<<Interface>>
Dungeon

+ createDungeon(type : DungeonType, interconnectivity : int)
+ enterDungeon()
+ getPlayerDescription() : String
+ getLocationDescription() : String
+ move(d : Direction)
+ pickUpTreasure()
+ isGameOver() : boolean

<<enumeration>>
LocationType

TUNNEL
CAVE

<<enumeration>>
TreasureType

DIAMOND
RUBY
SAPPHIRE

<<enumeration>>
DungeonType

WRAPPING
NON-WRAPPING

<<enumeration>>
Direction

NORTH
EAST
SOUTH
WEST

DungeonAbstract

rowNum : int

columnNum : int

locationNum : int

uf : UnionFind

treasure : TreasureType[][]

direction : Direction[][]

DungeonAbstract(rowNum : int, columnNum : int)

+ enterDungeon()

+ getPlayerDescription() : String

+ getLocationDescription() : String

+ move(d : Direction)

+ pickUpTreasure()

+ isGameOver() : boolean

- generateTreasure()

UnionFind

- count : int

- parent : int[]

+ UnionFinde(n : int)

+ union(locationP : int, locationQ : int)

+ getCount() : int

- findRoot(location : int)

+ isConnected(locationP : int, locationQ : int) : boolean

1   1

WrappingDungeon

+ WrappingDungeon(rowNum : int, columnNum : int)

NonWrappingDungeon

+ NonWrappingDungeon(rowNum : int, columnNum : int)

Testing create wrapping dungeon	Input	expected value
Instantiate a dungeon with the type of WrappingDungeon and 0 degree	createDungeon(DungeonType.WRAPPINGDUNGEON, 0, 6, 10)	A wrapping dungeon with size 6 * 10 and there is exactly one path
Instantiate a dungeon with the type of WrappingDungeon and 1 degree	createDungeon(DungeonType.WRAPPINGDUNGEON, 1, 6, 10)	A wrapping dungeon with size 6 * 10 and there are two paths
Instantiate a dungeon with illegal degree	createDungeon(DungeonType.WRAPPINGDUNGEON, -1, 6, 10)	IllegalArgumentException
Instantiate a dungeon with illegal row number	createDungeon(DungeonType.WRAPPINGDUNGEON, 11, -10, 10)	IllegalArgumentException
Instantiate a dungeon with illegal column number	createDungeon(DungeonType.WRAPPINGDUNGEON, 11, 6, -10)	IllegalArgumentException
Testing create non-wrapping dungeon		
Instantiate a dungeon with the type of NonWrappingDungeon and 0 degree	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 0, 6, 10)	A non-wrapping dungeon with size 6 * 10 and there is exactly one path
Instantiate a dungeon with the type of NonWrappingDungeon and 1 degree	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 1, 6, 10)	A non-wrapping dungeon with size 6 * 10 and there are two paths
Instantiate a dungeon with illegal degree	createDungeon(DungeonType.NONWRAPPINGDUNGEON, -1, 6, 10)	IllegalArgumentException
Instantiate a dungeon with illegal row number	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 11, -10, 10)	IllegalArgumentException
Instantiate a dungeon with illegal column number	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 11, 6, -10)	IllegalArgumentException
Testing enterDungeon() in a wrapping dungeon		
Instantiate a 8 * 8 wrapping dungeon with two degrees	createDungeon(DungeonType.WRAPPINGDUNGEON, 2, 8, 8)	player's location:position:[3,4], moves[north, east,

dungeon with two degrees	enterDungeon()	west],treasure[diamond * 2]
	getLocationDescrtiption()	
Testing enterDungeon() in a non-wrapping dungeon		
Instantiate a 8 * 8 non-wrapping dungeon with two degrees	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 2, 8, 8)	player's location:position:[1,1], moves[east, west],treasure[diamond * 1, ruby * 1]
	enterDungeon()	
	getLocationDescrtiption()	
Testing getPlayerDescription() in a wrapping dungeon		
Instantiate a 6 * 8 wrapping dungeon with 4 degrees	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	player's treasure : diamond * 8, sapphire * 2]
	getPlayerDescription()	
Testing getPlayerDescription() in a non-wrapping dungeon		
Instantiate a 10 * 8 wrapping dungeon with 8 degrees	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 8, 10, 8)	player's treasure : diamond * 15, sapphire * 12, ruby * 11]
	getPlayerDescription()	
Testing getLocationDescription() in a wrapping dungeon		
Instantiate a 6 * 8 wrapping dungeon with 4 degrees	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	player's location:position:[1,1], moves[east, west],treasure[diamond * 1, ruby * 1]
	getPlayerDescrtion()	
Testing getLocationDescription() in a non-wrapping dungeon		

Instantiate a 10 * 8 wrapping dungeon with 8 degrees	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 8, 10, 8)	player's location:position:[1,1], moves[north,east, west],treasure[sopphire * 5]
	getPlayerDescription()	
Testing move() in a wrapping dungeon		
Instantiate a 6 * 8 wrapping dungeon with 4 degrees and move the player by passing illegal parameter	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	IllegalArgumentException
	getLocationDescrtiption(), it returns possible directions : west, east	
	move(Direction.NORTH)	
Instantiate a 6 * 8 wrapping dungeon with 4 degrees and move the player by passing correct parameter	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	player's position is [1,2]
	getLocationDescrtiption(), it returns player's position is [1,1]	
	move(Direction.EAST)	
	getLocationDescrtiption()	
Testing move() in a non-wrapping dungeon		
Instantiate a 6 * 8 non-wrapping dungeon with 4 degrees and move the player by passing illegal parameter	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 4, 6, 8)	IllegalArgumentException
	getLocationDescrtiption(), it returns possible directions : west, east	
	move(Direction.NORTH)	
Instantiate a 6 * 8 non-wrapping dungeon with 4 degrees and move the player by passing correct parameter	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 4, 6, 8)	player's position is [1,2]
	getLocationDescrtiption(), it returns player's position is [1,1]	

	move(Direction.EAST)	
	getLocationDescrtiption()	
Testing pickUpTreasure() in a wrapping dungeon		
Instantiate a 6 * 8 wrapping dungeon with 4 degrees and pickUpTreasure() at that location	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	after = "ruby * 2, sopphire * 1"
	getLocationDescrtiption(), it returns the treasure in the room: ruby * 2, sopphire * 1	
	String before = getPlayerDscription(); return ""	
	pickUpTreasure()	
	String after = getPlayerDscription()	
Instantiate a 6 * 8 wrapping dungeon with 4 degrees and pickUpTreasure() at that location	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	after = "diamond * 1,ruby * 3, sopphire * 2"
	getLocationDescrtiption(), it returns the treasure in the room: ruby * 2, sopphire * 1	
	String before = getPlayerDscription(); return "diamond * 1, ruby * 1, sopphire * 1"	
	pickUpTreasure()	
	String after = getPlayerDscription()	
Instantiate a 6 * 8 wrapping dungeon with 4 degrees and pickUpTreasure() at that location	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	after = "diamond * 1,ruby * 1, sopphire * 1"
	getLocationDescrtiption(), it returns the treasure in the room: nothing	
	String before = getPlayerDscription(); return "diamond * 1, ruby * 1, sopphire * 1"	
	pickUpTreasure()	

	String after = getPlayerDscription()	
Testing pickUpTreasure() in a non-wrapping dungeon		
Instantiate a 6 * 8 non-wrapping dungeon with 4 degrees and pickUpTreasure() at that location	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 4, 6, 8)	after = "ruby * 2, sopphire * 1"
	getLocationDescrtiption(), it returns the treasure in the room: ruby * 2, sopphire * 1	
	String before = getPlayerDscription(); return ""	
	pickUpTreasure()	
	String after = getPlayerDscription()	
Instantiate a 6 * 8 non-wrapping dungeon with 4 degrees and pickUpTreasure() at that location	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 4, 6, 8)	after = "diamond * 1,ruby * 3, sopphire * 2"
	getLocationDescrtiption(), it returns the treasure in the room: ruby * 2, sopphire * 1	
	String before = getPlayerDscription(); return "diamond * 1, ruby * 1, sopphire * 1"	
	pickUpTreasure()	
	String after = getPlayerDscription()	
Instantiate a 6 * 8 non-wrapping dungeon with 4 degrees and pickUpTreasure() at that location	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 4, 6, 8)	after = "diamond * 1,ruby * 1, sopphire * 1"
	getLocationDescrtiption(), it returns the treasure in the room: nothing	
	String before = getPlayerDscription(); return "diamond * 1, ruby * 1, sopphire * 1"	
	pickUpTreasure()	

	String after = getPlayerDscription()	
Testing isGameOver() in a wrapping dungeon		
Instantiate a 6 * 8 wrapping dungeon with 4 degrees and test isGameOver() : false	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	FALSE
	getLocationDescrtiption(), it returns the location position[1,1]	
	String end = getEnd(); return "5, 5"	
	after several move()	
	getLocationDescrtiption(), it returns the location position[3,4]	
	isGameOver()	
Instantiate a 6 * 8 wrapping dungeon with 4 degrees and test isGameOver() : true	createDungeon(DungeonType.WRAPPINGDUNGEON, 4, 6, 8)	TRUE
	getLocationDescrtiption(), it returns the location position[1,1]	
	String end = getEnd(); return "5, 5"	
	after several move()	
	getLocationDescrtiption(), it returns the location position[5,5]	
	isGameOver()	
Testing isGameOver() in a non-wrapping dungeon		

Instantiate a 6 * 8 non-wrapping dungeon with 4 degrees and test isGameOver() : false	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 4, 6, 8)	FALSE
	getLocationDescrtiption(), it returns the location position[1,1]	
	String end = getEnd(); return "5, 5"	
	after several move()	
	getLocationDescrtiption(), it returns the location position[3,4]	
	isGameOver()	
Instantiate a 6 * 8 non-wrapping dungeon with 4 degrees and test isGameOver() : true	createDungeon(DungeonType.NONWRAPPINGDUNGEON, 4, 6, 8)	TRUE
	getLocationDescrtiption(), it returns the location position[1,1]	
	String end = getEnd(); return "5, 5"	
	after several move()	
	getLocationDescrtiption(), it returns the location position[5,5]	
	isGameOver()	