<<interface>>
DungeonController
+ void playConsoleGame(DungeonInterface model)
+ void playGraphicalGame()
+ boolean movePlayer(String location)
+ void pickUpTreasure()
+ void pickUpArrows()
+ void shoot(String direction, int distance)
+ void buildNewModel(int row, int column, int interconnectivity, DungeonType dungeonType, int treasurePercentage, int difficulty)
+ void setViewFocus()
<<interface>>
View
+ void refresh()
+ void makeVisible()
+ void setNewSettings(ReadOnlyDungeonInterface newReadOnlyModel, int row, int column, int interConnectivity, DungeonType dunType, int treasurePercent, int diff)
+ void setFocus()
+ void makeInvisible()
DungeonConsoleController
+ DungeonConsoleController(Readable in, Appendable out)
+ void playConsoleGame(DungeonInterface dungeonModel)
DungeonGraphicalController
- DungeonInterface model;
- View view;
- ConfigurationsView configurationsView;
+ DungeonGraphicalController(Dungeon dungeonModel, MockView view)
+ void playGraphicalGame()
+ boolean movePlayer(String location)
+ void pickUpTreasure()
+ void pickUpArrows()
+ void shoot(String direction, int distance)
+ void buildNewModel(int row, int column, int interconnectivity, DungeonType dungeonType, int treasurePercentage, int difficulty)
+ void setViewFocus()
+ void playConsoleGame(DungeonInterface dungeonModel)
DungeonView
+ void refresh()
+ void makeVisible()
+ void setNewSettings(ReadOnlyDungeonInterface newReadOnlyModel, int row, int column, int interConnectivity, DungeonType dunType, int treasurePercent, int diff)
+ void setFocus()
+ void makeInvisible()
<<interface>>
Randomizer
+ int getRandomValueInRange(int min, int max)
+ int getRandomValue(int max)
+ List<Integer> getStartStopRandom(int min, int max)
+ public int getRandomPathIndex()
+ public void randomMax(int max)
TrueRandom
- Random random = new Random();
- List<Path> pathList = new ArrayList<>();
- List<Integer> randomNumbers;
- int max;
+ int getRandomValueInRange(int min, int max)
+ int getRandomValue(int max)
+ List<Integer> getStartStopRandom(int min, int max)
+ public int getRandomPathIndex()
+ public void randomMax(int max)
MockRandom
- List<Path> pathList = new ArrayList<>();
- List<Path> randomPaths = new ArrayList<>();
- List<Integer> randomNumbers;
- int max;
+ int getRandomValueInRange(int min, int max)
+ int getRandomValue(int max)
+ List<Integer> getStartStopRandom(int min, int max)
+ public int getRandomPathIndex()
+ public void randomMax(int max)
MockRandomICWrapping
- List<Integer> randomNumbers;
- int max;
+ int getRandomValueInRange(int min, int max)
+ int getRandomValue(int max)
+ List<Integer> getStartStopRandom(int min, int max)
+ public int getRandomPathIndex()
+ public void randomMax(int max)
<<interface>>
DungeonInterface
+ void buildDungeonGrid()
+ void pathBuilder()
+ void buildMST()
+ void addInterconnectivity()
+ void setDirections()
+ void identifyCavesAndTunnels()
+ void putTreasureInCaves()
+ Location[][] getDungeon()
+ List getPathList()
+ List<Path> getFinalSelectedPathsForDungeon()
+ void putMonsterInCaves()
+ void putArrowsInDungeon()
+ Smell detectSmell(Location location)
+ boolean movePlayer(String location)
+ OtyughStatus shootPlayer(String direction, int distance)
+ Location getPlayerLocation()
+ void setPlayerLocation(Location playerLocation)
+ void pickUpTreasure()
+ void pickUpArrows()
+ List<Treasure> getPlayerWealth()
+ int getPlayerArrowCount()
+ getTreasureListFromLocation()
+ getArrowCountInLocation()
+ void resetArrowCountInLocation()
+ int getLocationId()
+ Location getNorthLocation()
+ Location getSouthLocation()
+ Location getEastLocation()
+ Location getWestLocation()
DungeonIndex
-int i
-int j
+ getI()
+getJ()
DungeonPanel
- ReadOnlyDungeonInterface readOnlyDungeonModel;
- DungeonController controller;
- BufferedImage locationImg;
- BufferedImage playerImg;
- BufferedImage otyughImg;
- BufferedImage thieftreg;
- BufferedImage smellImage;
- BufferedImage combinedImage;
~ DungeonPanel(ReadOnlyDungeonInterface readOnlyDungeonModel, DungeonController controller)
+ void setModel(ReadOnlyDungeonInterface resetModel)
+ void paintComponent(Graphics p)
LocationDescriptionPanel
- ReadOnlyDungeonInterface readOnlyDungeonModel;
- DungeonController controller;
- BufferedImage treasureImage;
- BufferedImage arrowImage;
- JButton treasureBtn;
- JButton arrowBtn;
~ LocationDescriptionPanel(ReadOnlyDungeonInterface readOnlyDungeonModel, DungeonController controller)
+ void setModel(ReadOnlyDungeonInterface resetModel)
+ void paintComponent(Graphics p)
PlayerDescriptionPanel
- ReadOnlyDungeonInterface readOnlyDungeonModel;
- DungeonController controller;
- BufferedImage treasureImage;
- BufferedImage arrowImage;
~ PlayerDescriptionPanel(ReadOnlyDungeonInterface readOnlyDungeonModel, DungeonController controller)
+ void setModel(ReadOnlyDungeonInterface resetModel)
+ void paintComponent(Graphics p)
<<interface>>
LocationInterface
+ int getCaveId()
+ List getTreasureList()
+ void setTreasureList(Treasure treasure)
+ CaveType getCaveType()
+ void setCaveType(CaveType caveType)
+ Cave getNorth()
+ void setNorth(Cave north)
+ Cave getSouth()
+ void setSouth(Cave south)
+ Cave getEast()
+ void setEast(Cave east)
+ Cave getWest()
+ void setWest(Cave west)
+ void addMonster(Monster otyugh)
+ void addArrowToLocation()
+ void deductHealthOfOtyugh()
+ boolean hasMonster()
+ int getArrowCountInLocation()
+ void resetArrowCount()
+ Monster getMonster()
<<interface>>
PlayerInterface
+ Location getPlayerLocation()
+ void setPlayerLocation(Cave playerLocation)
+ List<Treasure> getWealth()
+ void setWealth(List<Treasure> wealth)
+ boolean move(String location)
+ void pickUpTreasure()
+ void pickUpArrows()
+ int getArrowCount()
+ void setArrowCount(int numOfPickedArrows)
+ OtyughStatus shoot(String direction, int distance)
Dungeon
- int row;
- int column;
- int interconnectivity;
- DungeonType dungeonType;
- int treasurePercentage;
- List<Path> pathList;
- List<Cave> caveList;
- Randomizer randomizer;
- List<Path> -finalSelectedPathsForDungeon = new -ArrayList<>();
- List<Path> leftOverPaths = new -ArrayList<>();
- int[] father;
- int edgeCount = 0;
- List<Integer> caveLocations = new ArrayList();
- Cave[][] dungeonGrid;
+ void buildDungeonGrid()
+ void pathBuilder()
+ void buildMST()
+ void addInterconnectivity()
+ void setDirections()
+ void identifyCavesAndTunnels()
+ void putTreasureInCaves()
+ Location[][] getDungeon()
+ List getPathList()
+ List<Path> getFinalSelectedPathsForDungeon()
+ void putMonsterInCaves()
+ void putArrowsInDungeon()
+ Smell detectSmell(Location location)
+ boolean movePlayer(String location)
+ OtyughStatus shootPlayer(String direction, int distance)
+ Location getPlayerLocation()
+ void setPlayerLocation(Location playerLocation)
+ void pickUpTreasure()
+ void pickUpArrows()
+ List<Treasure> getPlayerWealth()
+ int getPlayerArrowCount()
+ getTreasureListFromLocation()
+ getArrowCountInLocation()
+ void resetArrowCountInLocation()
+ int getLocationId()
+ Location getNorthLocation()
+ Location getSouthLocation()
+ Location getEastLocation()
+ Location getWestLocation()
Path
- Cave source;
- Cave destination;
+ Cave getSource()
+ Cave getDestination()
Player
- Location playerLocation;
- List<Treasure> wealth = new ArrayList<>();
- int arrowCount;
+ Location getPlayerLocation()
+ void setPlayerLocation(Cave playerLocation)
+ List<Treasure> getWealth()
+ void setWealth(List<Treasure> wealth)
+ boolean move(String location)
+ void pickUpTreasure()
+ void pickUpArrows()
+ int getArrowCount()
+ void setArrowCount(int numOfPickedArrows)
+ OtyughStatus shoot(String direction, int distance)
<<ENUM>>
CaveType
CAVE, TUNNEL
Cave
- int caveId;
- Cave north;
- Cave south;
- Cave east;
- Cave west;
- CaveType caveType;
- List<Treasure> treasureList;
- List<Monster> monster
- int arrowCount
+ int getCaveId()
+ List getTreasureList()
+ void setTreasureList(Treasure treasure)
+ CaveType getCaveType()
+ void setCaveType(CaveType caveType)
+ Cave getNorth()
+ void setNorth(Cave north)
+ Cave getSouth()
+ void setSouth(Cave south)
+ Cave getEast()
+ void setEast(Cave east)
+ Cave getWest()
+ void setWest(Cave west)
+ void addMonster(Monster otyugh)
+ void addArrowToLocation()
+ void deductHealthOfOtyugh()
+ boolean hasMonster()
+ int getArrowCountInLocation()
+ void resetArrowCount()
+ Monster getMonster()
Monster
-int health
+ void deductHealth(int health)
+ int getHealth()
<<Smell>>
STRONG_PUNGENT_SMELL
PUNGENT_SMELL
<<ENUM>>
TreasureType
DIAMOND,
RUBY,
SAPPHIRE
<<ENUM>>
DungeonType
WRAPPING,
NON_WRAPPING