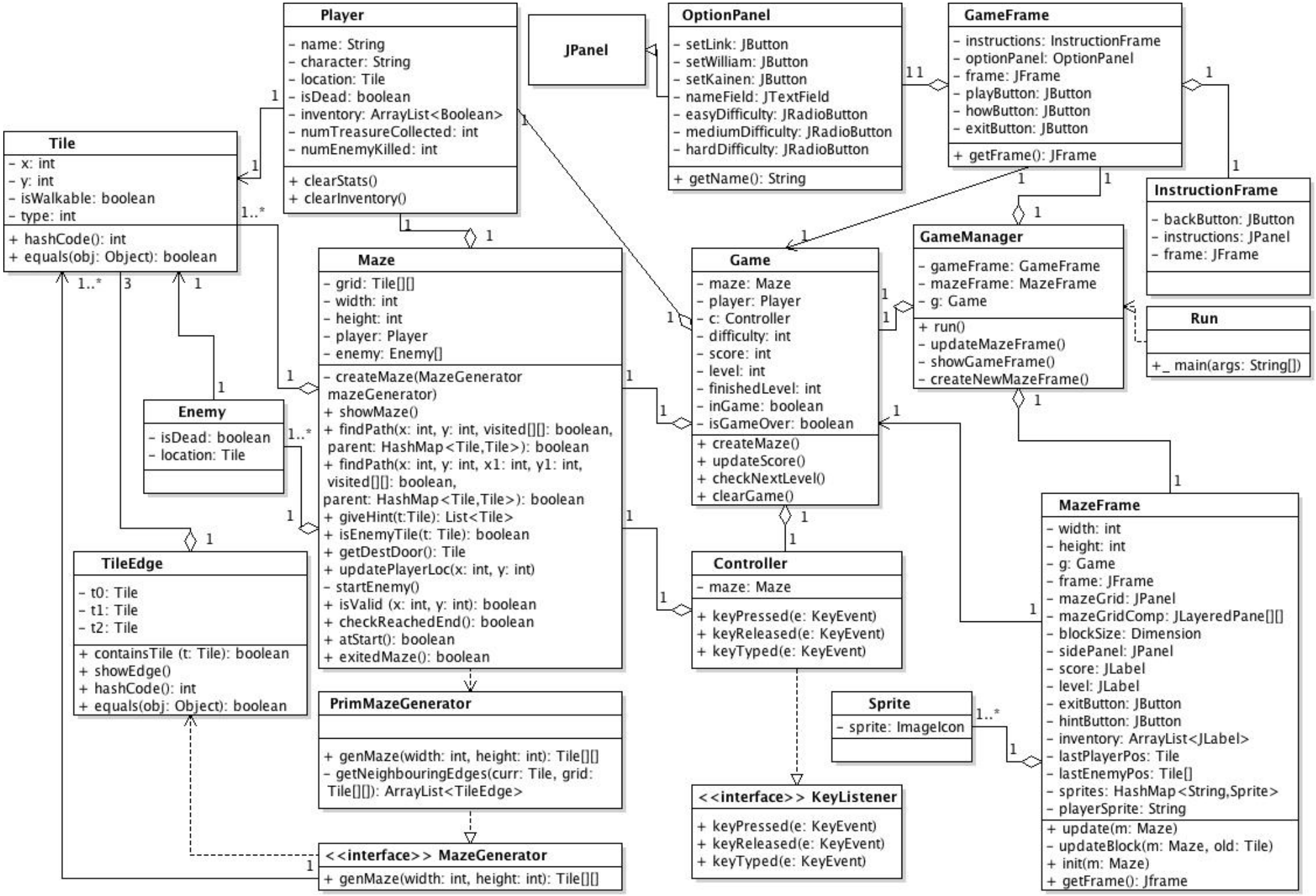
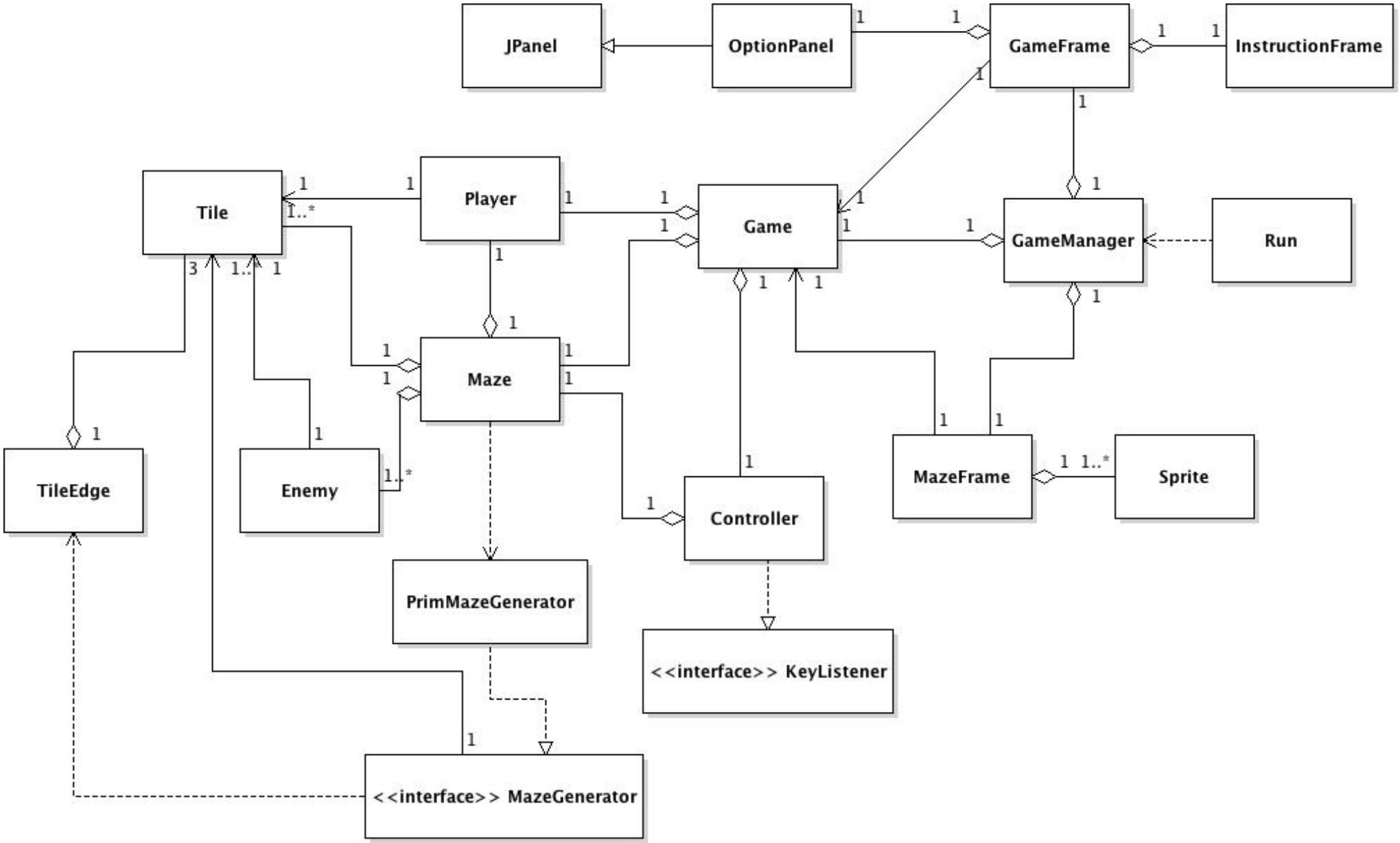

Player
- name: String
- character: String
- location: Tile
- isDead: boolean
- inventory: ArrayList<Boolean>
- numTreasureCollected: int
- numEnemyKilled: int
+ clearStats()
+ clearInventory()
JPanel
OptionPanel
- setLink: JButton
- setWilliam: JButton
- setKainen: JButton
- nameField: JTextField
- easyDifficulty: JRadioButton
- mediumDifficulty: JRadioButton
- hardDifficulty: JRadioButton
+ getName(): String
GameFrame
- instructions: InstructionFrame
- optionPanel: OptionPanel
- frame: JFrame
- playButton: JButton
- howButton: JButton
- exitButton: JButton
+ getFrame(): JFrame
InstructionFrame
- backButton: JButton
- instructions: JPanel
- frame: JFrame
Tile
- x: int
- y: int
- isWalkable: boolean
- type: int
+ hashCode(): int
+ equals(obj: Object): boolean
Maze
- grid: Tile[][]
- width: int
- height: int
- player: Player
- enemy: Enemy[]
- createMaze(MazeGenerator mazeGenerator)
+ showMaze()
+ findPath(x: int, y: int, visited[][]: boolean, parent: HashMap<Tile,Tile>): boolean
+ findPath(x: int, y: int, x1: int, y1: int, visited[][]: boolean, parent: HashMap<Tile,Tile>): boolean
+ giveHint(t:Tile): List<Tile>
+ isEnemyTile(t: Tile): boolean
+ getDestDoor(): Tile
+ updatePlayerLoc(x: int, y: int)
- startEnemy()
+ isValid (x: int, y: int): boolean
+ checkReachedEnd(): boolean
+ atStart(): boolean
+ exitedMaze(): boolean
Game
- maze: Maze
- player: Player
- c: Controller
- difficulty: int
- score: int
- level: int
- finishedLevel: int
- inGame: boolean
- isGameOver: boolean
+ createMaze()
+ updateScore()
+ checkNextLevel()
+ clearGame()
GameManager
- gameFrame: GameFrame
- mazeFrame: MazeFrame
- g: Game
+ run()
- updateMazeFrame()
- showGameFrame()
- createNewMazeFrame()
Run
+_ main(args: String[])
Enemy
- isDead: boolean
- location: Tile
TileEdge
- t0: Tile
- t1: Tile
- t2: Tile
+ containsTile (t: Tile): boolean
+ showEdge()
+ hashCode(): int
+ equals(obj: Object): boolean
Controller
- maze: Maze
+ keyPressed(e: KeyEvent)
+ keyReleased(e: KeyEvent)
+ keyTyped(e: KeyEvent)
MazeFrame
- width: int
- height: int
- g: Game
- frame: JFrame
- mazeGrid: JPanel
- mazeGridComp: JLayeredPane[][]
- blockSize: Dimension
- sidePanel: JPanel
- score: JLabel
- level: JLabel
- exitButton: JButton
- hintButton: JButton
- inventory: ArrayList<JLabel>
- lastPlayerPos: Tile
- lastEnemyPos: Tile[]
- sprites: HashMap<String,Sprite>
- playerSprite: String
+ update(m: Maze)
- updateBlock(m: Maze, old: Tile)
+ init(m: Maze)
+ getFrame(): Jframe
Sprite
- sprite: ImageIcon
PrimMazeGenerator
+ genMaze(width: int, height: int): Tile[][]
- getNeighbouringEdges(curr: Tile, grid: Tile[][]): ArrayList<TileEdge>
<<interface>> KeyListener
+ keyPressed(e: KeyEvent)
+ keyReleased(e: KeyEvent)
+ keyTyped(e: KeyEvent)
<<interface>> MazeGenerator
+ genMaze(width: int, height: int): Tile[][]


JPanel
OptionPanel
GameFrame
InstructionFrame
Tile
Player
Game
GameManager
Run
Maze
TileEdge
Enemy
Controller
MazeFrame
Sprite
PrimMazeGenerator
<<interface>> KeyListener
<<interface>> MazeGenerator