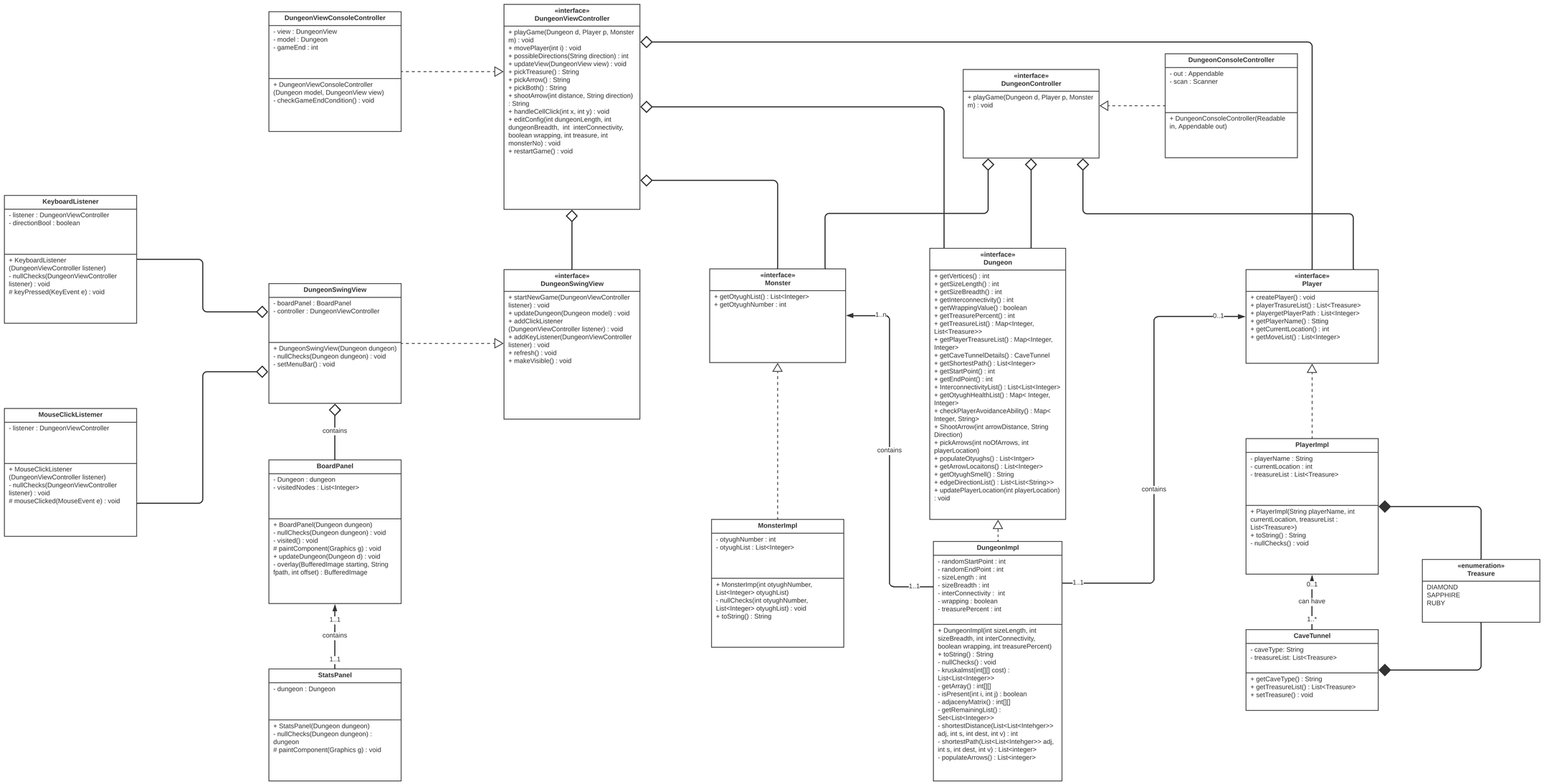
DungeonViewConsoleController
- view : DungeonView
- model : Dungeon
- gameEnd : int
+ DungeonViewConsoleController (Dungeon model, DungeonView view)
- checkGameEndCondition() : void
«interface» DungeonViewController
+ playGame(Dungeon d, Player p, Monster m) : void
+ movePlayer(int i) : void
+ possibleDirections(String direction) : int
+ updateView(DungeonView view) : void
+ pickTreasure() : String
+ pickArrow() : String
+ pickBoth() : String
+ shootArrow(int distance, String direction) : String
+ handleCellClick(int x, int y) : void
+ editConfig(int dungeonLength, int dungeonBreadth, int interConnectivity, boolean wrapping, int treasure, int monsterNo) : void
+ restartGame() : void
«interface» DungeonController
+ playGame(Dungeon d, Player p, Monster m) : void
DungeonConsoleController
- out : Appendable
- scan : Scanner
+ DungeonConsoleController(Readable in, Appendable out)
KeyboardListener
- listener : DungeonViewController
- directionBool : boolean
+ KeyboardListener (DungeonViewController listener)
- nullChecks(DungeonViewController listener) : void
keyPressed(KeyEvent e) : void
DungeonSwingView
- boardPanel : BoardPanel
- controller : DungeonViewController
+ DungeonSwingView(Dungeon dungeon)
- nullChecks(Dungeon dungeon) : void
- setMenuBar() : void
«interface» DungeonSwingView
+ startNewGame(DungeonViewController listener) : void
+ updateDungeon(Dungeon model) : void
+ addClickListener (DungeonViewController listener) : void
+ addKeyListener(DungeonViewController listener) : void
+ refresh() : void
+ makeVisible() : void
«interface» Monster
+ getOtyughList() : List<Integer>
+ getOtyughNumber : int
1..n
«interface» Dungeon
+ getVertices() : int
+ getSizeLength() : int
+ getSizeBreadth() : int
+ getInterconnectivity() : int
+ getWrappingValue() : boolean
+ getTreasurePercent() : int
+ getTreasureList() : Map<Integer, List<Treasure>>
+ getPlayerTreasureList() : Map<Integer, Integer>
+ getCaveTunnelDetails() : CaveTunnel
+ getShortestPath() : List<Integer>
+ getStartPoint() : int
+ getEndPoint() : int
+ InterconnectivityList() : List<List<Integer>
+ getOtyughHealthList() : Map< Integer, Integer>
+ checkPlayerAvoidanceAbility() : Map< Integer, String>
+ ShootArrow(int arrowDistance, String Direction)
+ pickArrows(int noOfArrows, int playerLocation)
+ populateOtyughs() : List<Intger>
+ getArrowLocaitons() : List<Integer>
+ getOtyughSmell() : String
+ edgeDirectionList() : List<List<String>>
+ updatePlayerLocation(int playerLocation) : void
«interface» Player
+ createPlayer() : void
+ playerTrasureList() : List<Treasure>
+ playergetPlayerPath : List<Integer>
+ getPlayerName() : Stting
+ getCurrentLocation() : int
+ getMoveList() : List<Integer>
0..1
contains
MouseClickListemer
- listener : DungeonViewController
+ MouseClickListener (DungeonViewController listener)
- nullChecks(DungeonViewController listener) : void
mouseClicked(MouseEvent e) : void
contains
BoardPanel
- Dungeon : dungeon
- visitedNodes : List<Integer>
+ BoardPanel(Dungeon dungeon)
- nullChecks(Dungeon dungeon) : void
- visited() : void
paintComponent(Graphics g) : void
+ updateDungeon(Dungeon d) : void
- overlay(BufferedImage starting, String fpath, int offset) : BufferedImage
1..1
contains
1..1
StatsPanel
- dungeon : Dungeon
+ StatsPanel(Dungeon dungeon)
- nullChecks(Dungeon dungeon) : dungeon
paintComponent(Graphics g) : void
MonsterImpl
- otyughNumber : int
- otyughList : List<Integer>
+ MonsterImp(int otyughNumber, List<Integer> otyughList)
- nullChecks(int otyughNumber, List<Integer> otyughList) : void
+ toString() : String
contains
1..1
DungeonImpl
- randomStartPoint : int
- randomEndPoint : int
- sizeLength : int
- sizeBreadth : int
- interConnectivity : int
- wrapping : boolean
- treasurePercent : int
+ DungeonImpl(int sizeLength, int sizeBreadth, int interConnectivity, boolean wrapping, int treasurePercent)
+ toString() : String
- nullChecks() : void
- kruskalmst(int[][] cost) : List<List<Integer>>
- getArray() : int[][]
- isPresent(int i, int j) : boolean
- adjacenyMatrix() : int[][]
- getRemainingList() : Set<List<Integer>>
- shortestDistance(List<List<Intehger>> adj, int s, int dest, int v) : int
- shortestPath(List<List<Intehger>> adj, int s, int dest, int v) : List<integer>
- populateArrows() : List<integer>
1..1
PlayerImpl
- playerName : String
- currentLocation : int
- treasureList : List<Treasure>
+ PlayerImpl(String playerName, int currentLocation, treasureList : List<Treasure>)
+ toString() : String
- nullChecks() : void
«enumeration» Treasure
DIAMOND
SAPPHIRE
RUBY
0..1
can have
1..*
CaveTunnel
- caveType: String
- treasureList: List<Treasure>
+ getCaveType() : String
+ getTreasureList() : List<Treasure>
+ setTreasure() : void