**Operations Contract**

**Operation:** moveNoblePhantasm(input: Key)

**Cross References:** Use Cases: Move Noble Phantasm Horizontally.

**Preconditions:**

* Player has enough chances.
* There is room for movement in the specified direction.

**Postconditions:** Noble Phantasm is moved horizontally.

**Operation:** rotateNoblePhantasm(input: Key)

**Cross References:** Use Cases: Rotate Noble Phantasm up to 45 degrees.

**Preconditions:**

* Player has enough chances.
* There is room for rotation in the specified direction.

**Postconditions:** Noble Phantasm is rotated.

**Operation:** compareCoordinates(objectName1: Object, objectName2: Object)

**Cross References:** Use Cases: Hit a Wall, Pick Up a Magical Ability, Hit an Obstacle

**Preconditions:** The two objects have not hit each other.

**Postconditions:** - A Boolean called “satisfyingCoordinates” is created. The boolean is indicating whether they crash or not in that instance.

**Operation:** destroyGift()

**Cross References:** Use Cases: Pick Up a Magical Ability

**Preconditions:** Enchanted Sphere has hit the Gift Obstacle.

**Postconditions:** - A Box object is created.

-The box starts falling downwards.

**Operation:** pickUpAbility()

**Cross References:** Use Cases: Pick Up a Magical Ability

**Preconditions:** Noble Phantasm touched the box.

**Postconditions:** - The player has one of the four random magical abilities.

**Operation:** reflect(angle: integer, moving: Boolean, corner: Boolean)

**Cross References:** Use Cases: Hit a Wall

**Preconditions:** Enchanted Sphere has hit an object.

**Postconditions:** - Depending on the values of the parameters, Enchanted Sphere reflects off of the object it hit and continues moving with the same speed.

**Operation:** loadGame(game: Game)

**Cross References:** Use Cases: Load a Game

**Preconditions:** Player hasn’t started any game.

**Postconditions:** Player loads the selected game and continues playing it from where it was exactly left.

**Operation:** useAbility(abilityType: Ability)

**Cross References:** Use Cases: Use an Ability

**Preconditions:** Player has at least one ability in the ability pack.

**Postconditions:** - The ability is activated.

- The ability pack is updated after activation.

**Operation:** shootES()

**Cross References:** Use Cases: Hit an Obstacle

**Preconditions:** Enchanted Sphere is on the Noble Phantasm and not moving.

**Postconditions:** - Enchanted Sphere starts moving in the shooting direction.

**Operation:** obstacleHit(obstacle: Object)

**Cross References:** Use Cases: Hit an Obstacle

**Preconditions:** Enchanted Sphere and the obstacle clash with each other.

**Postconditions:** - Obstacle is destroyed or the number on obstacle is decreased by 1 (firm obstacle).

**Operation:** addObstacle(obstacleType: Object, x: Integer, y: Integer)

**Cross References:** Use Cases: Build the Game

**Preconditions:** Player is in building mode.

**Postconditions:** - Obstacle is added to the layout.

- Obstacles do not overlap.

**Operation:** enterBuildingMode()

**Cross References:** Use Cases: Build the Game

**Preconditions:** Player is logged in.

**Postconditions:** - Player is able to add obstacles.

**Operation:** placeRandom(number: Integer)

**Cross References:** Use Cases: Build the Game

**Preconditions:** Player is in building mode.

**Postconditions:** - The obstacles in specified numbers are placed randomly in layout.

- Obstacles do not overlap.

**Operation:** startGame()

**Cross References:** Use Cases: Build the Game

**Preconditions:** Minimum criteria in building mode are satisfied.

**Postconditions:** - Player is able to play the game.

**Operation:** gameOver()

**Cross References:** Use Cases: End the Game

**Preconditions:**

* Player has chances left
* All obstacles destroyed.

**Postconditions:** None.