

## <<interface>> Game1Builder + buildHearts(context Context): List<Hearts> + buildBackground(context Context): Bitmap

+ getGame1StatusUpdater(): ObservableInterface

- constructItems(game1Builder:Game1Builder,

+ drawManager(canvas: Canvas):void

- drawGemsList(canvas: Canvas):void

- drawHeartsList(canvas: Canvas): void

context:Context):void

+ updateManager(): void

BuilderFactory

GameManager <<interface>> Runnable - width:int - height:int + *run(): void* + GameManager(width:int,height:int) + getWidth():int + setWidth(width:int) + getHeightt():int + setHeight(height:int) GameThread + drawManager(canvas:Canvas) - surfaceHolder: final SurfaceHolder + updateManager() - canvas: Canvas - gameView: GameView - isRunning: boolean + GameThread(surfaceHolder:SurfaceHolder, gameView GameView) + getCanvas():Canvas Game1Manager + setCanvas(canvas:Canvas):void - stairsList: List<Stairs> + getSurfaceHolder():SurfaceHolder - gemList: List<Gem> + isRunning():boolean - collectedGems: List<Gem> + setRuuning(isRunning: boolean):void heartsList: List<Hearts> + setGameView(gameView GameView):void - collectedHearts: List<Hearts> + run():void - Game1Hero: game1Hero - background: Bitmap - game1StatusUpdater: Game1StatusUpdater + getBackground():Bitmap Game1Thread + getGame1Hero():Game1Hero + getLivesCount():int + Game1Manager(gridWidth:int, gridHeight:int, game1Builder:Game1Builder, context:Context)

Game1View)

+ Game1Thread(surfaceHolder:SurfaceHolder, view

**GameView** - screenWidth:int - screenHeight:int - scale:int - gameThread:GameThread - background: Bitmap - rect: Rect + GameView(context:Context) + GameView(context:Context. gameThread:GameThread) + getScreenWidth():int + getScale():int + getGameThread():GameThread + setGameThread(gameThread:GameThread):void + getThisBackground():Bitmap + setBackground(background: Bitmap) + getRect():Rect + setRect(rect:Rect):void + update():void + surfaceChanged(hoder:SurfaceHolder, format:int, width:int, height:int):void + surfaceCreated(holder:SurfaceHolder):void + surfaceDestroyed(holder:SurfaceHolder):void

Game1View

- game1Activity:Game1Activity

Game1Activity, level: String)

+ draw(canvas:Canvas):void

+ update():void

- game1Manager:Game1Manager

+ Game1View(context:Context, game1Activity

+ update(o:ObservableInterface, arg:Object):void