Main:MazeGame

main():void

Main:GamePanel

+ WIDTH:final int
+ HEIGHT:final int
+ SCALE:final int
- Thread:thread
- running:boolean
- FPS:int
- targetTime:long
- BufferedImage:image
- g:Graphics2D
- gsm:GameStateManager

- init():void
+ addNotify():void
+ run():void
- draw():void
- drawToScreen():void
+ keyTyped(KeyEvent):void
+ keyPressed(KeyEvent):void
+ mouseClicked(MouseEvent):<<Override>>void
+ mousePressed(MouseEvent):<<Override>>void
+ mouseReleased(MouseEvent):<<Override>>void
+ mouseMoved(MouseEvent):<<Override>>void

GameState:MainMenu

- font:Front

- select():void
+ keyPressed(int): void
+ keyReleased(int):void
+ mousePressed(MouseEvent):void
+ mouseReleased(mouseEvent):void
+ mouseMoved(mouseEvent):void

GameState:GameState

gsm:GameStateManager

+ draw:(Graphics2D):void
+ keyPressed(int):void
+ keyReleased(int):void
+ MousePressed(double, double):void
+ MouseReleased(double, double):void
+ MouseMoved(double, double):void

GameState:Difficulty

- currentChoice:int
- options:String[]
- font:Font

+ draw:(Graphics2D):void
- select():void
+ keyPressed(int):void
+ keyReleased(int):void
+ MousePressed(double, double):void
+ MouseReleased(double, double):void

GameState:GameStateManager

- gameStates:GameState[]
- playStyle:int
- currentState:int
- startLevel:int
- generator:Generator
+ NUMGAMESTATES:int
+ MAINMENU:int
+ SINGLEPLAYER:int
+ MULTIPLAYER:int
+ DIFFICULTY:int
+ HISTORY:int
+ OPTIONSMENU:int
+ GENERATORMENU:int

- loadState(int):void
- unloadState(int):void
+ setState(int):void
+ setStartLevel(int):void
+ draw(Graphics2D): void
+ keyPressed(int): void
+ keyReleased(int):void
+ mousePressed(MouseEvent):void
+ mouseReleased(mouseEvent):void
+ mouseMoved(mouseEvent):void

GameState:SinglePlayer

- m:Maze
- p1:Player
- keys:List<keyObject>
- ghosts:List<ghost>
- swords:List<swordObject>
- mark1:List<Tracemark>
- keyNum:int
- ghostNum:int
- level:int
- pauseFlag:int
- menuselection = 0;
- height:int
- width:int
- tilelength:int
- showtile:int
- mzaemap:minimap
- numberOfSword:int
- exitX:int
- exitY:int

+init():void
-SPlevelGenerator():void
-tips():void
+draw():void
+ keyPressed(int): void
+ keyReleased(int):void
+ mousePressed(MouseEvent):void
+ mouseReleased(mouseEvent):void
+ mouseMoved(mouseEvent):void

GameState:MultiPlayer

- m:Maze
- p1:Player
- p2:Player
- keys:List<keyObject>
- ghosts:List<ghost>
- swords:List<swordObject>
- mark1:List<Tracemark>
- mark2:List<Tracemark>
- keyNum:int
- ghostNum:int
- level:int
- pauseFlag:int
- menuselection = 0;
- height:int
- width:int
- tilelength:int
- showtile:int
- numberOfSword1:int
- numberOfSword2:int
- exitX:int
- exitY:int
- noticeFlag:int

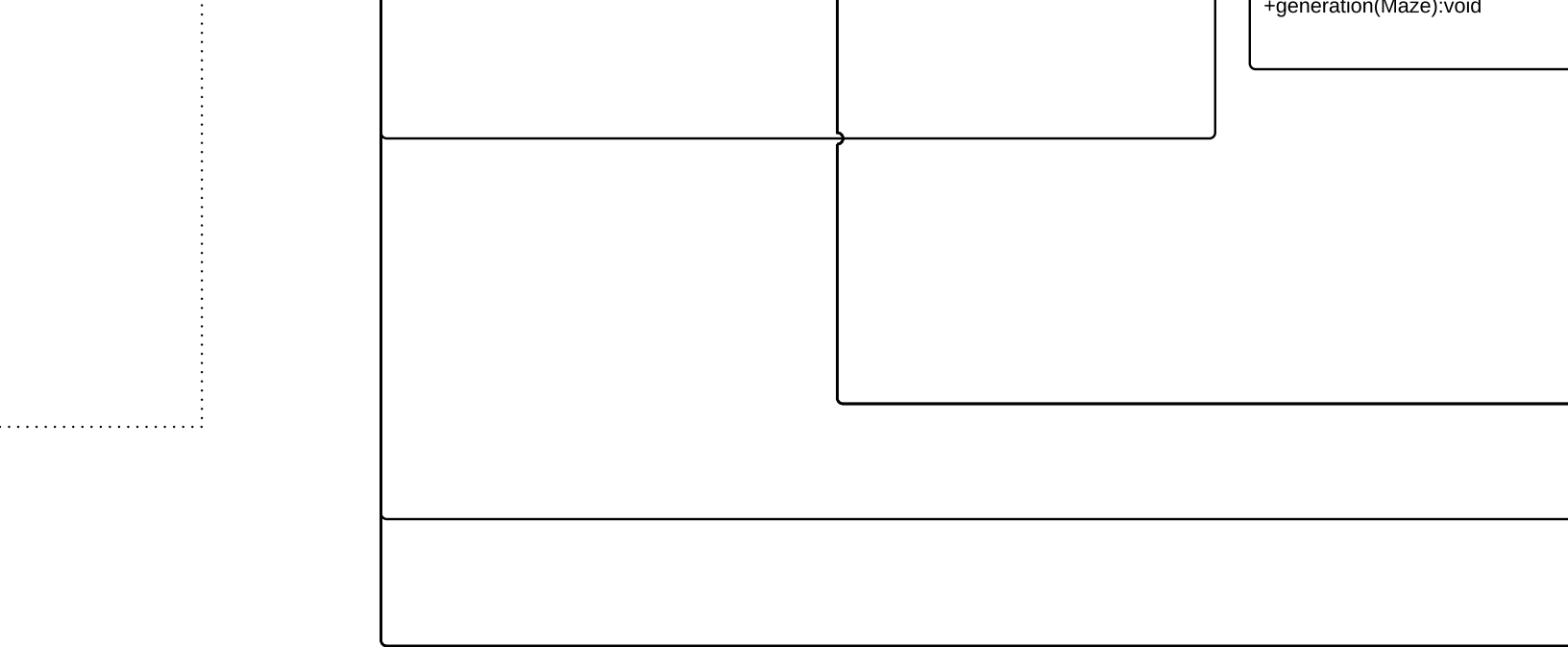+init():void
-MPlevelGenerator():void
-tips():void
+draw():void
+ keyPressed(int): void
+ keyReleased(int):void
+ mousePressed(MouseEvent):void
+ mouseReleased(mouseEvent):void
+ mouseMoved(mouseEvent):void

Generator:KruskalGenerat

+generation(Maze):void

Maze:Maze

- mymaze:Wall[][]
- height:int
- width:int
- generator:Generator
- entireMaze:BufferedImage
- tileImagelength:int

+ getWall(int, int):Wall
+ getX():int
+ getY():int
+ ifwallbyint(int, int, int):boolean
+ deleteEdge(int, int, String):void
- GetOppositeEdge(String):String
+ DispalyMaze():void
+ dirtoname(int):String
+ getNeighbour(Wall):ArrayList<GameObject>
- calAllReference(): void
- buildmazeImage():void
- referencetoname(int):String
+ getmazemap():BufferedImage
+ gettotalHeight():int
+ gettotalWidth():int
+ gettitleimagelength():int
+ getTransparency():void

Maze:Wall

- North:GameObject
- South:GameObject
- West:GameObject
- East:GameObject
- xCo:int
- yCo:int
- imagereference:int

+ flip(String):void
+ isWall(String):boolean
+ PrintOnConsole(String):void
+ iswallbyint(String):boolean
+ getWall(String):GameObject
+ GetX():int
+ GetY():int
+ calReference():int
+ getImageReference():int
+ getImageReference():void

Maze:Gam

- alive:boolean
- myName:String
- tilex:int
- tiley:int
- image:Image

+ ChangeState():void
+ IsAlive():boolean
+ GetName():String
+ equals(Object):boole
+ settilex(int):void
+ settiley(int):void
+ gettilex():int
+ gettiley():int

Maze:minimap

- lenX:int
- lenY:int
- R:int[][]
- quaterlowerX:int
- quaterhigherX:int
- quaterlowerY:int
- quaterhigherY:int
- size:int

-drawMap():void
-init():void
+getLength():int
-change(int, int):void
+getminimap(int, int):Image
+getTransparency():int

Maze:Player

- dy:int
- dx:int
- delta:int
- player_south:Image
- player_north: Image
- player_west: Image
- player_east: Image
- numberofsword:int
- steps:int

+ move():void
+ getswordnumber():int
+ changeswordnumber(int):void
+ getSteps():int
+ setDx(int):void
+ setDy(int):void
+ getDirection():int
+ equals(Object):boolean
+ getPlayerNorth():Image
+ getPlayerSouth():Image
+ getPlayerEast():Image
+ getPlayerWest():Image

Generator<interface>

+generation(Maze):void

Maze:swordObject

- sword:Image

+ equals(Object):boolean
+ getSword():Image

Generator

Generator:BackTrackingGenerator

+generation(Maze):void
+findNeighbour(GameObject, Maze):Wall

Maze:GameObject (partially cut off)

ze:GameObject

ng

e():void
lean
String
ct):boolean
oid
oid

Maze:Ghost

- dy:int
- dx:int
- delta:int
- ghost_south:Image
- ghost_north: Image

+ move():void
+ reverseMove():void
+ setDx(int):void
+ setDy(int):void
+ getDirection():int
+ equals(Object):boolean
+ getGhostNorth():Image
+ getGhostSouth():Image

Maze:keyObject

- map:Image

+ equals(Object):boolean
+ getMap():Image

Maze:Tracemark

- mark:Image

+ equals(Object):boolean
+ getMark():Image

+ keyReleased(int):void
+ MousePressed(double, double):void
+ MouseReleased(double, double):void
+ MouseMoved(double, double):void

GameState:History

- score:int
- level:int
- name:String
- sc:Scanner
- backButton:int
- mylist:ArrayList<history>;

+ compareTo(self):int
+ getIntScore():int
+ draw(Graphics2D):void
+ read():void
- printList(Graphics2D):void
+ keyPressed(int): void
+ keyReleased(int):void
+ mousePressed(MouseEvent):void
+ mouseReleased(mouseEvent):void
+ mouseMoved(mouseEvent):void

+generation(Maze):void

+findNeighbour(GameObject, Maze):Wall