gameApplicationMain

<<constructor>> +gameApplicationMain()
+start(Stage): void

+main(String[]): void

ASCIIGameController

-strToDirection(String): Direction
-getUse(String): UseAction
+visualiseLevel(Level): String
-visTypeToSprite(VisType): String
-printInventory(Iterator<Action>): void

+main(String[]): void

PauseMenuController

-howToPlayer: Button
-restart: Button
-reusme: Button
-currStage: Stage

<<constructor>> +PauseMenuController(Stage)
+handleHowToPlayButton(): void
+handleRestartButton(): void
+handleResumeButton(): void

PauseMenuScreen

-s: Stage
-title: String
-fxmlLoader: FXMLLoader

<<constructor>> +PauseMenuScreen()
+start(): void

OpeningController

-playerModeButton: Button
-designerMode: Button
-currStage: Stage

<<constructor>> +OpeningController(Stage)
+initiailize(): void
+handlePlayerModeButton(ActionEvent): void
+handleDesignerModeButton(ActionEvent): void

OpeningScreen

-s:Stage
-title:String
-fxmlLoader: FXMLLoader

<<constructor>> #OpeningScreen(Stage)
+start(): void

InstructionsController

-exitScreen:Button
-currStage: Stage

<<constructor>> +InstructionsController(Stage)
+exitScreen(ActionEvent): void

InstructionsScreen

-s: Stage
-title: String
-fxmlLoader: FXMLLoader

<<constructor>> +InstructionsScreen()
+start(): void

<<uses>>

<<uses>>

<<uses>>

PlayingController

-pause: Button
-map: GridPane
-lostPane: Pane
-wonPane: Pane
-player1Lost: Text
-player2Lost: Text
-won: Text
-wonImage: ImageView
-lostImage: ImageView
-player1Flying: ImageView
-player2Flying: ImageView
-inventory: GridPane
-inventoryDisplayPanes: ArrayList<ArrayList<JFXPane>>
-players: ArrayList<PlayerMobileEntity>
-player1: PlayerMobileEntity
-player2: PlayerMobileEntity
-currStage: Stage
-parentStage: Stage
#hunterThread: Thread
#timer: Timer
-player1MovedLastTick: boolean
-player2MovedLastTick: boolean
-stopPlayer1NextTick: boolean
-stopPlayer2NextTick: boolean
-l: Level

<<constructor>> +PlayingController(Stage)
<<constructor>> +PlayingController(Stage, Stage)
+handlePauseButton(ActionEvent): void
+returnHome(): void
+setMap(Level): void
+setInventory(): void
-updateInventory(): void
+setPlayers(ArrayList<PlayerMobileEntity>): void
+keyDownToAction(KeyCode): void
+keyUpToAction(KeyCode): void
+runTick(): void
+getMap(): GridPane

PlayingScreen

-s: Stage
-parentStage: Stage
-title: String
-fxmlLoader: FXMLLoader
-mapView: GridPane

<<constructor>> +PlayingScreen(Stage)
+start(): void
+setMap(Level): void
+setPlayers(ArrayList<PlayerMobileEntity>): void

<<uses>>

<<Interface>>
LevelController

+getNumRows(): int
+getNumCols(): int
+newMaze(int, int): void
+placeEntity(VisType, int, int): boolean
+removeEntity(VisType, int, int): boolean
+clearTile(int, int): boolean
+hasLost(): boolean
+hasWon(): boolean
+setPlayerDirection(Direction): void
+setPlayerSpeed(int): void
+useItem(UseAction): boolean

PlayerMobileEntity

0..*

Level

1

Tile

1..*

«interface»
Observable

«interface»
Observer

JFXPane

<<uses>>

PlayerModeHomeController

-easy: Level
-players: ArrayList<PlayerMobileEntity>
-homeButton: Button
easyLevelButton: Button
-intermediateLevelButton: Button
-hardLevelButton: Button
-currStage: Stage

<<constructor>> +PlayerModeHomeController(Stage)
+handleHomeButton(ActionEvent): void
+handleEasyLevelButton(ActionEvent): void
+handleIntermediateLevelButton(ActionEvent): void
+handHardLevelButton(ActionEvent): void
-makeEasyMap(): void

PlayerModeHomeScreen

-s: Stage
-title: String
-fxmlLoader: FXMLLoader

<<constructor>> +PlayerModeHomeScreen()
+start(): void

MutliThreading

<<constructor>> #Multithreading()
+run(): void

DesignerModeHomeScreen

-s:Stage
-title: String
-fxmlLoader: FXMLLoader

<<constructor>> +DesignerModeHomeScreen(Stage)
+start(): void

DesignerModeHomeController

-homeButton: Button
-createNewButton: Button
-play: Button
-editMao: Button
-delete: Button
-scrollPane: ScrollPane
-listOfGames: ListView<DesigningController>
-currStage: Stage
-savedControllers: ObservableList<DesigningController>
-listOfDesignControllers: ObservableList<DesigningController>
-designingScreen: DesigningScreen

<<constructor>> +DesignerModeHomeController(Stage)
-handleHomeButton(ActionEvent): void
+initialize(): void
-handleCreateNewButton(ActionEvent): void
+playMap(): void
+editMap(): void
+deleteMap(ActionEvent): void
-listViewSet(): void

DesigningScreen

-s: Stage
-parentStage: Stage
-title: String
-fxmlLoader: FXMLLoader

<<constructor>> +DesigningScreen(Stage)
+start(): DesigningController
-loadController(DesigningController): void

DesigningController

-currentlySelected: VisType
-name: String
-players: ArrayList<PlayerMobileEntity>
-buttons for all entities
-winConditions: CheckBox
-clear: Button
-save:Button
-exitScreen: Button
-play: Button
-textfields for headings
-difficultySlider: Slider
-map: GridPane
-parentStage: Stage
-currStage: Stage
-saveState: SimpleBooleanProperty
-I: Level
-make: EntityMaker
-newPlayer: PlayerMobileEntity
-setRow: int
-setCol: int
-playerSelected: PlayerMobileEntity

-DEFAULT_NROWS: int
-DEFAULT_NCOLS: int

<<constructor>> +DesigningController(Stage, Stage)
+initialize(): void
+setters for all entities
+clearSelected(): void
+saveMap(): void
+playMap(ActionEvent): void
+exitScreen(): void
-setItem(int, int): void
+getSaveProperty(): SimpleBooleanProperty
+resetSaveProperty(): void
+getLevel(): Level
+getName(); String
+setMapValues(): void
-playerOnTile(int, int)< PlayerMobileEntity
+getPlayers(): ArrayList<PlayerMobileEntity>
-generateMap(): void

JFXPane

-pane:Pane
-image: ImageView
-row: int
-column: int
-clicked: SimpleBooleanProperty

<<constructor>> +JFXPane()
<<constructor>> +JFXPane(int, int)
+getPane(): Pane
+detectMouseClicks(): void
+getRow(): int
+getColumn(): int
+clickedProperty(): BooleanProperty
+resetClicked(): void
+update(Observable, Object): void
+update(Object): void
+visTypeToPath(VisType): String
+invVistypeToPath(VistType): String

«interface» Observer

«interface» Observable

Tile

MobileEntity

Level

<<uses>>

<<uses>>