

Operation Contracts

Group: Group 9

Operation: pauseGame()
References: Use Cases: Pause/Resume Game
Pre-conditions: The game is in playing mode. Specifically, the game is running.
Post-conditions: The game was paused.

Operation: resumeGame()
References: Use Cases: Pause/Resume Game
Pre-conditions: The game is in playing mode. Specifically, the game is paused.
Post-conditions: The game resumed.

Operation: buildGame()
References: Use Cases: Select Game Mode
Pre-conditions: The game must be launched, and the player must be at the Main Menu screen.
Post-conditions: It shows us the map of all the halls.

Operation: playGame()
References: Use Cases: Select Game Mode
Pre-conditions: -All halls were filled with at least as many as the minimum number of objects as specified in the game rules.
Post-conditions: -The play mode is successfully initialized
-The map of the Hall of Earth was displayed.

Operation: help()
References: Use Cases: Select Game Mode
Pre-conditions: Game application is executed and working successfully.
Post-conditions: It helps us when clicked. Information regarding how to use the

interface, play the game and some fundamental rules of the game were displayed.

Operation: buildHall(hall)
References: Use Cases: BuildGame
Pre-conditions: -The player has clicked Build Mode from the Main Menu.
-The game transitioned to Build Mode.
Post-conditions: -The objects are placed in the hall.

Operation: checkNumberOfObjects()
References: Use Cases: BuildGame
Pre-conditions: -The player has clicked Build Mode from the Main Menu.
-The game transitioned to Build Mode.
Post-conditions: The number of objects populated in the hall based on the player's object placement choice was compared with the minimum number of object criteria for that hall.

Operation: transitionNextHall(Hall)
References: Use Cases: BuildGame
Pre-conditions: The value of checkNumberOfObject must be at least the required minimum number.
Post-conditions: -Next hall was unlocked to place the objects on it.

Operation: finishBuild()
References: Use Cases: BuildGame
Pre-conditions: All halls must be successfully filled with objects according to the rules.
Post-conditions: The player proceeds to Play Mode.

Operation: moveHero()
References: Use Cases: Move Hero
Pre-conditions: The game is in play mode.

Post-conditions: -The hero was moved east/west/south/north in response to user keyboard commands.
-The hero remained within the boundaries of the hall.

Operation: openObject()

References: Use Cases: Gather Rune

Pre-conditions: -The game is in play mode.
-Hero is next to the object.

Post-conditions: The rune emerged if it was hidden in that object before opening. Otherwise, nothing was displayed and the object was closed.

Operation: gatherRune()

References: Use Cases: Gather Rune

Pre-conditions: -The game is in play mode
-We know that the rune exists.

Post-conditions: -The door of the hall opened.
-The system made a sound indicating that the door was opened.

Operation: gatherEnchantment()

References: Use Cases: Gather Enchantment

Pre-conditions: The game is in playing mode

Post-conditions: -The enchantments have been manipulated for players' decisions (used or sent to inventory).
-The enchantment has been moved to the inventory or used.

Operation: updateInventory(Enchantment)

References: Use Cases: Gather Enchantment

Pre-conditions: -The game is in play mode.
-We need to be nearby.

Post-conditions: Hero's inventory has been updated based on gathered enhancements

Operation:	teleportHero(Hall)
References: Use Cases:	Teleport Hero
Pre-conditions:	<ul style="list-style-type: none"> -The game is in running mode. -The player successfully opened the door.
Post-conditions:	<ul style="list-style-type: none"> -The hall is updated accordingly to the next stage. -Hero is teleported to the next hall.
Operation:	revealRune(Inventory)
References: Use Cases:	Reveal Rune
Pre-conditions:	-The game is in running mode.
Post-conditions:	Number of reveal enchantments in the Inventory was checked.
Operation:	showRune(hasReveal)
References: Use Cases:	Reveal Rune
Pre-conditions:	<ul style="list-style-type: none"> -The game is in running mode. -The hero has at least one reveal enchantment in his/her inventory. -The hero used one of his/her reveal enchantments.
Post-conditions:	4x4 square one of which contains the rune is highlighted.
Operation:	decreaseRune()
References: Use Cases:	Reveal Rune
Pre-conditions:	The game is in running mode.
Post-conditions:	-The number of reveal enchantments has decreased in the hero's inventory by 1.
Operation:	protectHero(Inventory)
References: Use Cases:	Protect Hero
Pre-conditions:	-The game is in running mode.
Post-conditions:	Number of Cloak of Protection enchantments in the Inventory was checked.

Operation: hideHero(hasReveal)
References: Use Cases: Protect Hero
Pre-conditions: The player has at least one cloak of protection in his/her inventory.
Post-conditions: The archer can no longer see the hero.

Operation: decreaseCloak()
References: Use Cases: Protect Hero
Pre-conditions: The game is in running mode.
Post-conditions: -The number of the cloak of protection enchantments has decreased in the hero's inventory by 1.

Operation: throwLuringGem(Inventory)
References: Use Cases: Throw Luring Gem
Pre-conditions: -The game is in running mode
Post-conditions: Number of Luring Gem enchantments in the Inventory was checked.

Operation: chooseThrowDirection(hasLuringGem)
References: Use Cases: Throw Luring Gem
Pre-conditions: -The game is in playing mode
-The player has at least one cloak of protection in his/her inventory.
Post-conditions: Fighter monster started moving towards the luring gem.

Operation: decreaseLuringGem()
References: Use Cases: Throw Luring Gem
Pre-conditions: The game is in running mode
Post-conditions: The number of Luring Gem enchantments has decreased in the hero's inventory by 1.