $10\% \times \text{Screen width}$		
Noble Phantasm	The paddle that user moves to catch the enchanted sphere and to direct the enchanted sphere to hit obstacles. Can be moved horizontally and be rotated	Graphical Object Length : L Thickness: 20px Speed: <i>If the left or right arrow is pressed and released: The noble phantasm should move by an offset equal to $L/2$ with a speed of L/second. If the button is down, it should move with the speed of $2 \times L/\text{second}$.</i>		Paddle
Wall Maria	One of 4 types of obstacles. Can be destroyed and disappear after one hit	Graphical object Speed:		
Steins Gate	One of 4 types of obstacles. Has specific number on it Can be broken and disappear after number of specified hits.	Graphical object		

Pandora's Box	One of 4 types of obstacles. Explodes after a hit and starts to fall towards Noble Phantasm. If it hits the paddle, user loses a chance	Graphical Object		
Gift of Uranus	One of 4 types of obstacles. Can be broken and disappear after one hit. After it is destroyed, drops a box towards the paddle. If paddle catch the box, box opens and a magical ability is earned	Graphical Object		
Score	The numerical value that will indicate the total of the points which the player earns throughout the game.	$\text{NewScore} = \text{OldScore} + 300 / (\text{CurrentTime} - \text{GameStartingTime})$		
Health	The numerical value of the remaining lives which the player has during gameplay.	Number		Lives, chance
Chance Giving Ability (Magical Ability)	A powerup which appears on the gameplay screen. It increases the player's chances by 1.	Graphical Object		
Noble Phantasm Expansion (Magical Ability)	A powerup which appears on the gameplay screen and that can be collected by the player to be used directly or kept for later use. It doubles the length of the Noble Phantasm.	Graphical Object	Once activated, it only lasts for 30 seconds.	T

Magical Hex (Magical Ability)	A powerup which appears on the gameplay screen and that can be collected by the player to be used directly or kept for later use. It adds two magical canons on both ends of the Noble Phantasm.	Graphical Object	Once activated, it only lasts for 30 seconds.	H
Unstoppable Enchanted Sphere (Magical Ability)	A powerup which appears on the gameplay screen. It makes the Enchanted Sphere so powerful that it breaks all kinds of obstacles and passsews through them.	Graphical Object	Once activated, it only lasts for 30 seconds.	