COMP-302 TERM PROJECT PHASE 1 - R1M1

Group Name: Brogrammers
Date: 31.10.2021

TABLE OF CONTENTS

Use Cases

USE CASE 1: Register

Actor: User

Pre Condition: User successfully opened the game.
Post Condition: User is successfully directed to the main opening page.

Happy Scenario:

1. System asks the user to provide a username and password.
2. User types a username and password and submits.
3. System connects to the database.
4. System adds the username and password to the database.
5. System opens the main page.

Extensions (or Alternative Flows):

2a. User provided missing information:
    1. System displays “Please write both username and password” and stays in the register page.
3a. There is a failure with connection to the database:
    1. System displays “An error occurred, please try again” and stays in the login page.
4a. Username exists in the database:
    1. System displays “Username is already taken. Try a different one.” and stays in the register page.

USE CASE 2: Log In

Actor: User

Pre Condition: User successfully opened the game.
Post Condition: User is successfully directed to the main opening page.

Happy Scenario:

1. System asks the user to provide a username and password.
2. User types his username and password and submits.
3. System connects to the database.
4. System checks whether a user exists with the provided information in database.
5. System opens the main page.

Extensions (or Alternative Flows):

2a. User provided missing information:
    1. System displays “Please write both username and password” and stays in the log in page.
3a. There is a failure with connection to the database:

1. System displays “An error occurred, please try again” and stays in the login page.

4a. User does not exist in the database:

1. System displays “Username or password is invalid, try again.” and stays in the login page.

USE CASE 3: Customize Game

Actor: User

Pre Condition: Application has launched.

Happy Scenario:

1. User clicks the menu button which is below the start the game button.
2. System shows some options: Sound, Theme, Guide.
3. User selects an option.

Extensions (or Alternative Flows):

3a. User selects the sound button:

1. Sound settings opens.
2. User increases or decreases the sound.

3b. User selects the theme button:

1. Theme settings opens.
2. User selects a desired theme for the game.

3c. User selects the guide button:

1. A guide file pops up which includes the basics of how to play the game.

USE CASE 4: Create the Game Environment

Actor: User

Pre Condition: User successfully logged in.
Post Condition: Running mode is opened with the new game environment.

Happy Scenario:

1. User presses the “Create New World” button.
2. System asks the user to enter the number of each obstacle he/she wants.
3. User specifies the number of each type of the obstacles.
4. The system puts obstacles to arbitrary positions.
5. The user changes the positions of some of the obstacles by clicking the mouse.
6. World is created. The user can save the game or can start to play.

Extensions (or Alternative Flows):

1a. There is no empty space in the database:

1. No new game can be created before deleting a created one.

3a. User does not enter an integer (user entered double, float or string) in the place where the user is supposed to specify the number of each obstacle type:

1. System displays “An error occurred, please try again” and opens the main page.

3b. One of the obstacle numbers is not adequately specified by the user:

1. The number is positive but less than the minimum requirement. System displays “Minimum criteria numbers for the obstacles is not met” and asks the user to change the number until the minimum number criteria is reached.
2. The number is negative. System displays “Invalid obstacle number is entered” error message and asks the user to change the number until the number is valid.

4a. System cannot put obstacle to the world since there is not enough space in the editing area for the user:

1. System displays “There is not enough space, please decrease the number of obstacles.” and asks the user to change the number until the obstacles fits to the area.

5a. User puts one obstacle on top of another:

1. System displays “Obstacles cannot be overlapped.” and puts the obstacle to its old position.

USE CASE 5: Load Game

Actor: User

Pre Condition: User successfully logged in.
Post Condition: Running mode is opened with the saved game.

Happy Scenario:

1. User presses the “Load” button.
2. System connects to the database.
3. System pulls the information of the saved game (by the current user) from the database.
4. System creates the world by the data it got from the database.

Extensions (or Alternative Flows):

2a. There is a failure with connection to the database:

1. System displays “An error occurred, please try again” and opens the main page (the page which is directly opened after log in).

3a. There is a failure with pulling data from the database:

1. System displays “An error occurred, please try again” and opens the main page.

3b. There is not any game saved by the user:

1. System displays “You do not have any saved game.” and opens the main page.

USE CASE 6: Save Game

Actor: User

Pre-condition: The game is successfully started.

Happy Scenario:

1. User presses the save button.
2. System connects to the database.
3. System records the current data (number of objects, positions etc.) to the database.

Extensions (or Alternative Flows):

1a. User did not pause:
    1. Warning "you should pause first" appears.
2a. There is a failure with connection to the database:
    1. System throws an error.

USE CASE 7: Destroy Obstacle

Actor: User

Pre condition: The game is already successfully running.
Post condition: The game continues running.

Happy Scenario:

1. User makes Enchanted Sphere hit the obstacle:
    1a. User makes Enchanted Sphere hit the Simple Obstacle (Wall Maria):
        1. Simple Obstacle is broken.
    1b. User makes Enchanted Sphere hit the Firm Obstacle (Steins Gate):
        1. Firm Obstacle has more than 1 required hit:
            1a. The required hit that Firm Obstacle has is decremented by 1.
            1b. Firm Obstacle could not be broken.
        2. Firm Obstacle has 1 required hit:
            2a. Firm Obstacle is broken.
    1c. User makes Enchanted Sphere hit the Explosive Obstacle (Pandora's Box):
        1. Explosive Obstacle explode and its remains fall down:
            1a. Remains touch the noble phantasm:
                1. The user loses a chance.
            1b. Remains do not touch the noble phantasm:
                1. The user does not lose a chance.
    1d. User makes Enchanted Sphere hit the Gift Obstacle (Gift of Uranus):
        1. Gift Obstacle is broken and drops a box.

USE CASE 8: Catch the Magical Ability

Actor: User

Pre Condition: Gift Obstacle is broken and dropped a box.
Post Condition: Running mode is opened with the saved game.

Happy Scenario:

1. User makes Noble Phantasm touch the dropped box by Gift obstacle.
2. System adds the magical ability that is inside the box to the user's pocket.

Extensions (or Alternative Flows):

1a. User cannot make Noble Phantasm touch the box, box falls down:
    1. User does not get magical ability.

USE CASE 9: Use a Magical Ability

Actor: User

Pre Condition: User has a magical ability in his/her packet.
Post Condition: User has one less magical ability in his/her packet.

Happy Scenario:

1. User selects an ability:
    1a. Warrior selects Chance Giving Ability:
        1. User's chances are increased by 1.
    1b. Warrior selects Noble Phantasm Expansion by pressing the button T, or by pressing the Noble Phantasm Expansion icon on the screen:
        1. The length of the noble phantasm doubles for 30 seconds.
    1c. Warrior selects Magical Hex by pressing the button H, or by pressing the Magical Hex icon on the screen:
        1. Noble phantasm is equipped with two magical canons on both of its ends for 30 seconds.
    1d. Warrior selects Unstoppable Enchanted Sphere:
        1. Noble phantasm becomes much more powerful for 30 seconds.

Extensions (or Alternative Flows):

1a. User does not have that magical ability:
    1. System displays "You don't have that ability.".
1b. The ability is already in action:
    1. The ability doubles.

USE CASE 10: Control the Noble Phantasm

Actor: User

Pre Condition: Game is in the running mode.

Happy Scenario:

1. User pushes a button and releases the button, or does not release the button.
2. System moves Noble Phantasm in that direction.

Extensions (or Alternative Flows):

*a. User does not push a button:
    1. The noble phantasm does not move.
*b. User gives the Noble Phantasm a degree of 45 or 135 with the appropriate keys:
    1. The Noble Phantasm is reorientated.

USE CASE 11: Catch the Enchanted Sphere

Actor: User

Pre-Condition: Game is in the running mode.

Happy Scenario:

1. Enchanted sphere is falling down.
2. System makes Enchanted Sphere hit Noble Phantasm so saves the Enchanted Sphere from falling down.
3. Enchanted sphere goes upward, so the game continues.

Extensions (or Alternative Flows):

2a. Noble phantasm could not catch the enchanted sphere:
    1. The warrior loses 1 chance.
3a. The warrior runs out of chances:
    1. Game ends, the warrior loses the game.

USE CASE 12: Win the Game

Happy Scenario:

1. User hits the last object with the enchanted sphere, and the main objective is done.
2. System displays “Game Over, You Won!”.

USE CASE 13: Quit Game

Actor: User

Pre Condition: Game has successfully started.
Post condition: The game is quitted successfully.

Happy Scenario:

1. User presses the pause button.
2. User selects the quit game button.
3. System asks the user if he wants to save the game or not.
4. After user's selection of the save button, system connects to database and saves the information of the current game.
5. User quits successfully.
6. System opens the login screen.

Extensions (or Alternative Flows):

4a. user selects not to save game:
    1. System displays "Are you sure" and "Yes" or "No" buttons below:
        1a. user presses "Yes":
            1- Game quits immediately and all the progress so far is lost.
        1b. user presses "No":
            1- System goes back to save the game display.

General Game Play Scenario (Not a Use Case):
(We added this because it might be helpful while coding.)

Actor: User

Happy Scenario:

1. User logs in to the game.
2. User hits the start button and the game starts.

Loop:

3. The Noble Phantasm throws the Enchanted Sphere.
4. Enchanted Sphere hits the objects:
    4a. Enchanted Sphere does not hit the object:
        1. The ball keeps moving, and the game goes on.
    4b. When the enchanted sphere hits non-moving object and if it hits a side of this object:
        1. It will bounce and move back in the same speed with the angle equal to and symmetric around the norm of the hit surface.
        2. The reflection of the enchanted sphere hitting the middle of a non-moving object. Here the original angle is 45, hence the enchanted sphere will reflect with an angle of 45.
        3. If it hits a corner of an object, the same rule applies. But the reference here is an imaginary line making an angle of 45 degrees with the corner legs.

5. The Noble Phantasm catches the Enchanted Sphere.

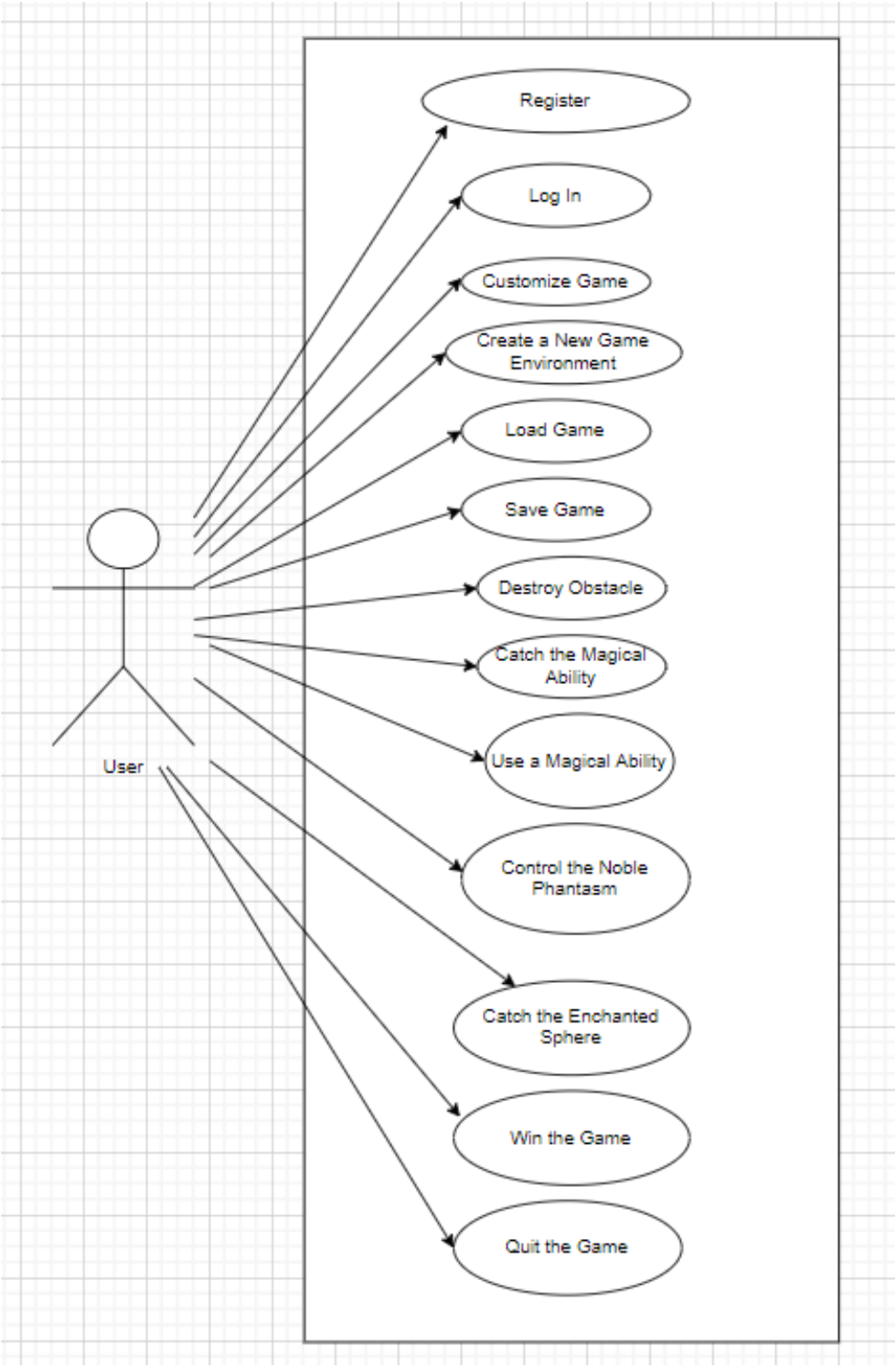
End of the loop:

6. All the objects are destroyed.
7. The user wins and obtains the Spear of Power.

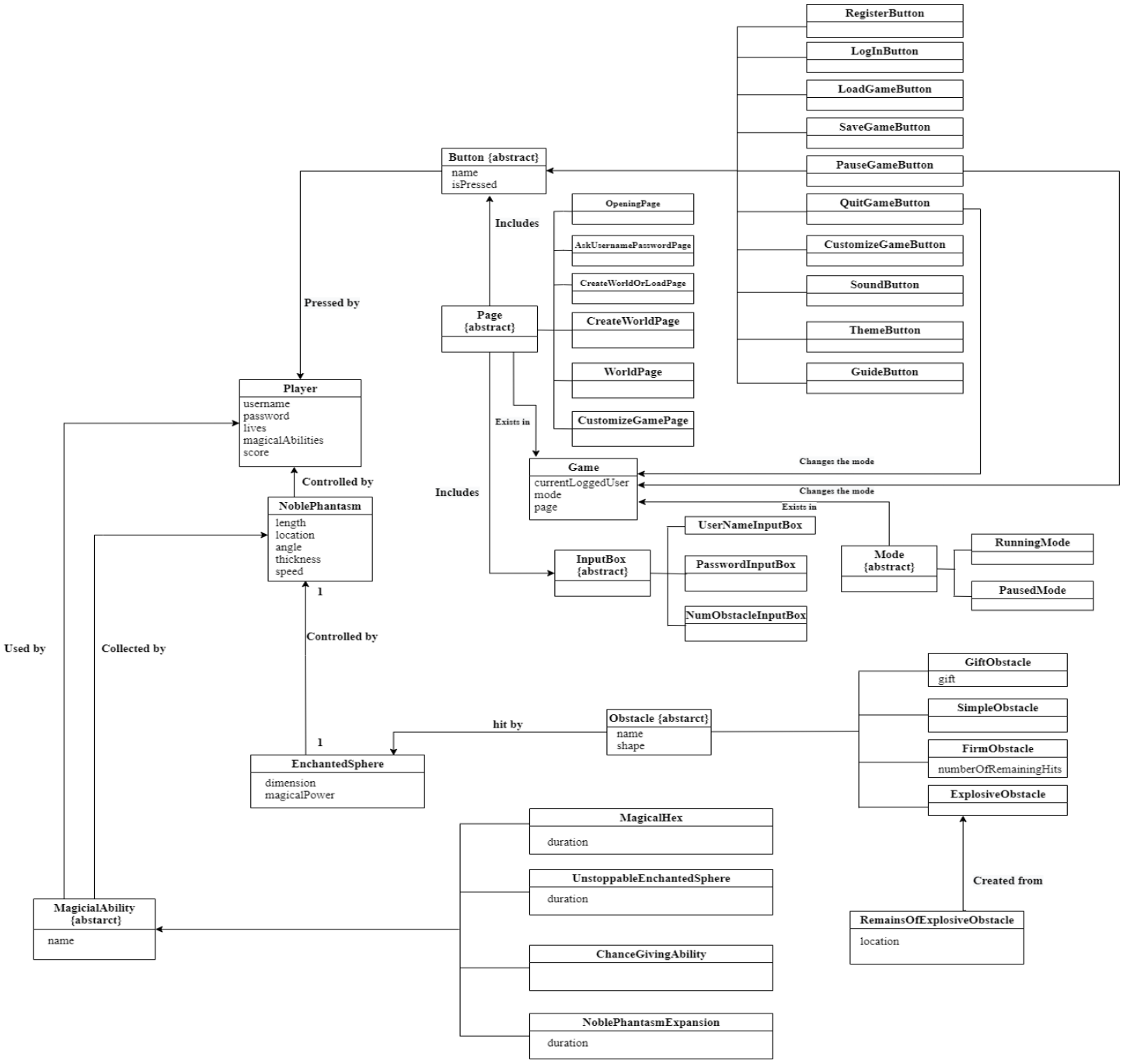
Extensions (or Alternative Flows):

*a. The noble phantasm catches a Magical Ability box:
    1. The magical ability that is inside the box is added to the user's pocket.
5a. The Noble Phantasm could not catch the Enchanted Sphere:
    1. The user has more than 1 live:
        1a. The game goes on with live decremented by 1.
    2. The user has 1 live:
        2a. The user loses the game.
6a. There are still objects in the game:
    1. The game goes on.

UML Use Case Diagram

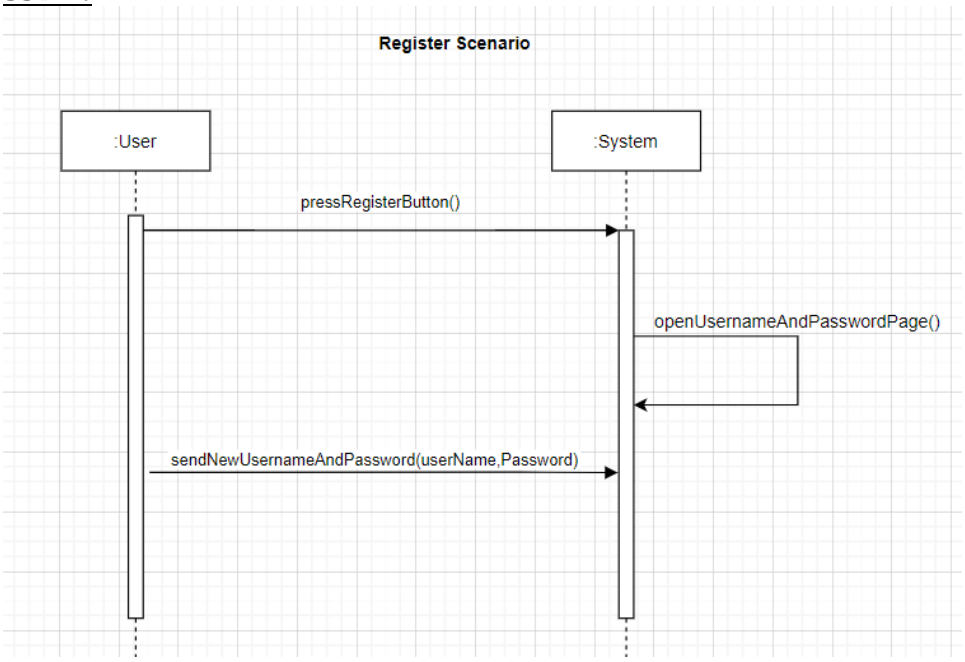


Domain Model

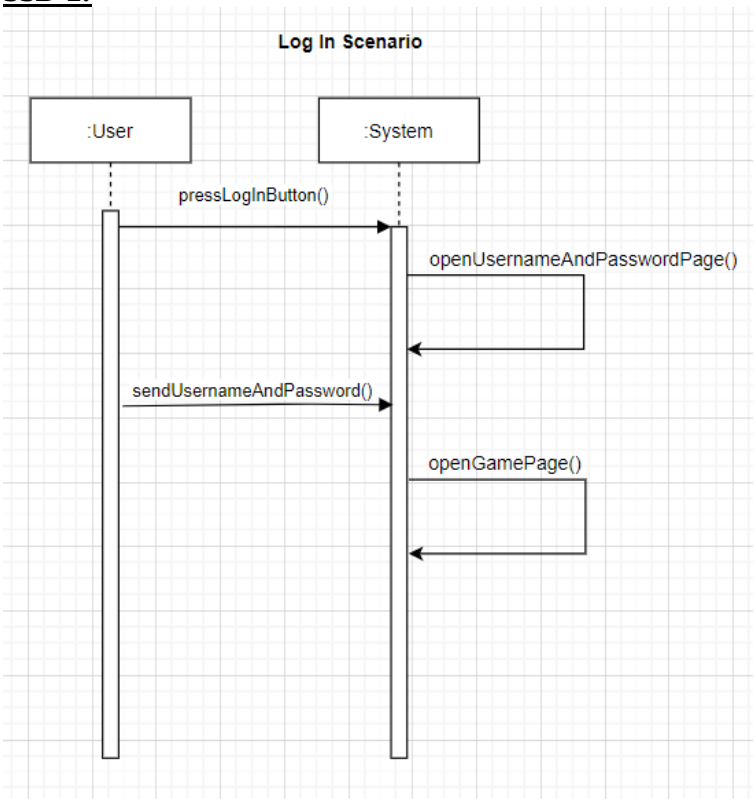


System Sequence Diagrams

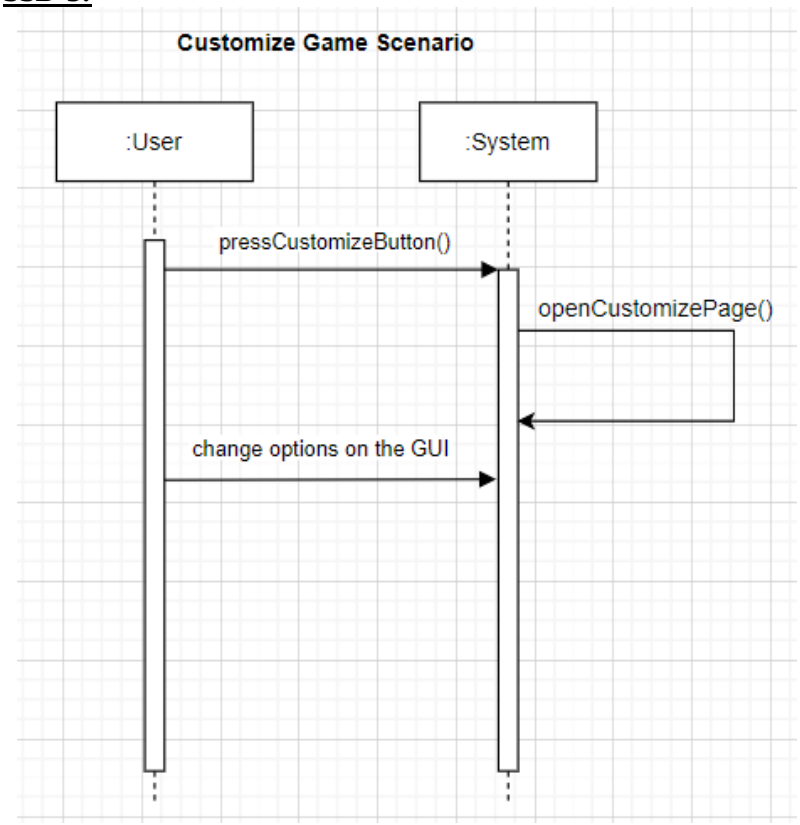
SSD-1:



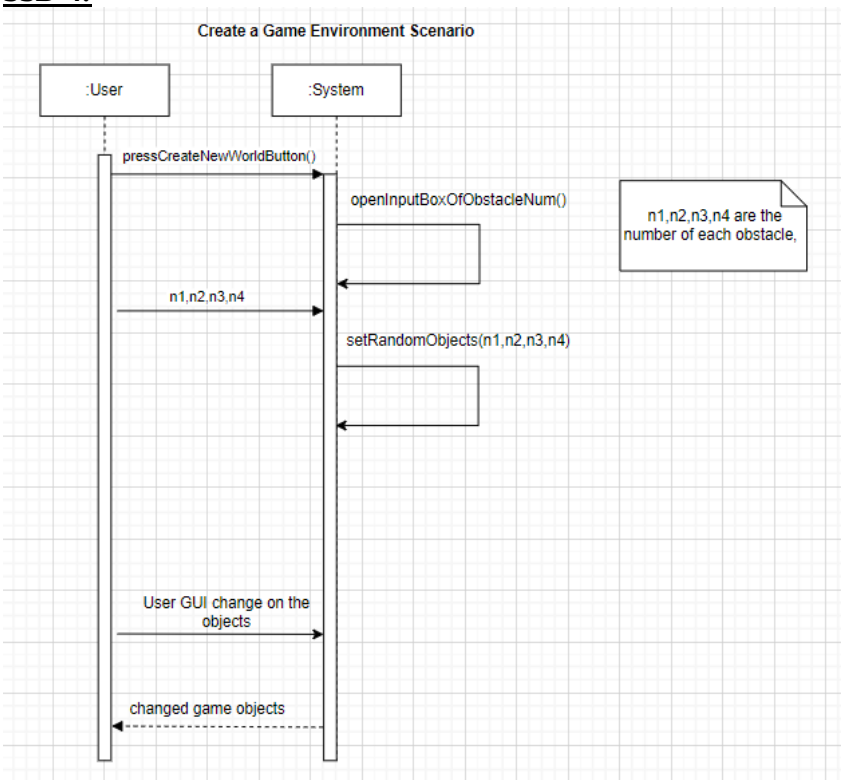
SSD-2:



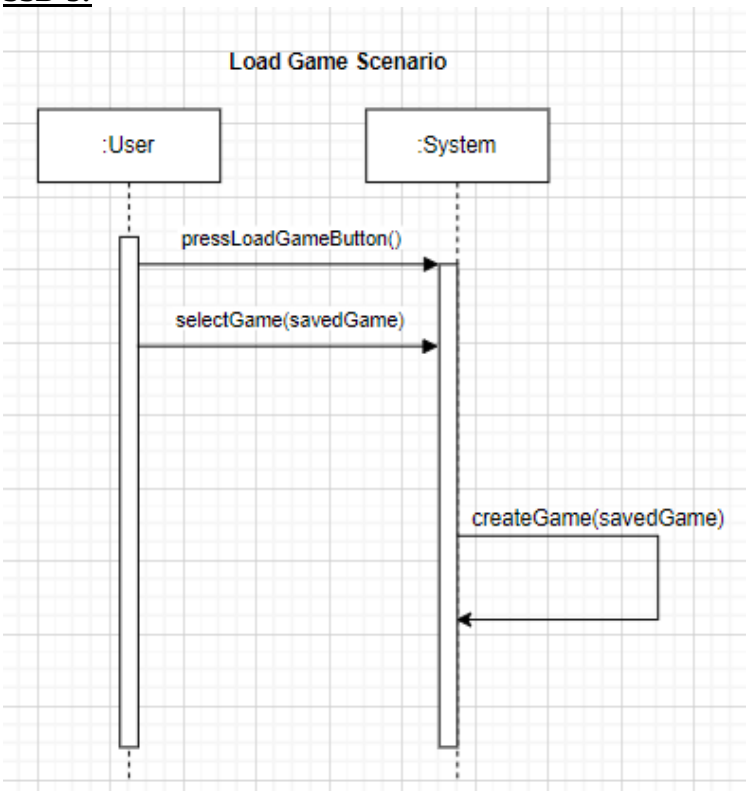
SSD-3:


Customize Game Scenario
:User
:System
pressCustomizeButton()
openCustomizePage()
change options on the GUI

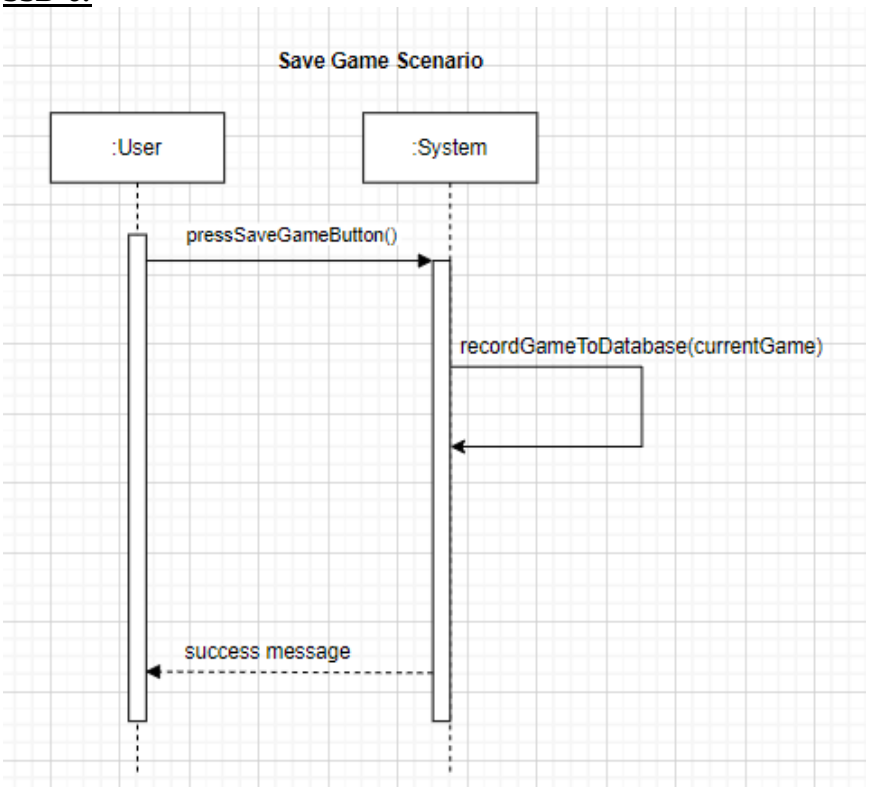
SSD-4:


Create a Game Environment Scenario
:User
:System
pressCreateNewWorldButton()
openInputBoxOfObstacleNum()
n1,n2,n3,n4 are the number of each obstacle,
n1,n2,n3,n4
setRandomObjects(n1,n2,n3,n4)
User GUI change on the objects
changed game objects

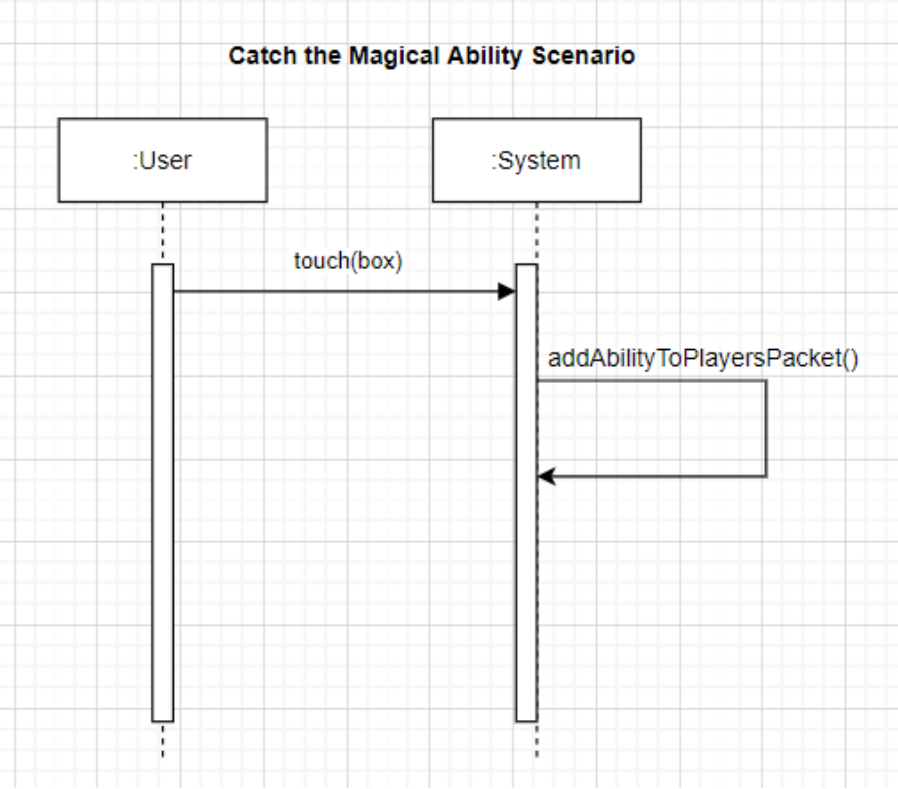
SSD-5:


Load Game Scenario
:User
:System
pressLoadGameButton()
selectGame(savedGame)
createGame(savedGame)

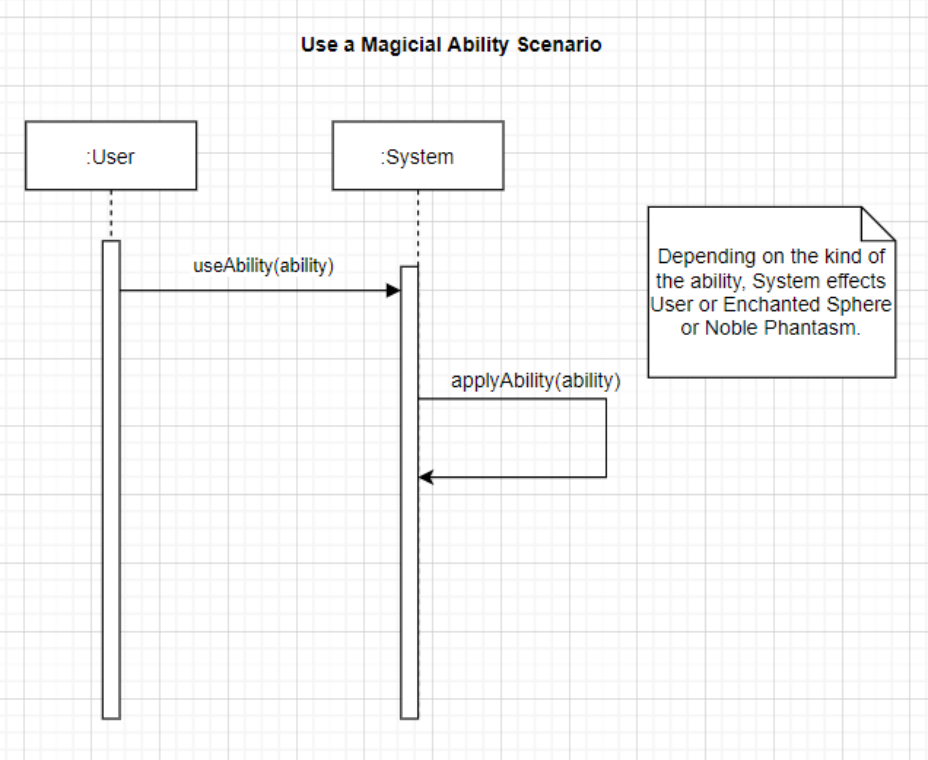
SSD-6:


Save Game Scenario
:User
:System
pressSaveGameButton()
recordGameToDatabase(currentGame)
success message

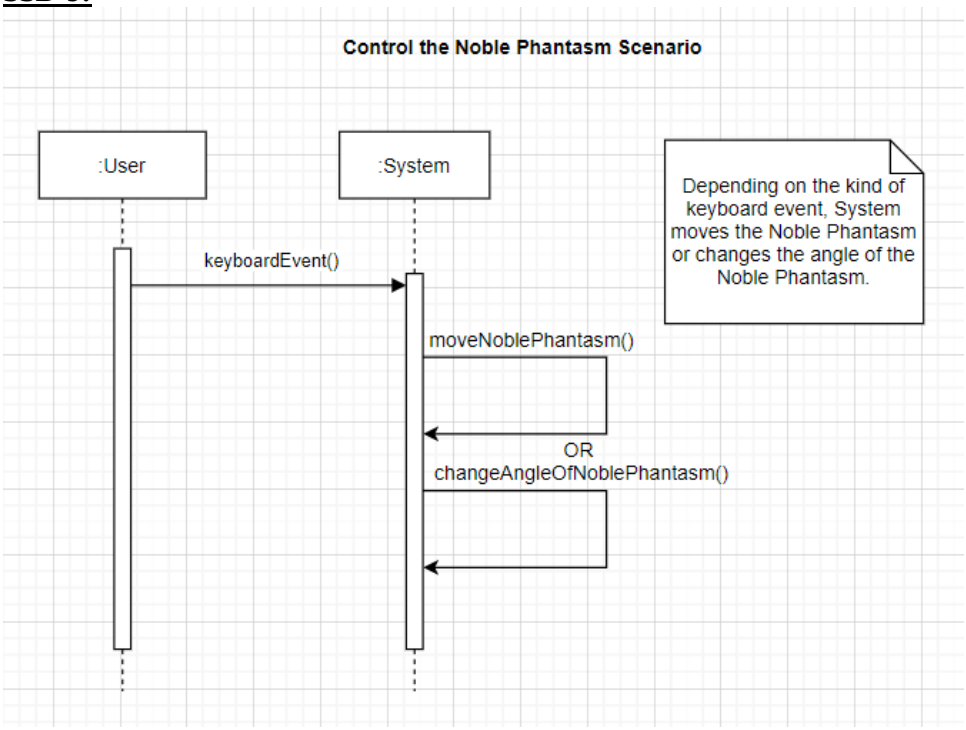
SSD-7:


Catch the Magical Ability Scenario
:User
:System
touch(box)
addAbilityToPlayersPacket()

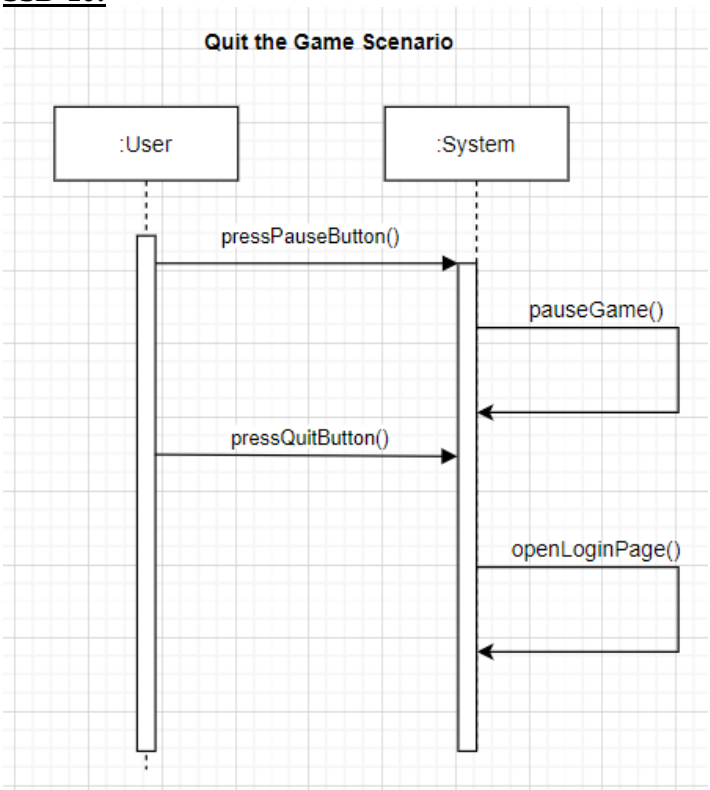
SSD-8:


Use a Magicial Ability Scenario
:User
:System
useAbility(ability)
applyAbility(ability)
Depending on the kind of the ability, System effects User or Enchanted Sphere or Noble Phantasm.

SSD-9:


Control the Noble Phantasm Scenario
:User
:System
Depending on the kind of keyboard event, System moves the Noble Phantasm or changes the angle of the Noble Phantasm.
keyboardEvent()
moveNoblePhantasm()
OR
changeAngleOfNoblePhantasm()

SSD-10:


Quit the Game Scenario
:User
:System
pressPauseButton()
pauseGame()
pressQuitButton()
openLoginPage()

Operation Contracts

Operation-1: pressRegisterButton()
Cross Reference: Use case 1 (Register)
Pre Condition: *Game.menu* was *OpeningPage.*
Post Condition: *Game.menu* was changed to *AskUserNamePasswordPage.*

Operation-2: pressLogInButton()
Cross Reference: Use case 2 (Log in to the game)
Pre Condition: *Game.page* was *OpeningPage.*
Post Condition: *Game.page* was *AskUserNamePasswordPage*.

Operation-3: openCustomizeMenu()
Cross Reference: Use Case 3 (Customize Game)
Pre condition: *Game.page* was *OpeningPage*
Post Condition: *Game.page* was changed to *CustomizingPage*.

Operation-4: pressCreateNewWorldButton()
Cross Reference: Use case 4 (Create the Game Environment)
Pre Condition: *Game.page* was *CreateOrLoadOrCustomizePage.*
Post Condition: *Game.page* was changed to *CustomizingPage.*

Operation-5: pressLoadGameButton()
Cross Reference: Use case 5 (Load Game)
Pre Condition: *Game.page* was *CreateOrLoadOrCustomizePage.* OR *Game.page* was *WorldPage.*.
Post Condition: Game.page was changed to *WorldPage.*

Operation-6: pressSaveGameButton()
Cross Reference: Use case 6 (Save Game)
Pre-condition: *Game.page* was *WorldPage.*
Post Condition: *Game.page* was not changed. Game's data is now in the database.

Operation-7: touch(box)
Cross Reference: Use Case 8 (Catch the Magical Ability)
Pre Condition: Game.page was WorldPage, and Game.mode was RunningMode. A box was dropped by a Gift Obstacle.
Post Condition: One value in (depending on the magical ability contained in the box) *user.magicalAbilities* was increased by one.

Operation-8: useAbility(ability)
Cross Reference: Use Case 9 (Use a Magical Ability)
Pre Condition: *user.magicalAbilities* != [0,0,0,0] (which means the user has an ability.) and *Game.mode* is *RunningMode.*
Post Condition: One value in (depending on the magical ability contained in the box) *user.magicalAbilities* was decreased by one. Magical ability is added to Enchanted Sphere or Noble Phantasm or user depending on the type of ability. (For example *NoblePhantasm.length* could be doubled.)

Operation-9: pressPauseButton()
Cross Reference: Use Case 13 **(**Quit Game)
Pre Condition: *Game.mode* was *RunningMode.*
Post Condition: *Game.Mode* was changed to *PausedMode*.

Operation-10: pressQuitButton()
Cross Reference: Use Case 13 **(**Quit Game)
Pre Condition: *Game.page* was *WorldPage and Game.Mode was PausedMode.*
Post Condition: *Game.page* was changed to *OpeningPage,* and *Game.mode* was changed to *null*.

Other Artifacts

Vision:

Introduction

- We will be designing an easy to play and enjoyable game targeting everyone. It is based on breaking blocks and gathering points. It interacts with the user by buttons on the display. There are some tricks like superpowers (gained by the user, enchanted sphere, and noble phantasm), and the obstacles in the game which makes it more fun.

Positioning

- Game will be available on the application stores and takes place in the game category. It comes forward with its basic and simple design but also various in-game scenarios by far compared to other similar games in the market. It targets all ages but especially children to ensure an enjoyable time.

Stakeholder Summary:

- All the COMP302 staff including our instructor, ta's, and the other people who will play this game will be the stakeholders.

Product overview:

- The game is designed to give people a break from a boring moment. It works by the movement of a spear which controlled by users so it is quite easy to learn and play. Overall, the Need for Spear will be a fun game and available on all digital platforms to download.

Supplementary Specification:

Introduction

- Supplementary specification document involves all the requirements which are not mentioned in the use cases.

Usability

- The customers (users) will see the menu including the "LogIn" and "Register" parts, the game's main screen (including "Save" "Load" "Pause" "Quit"), and the menu including the "create game", "load game", and "customize game" parts. So, there are some important human factors which affect the game play.

Human Factors

- The system messages, the writings in the buttons, and the writings in the pages should be readable and understandable enough so that users cannot encounter a problem about playing the game.

- Very shiny (also very dark colors and very light colors) colors should not be utilized in the game so that users and people having some defects of vision cannot face a problem about seeing the images and writings in the game.
- Moreover, to provide a better game play experience to the customers; the game should be easy to understand and play. The transaction from the login screen to the main screen of the game should be fast enough. Furthermore, to make the gameplay experience better, the process of hitting the obstacles and destroying the obstacles should be fast enough.To provide smooth and fast game play experience to the customers (users), the game should include almost no errors (preferably no errors).

Reliability

- If the user has a problem with his PC, such as freezed screen or broken keyboard and broken mouse due to rage quit, the game saves itself if the user does not do any IO event for some time - the time is specified in the game menu - so that the progress is not lost.
- If the user wants to delete a load game, it is asked for confirmation of the deletion, so that there is no trouble of a misclick.
- If the user wants to delete its account, he/she can do it any time.
- No data of the user is shared with third party companies, so the user's privacy is %100 safe. Also we don’t use cookies.

Performance

- We do not want users to spend a lot of time trying to start a game, so there is only a log-in menu, and the user can start the game with just 2 clicks.
- If the user wants to create a new level, it is implemented simply.
- When user types save, there is no failure in saving with %99.9.
- If the user has a low performance PC, then he/she can change the theme to a lower standart to get better performance in the game.
- Saved games are stored in the PC with the minimum storage capacity.

Supportability

- This game runs on all operating systems.
- users can use a mouse or a touchpad.
- The user can change the theme, so that it needs lower performance to display the screen which yields to a better performance.

Implementation Constraints

- We should be developing with Java by having a point of view that Java will support several Java-based tools like Maven (Java-based build automation tool). Moreover, it will be easy to develop with Java by using the concepts including modularity, inheritance, polymorphism, and the other object-oriented programming concepts.

Free Open Source Components

- To make the process of game development faster, we should try to find more Java-based tools which are freely available.

Legal Issues

- In the process of sales of this game, the taxations of the sales should be properly applied depending on the law regulations.

Glossary:

Definitions:

Terms	Definitions & Information
Score	The number which shows the user's success and increases when the obstacles are broken.
Noble Phantasm	The game object which allows the user to catch the enchanted sphere and throw it.
Lives	The number which shows how many times the user can fall down the enchanted sphere below the noble phantasm without losing the game.
Magical abilities	The abilities that can be gained by the enchanted sphere, user, or noble phantasm. The types of the magical abilities are “chance giving ability”, “noble phantasm expansion”, “magical hex”, “unstoppable enchanted sphere”.
Chance Giving Ability	The magical ability that increments the chances of the users by 1.
Noble Phantasm Expansion	The magical ability that doubles the length of the noble phantasm and lasts 30 seconds after the activation of it by the user.
Magical Hex	The magical ability which adds 2 magical canons to the both ends of the noble phantasm and which lasts 30 seconds after the activation. The magical canons can create magical hexes which hit the obstacles.
Unstoppable Enchanted Sphere	The magical ability which makes the enchanted sphere destroy each type of obstacle after hitting them, and this magical ability lasts 30 seconds after the activation.
Obstacles	The game objects that are destroyed by the enchanted sphere. The obstacles have different types, and every obstacle has a different effect in the game.

Simple obstacle	The obstacle that is destroyed with just one hit.
Firm obstacle	The obstacle that is destroyed with the number of hits written on the obstacle.
Explosive obstacle	The obstacle which has a sphere-like shape, and which explodes when it gets destroyed.
Gift obstacles	The obstacle that breaks with just one hit and drops a reward falling down. If the user catches it, the user gets the reward.