Entity - x: double - y: double - width: double - height: double + getBounds: Rectangle Inanimate Animate - x: double - x: double - y: double - y: double width: double width: double - height: double - height: double - speed: int getBounds(): Rectangle getBounds(): Rectangle Power-up x: double y: double width: double height: double Traps - x: double - y: double Scientist - width: double + getBounds(): Rectangle + afterInteraction: void - x: double height: doubledamage: int - y: double - width: double height: doubledetectionWidth: double + getBounds(): Rectangle + getDamage(): int + afterInteraction: void - detectionHeight: double - damage: int - isSeekingPlayer: bool + getBounds(): Rectangle + getDetectionBounds: Rectangle + onDetection(): void Spikes - x: double + getDamage(): int + afterInteraction: void - y: double - width: double - height: double + getBounds(): Rectangle + getDamage(): int + afterInteraction: void Potion Player - x: double - y: double - x: double - width: double - y: double - height: double - width: double - height: double - health: int + getBounds(): Rectangle - stamina: int - score: int + afterInteraction: void + getBounds(): rectangle + checkInteraction(Entity entity): void Draw + field: type + method(type): type - nonPlayerEntities: Entity[] - playerInteractionChecker: PlayerInteractionChecker Main Tile - game : Game + field: type + Maze(): void +Main(): + checkInteractions: void - createScientist():Scientist + method(type): type - createSpike(): Spike - createTile(int x, int y): void - createWall(int x, int y, Direction direction): void KeyListner Wall + field: type + method(type): type GameWindow GamePanel KeyboardInput + field: type + JFrame jframe + field: type + GameWindow(); + GamePanel() + KeyboardInput(GamePanel) -keyPressed (Keyevent) : void -keyReleased (Keyevent): void **JPanel** Classname **JFrame** + field: type + field: type + field: type + method(type): type + method(type): type + method(type): type MouseInputs + field: type + mouseClicked(MouseEvent e): void

MouseListener

MazeObject

+ MazeObject(x, y): void

- y: int

Game

+ field: type

+ method(type): type