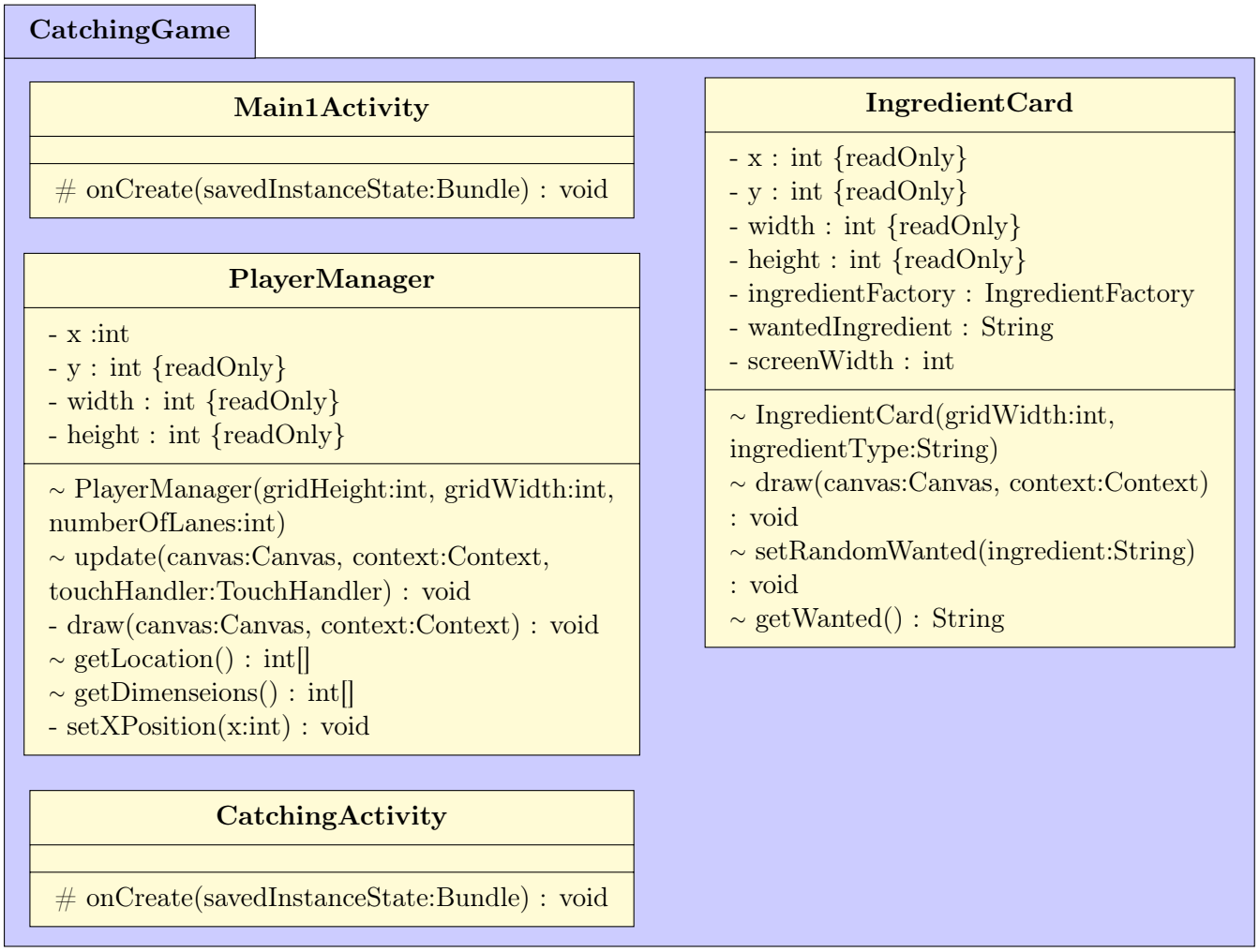
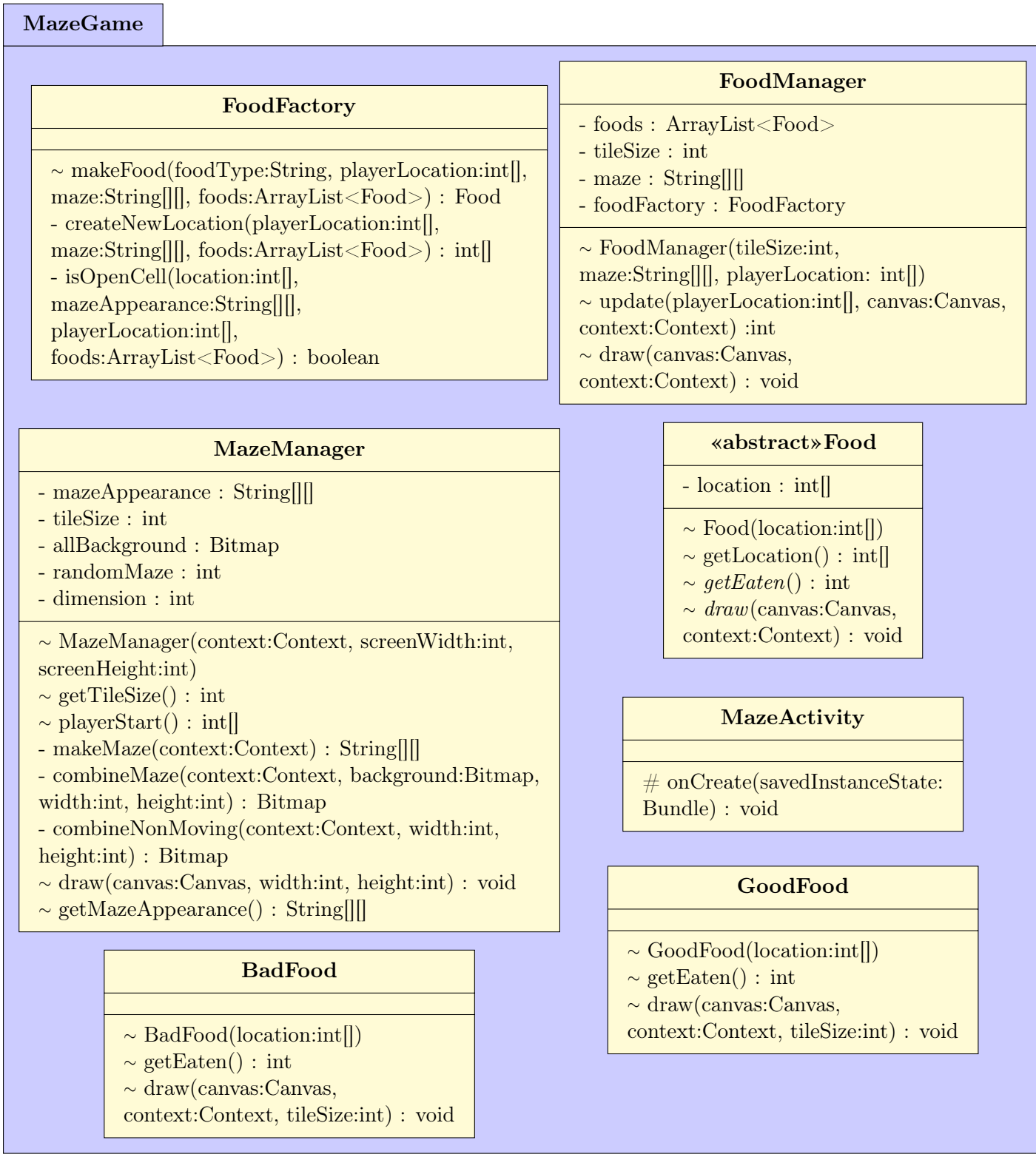
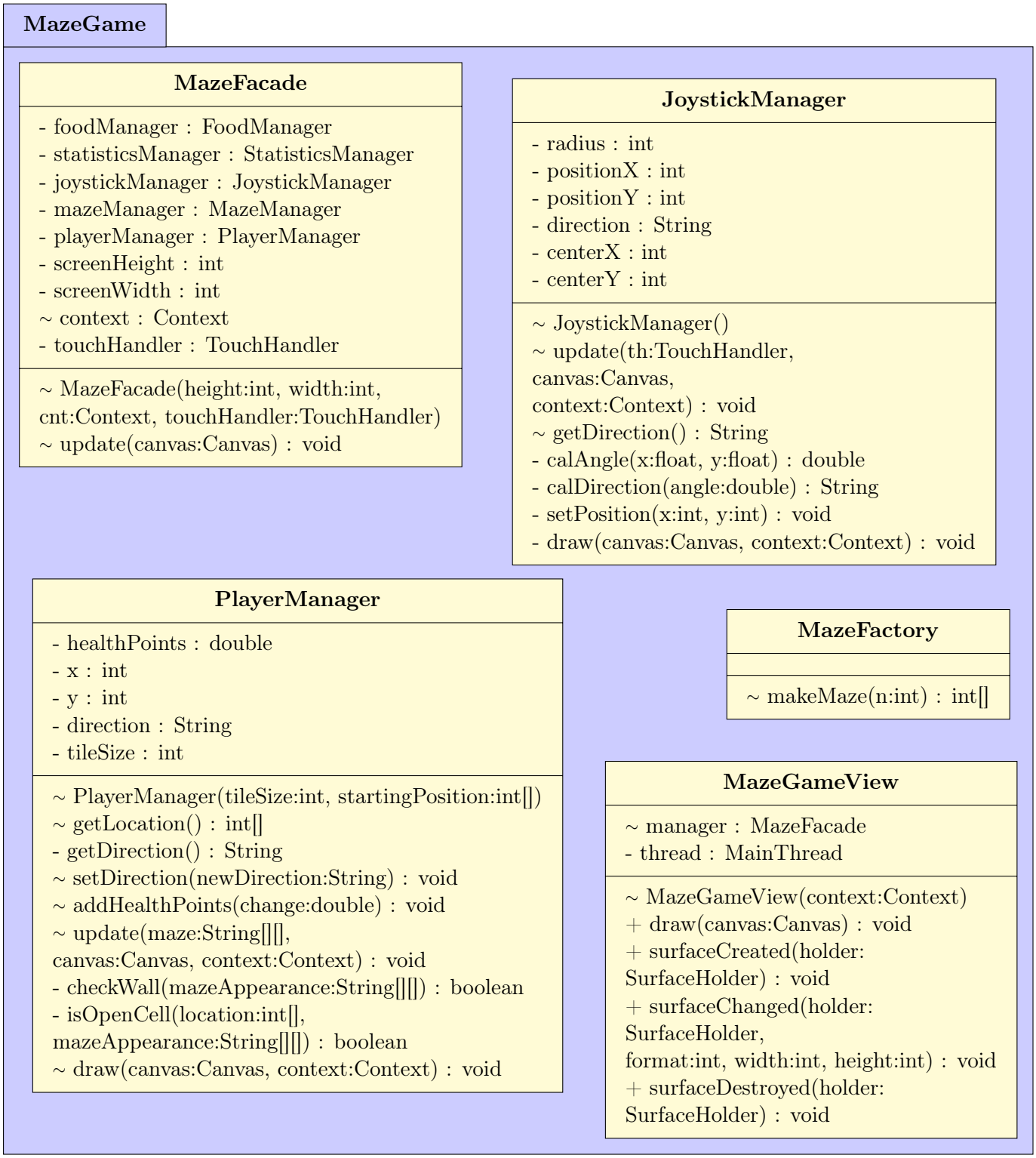
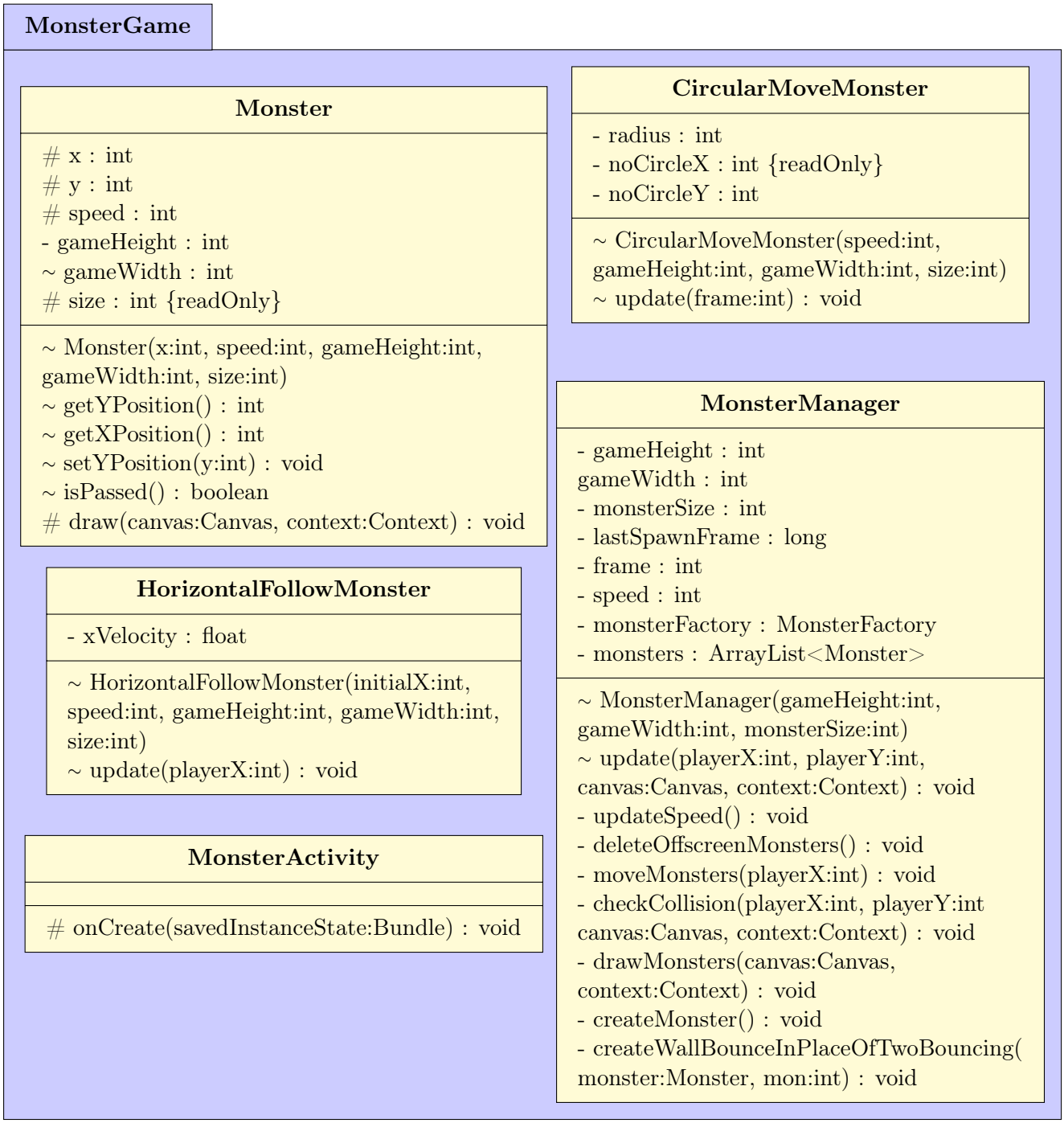
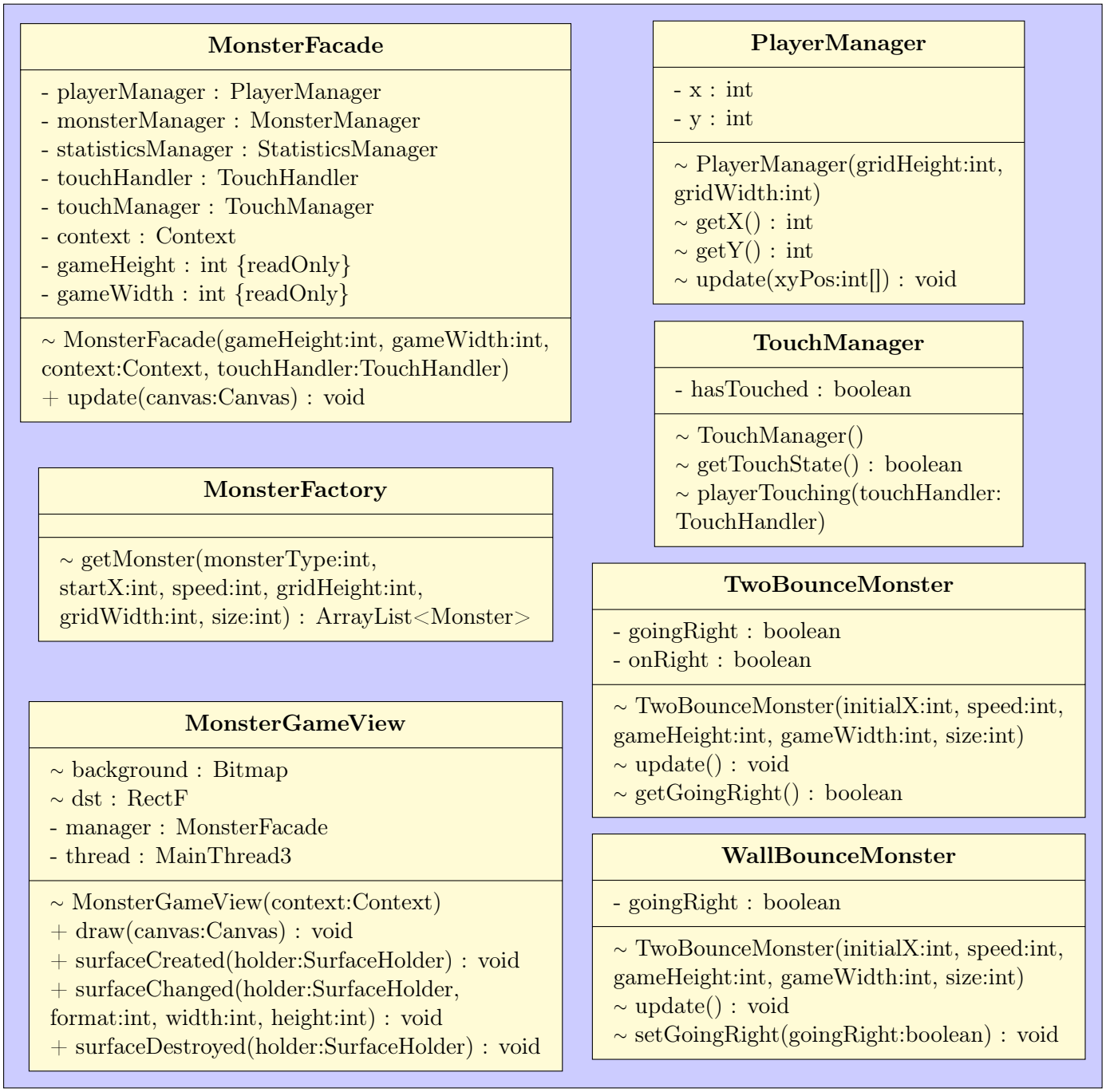

CatchingGame
CatchingFacade
- gridHeight : int
- gridWidth : int
- context : Context
- touchHandler : touchHandler
- ingredientManager : IngredientManager
- playerManager : PlayerManager
- statisticsManager : StatisticsManager
~ CatchingFacade(height:int, width:int, context:Context, touchHandler:TouchHandler)
+ update(canvas: Canvas) : void
Ingredient
- x : int {readOnly}
- y : int
- speed : int {readOnly}
+ name : String
~ Ingredient(x:int, speed:int, name:String)
~ getYPosition() : int
~ getXposition() : int
~ getName() : String
~ update() : void
CatchingGameView
+ catchingGameFacade : CatchingGameFacade
- thread : MainThread1
~ background : Bitmap
- dst : RectF
~ CatchingGameView(context:Context)
+ draw(canvas:Canvas) : void
+ surfaceCreated(holder:SurfaceHolder) : void
+ surfaceChanged(holder:SurfaceHolder, format:int, width:int, height:int) : void
+ surfaceDestroyed(holder:SurfaceHolder) : void
IngredientFactory
~ makeIngredient(x: int, speed:int, ingType: String)
~ getBitmap(ingredientName:String, context:Context) : Bitmap
IngredientManager
- ingredients : ArrayList<Ingredient>
- numberOfLanes : int {readOnly}
- lanes : int[]
- speed : int {readOnly}
- ingredientWidth : int {readOnly}
- ingredientHeight : int {readOnly}
- names : String[]
- ingredientFactory : IngredientFactory
- ingredientCard : IngredientCard
- frame : int
- lastSpawnFrame : int
~ IngredientManager(gameWidth:int, numberOfLanes:int)
~ update(canvas:Canvas, context:Context, playerLocation:int[], playerDimensions:int[]) : void
- createIngredient() : void
- moveIngredients() : void
- removableIngredients(yPlayer:int, heightPlayer:int) : void
- checkCollision(playerLocation:int[], playerDimensions:int[]) : void
- draw(canvas:Canvas, context:Context) : void

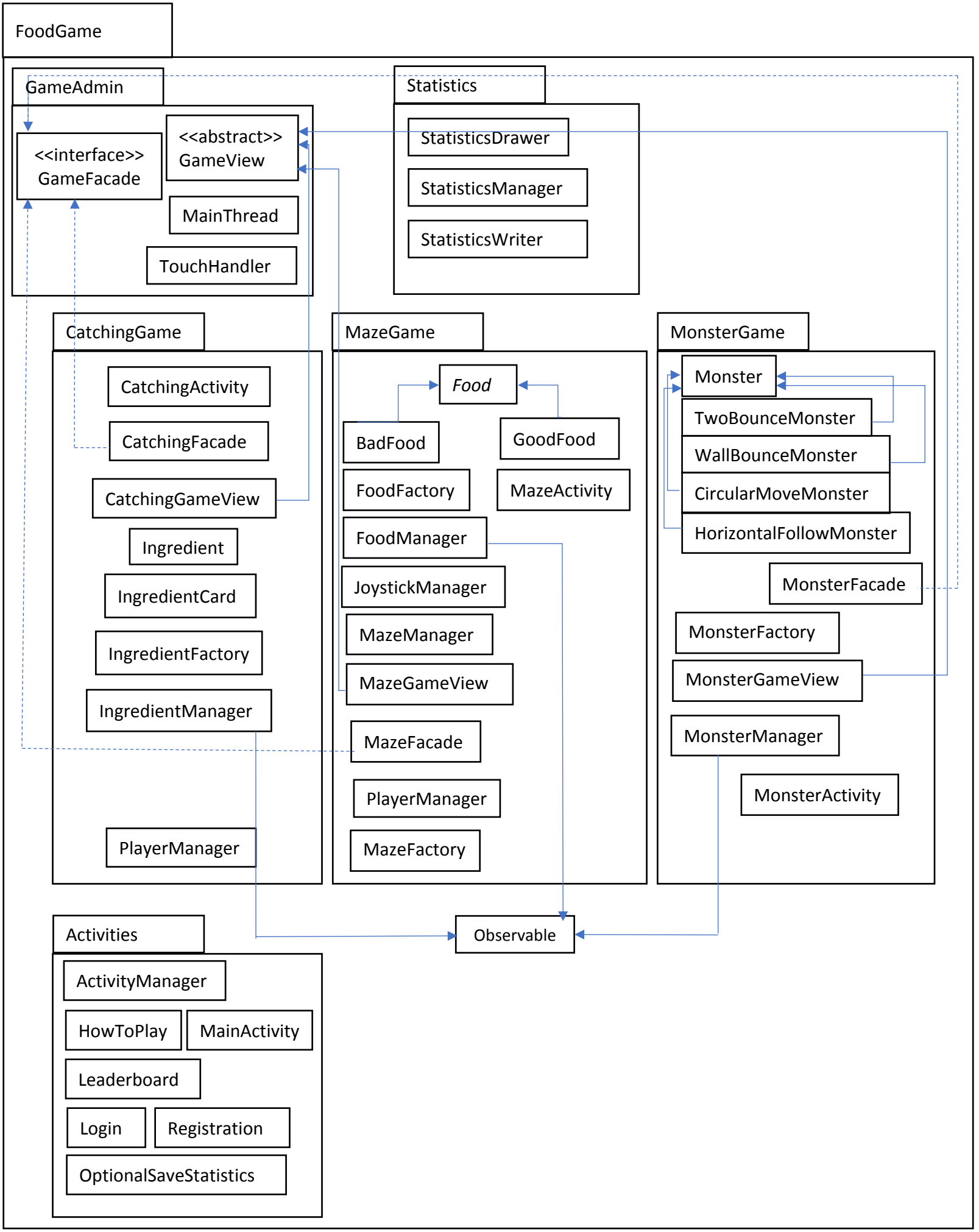

CatchingGame
Main1Activity
onCreate(savedInstanceState:Bundle) : void
PlayerManager
- x :int
- y : int {readOnly}
- width : int {readOnly}
- height : int {readOnly}
~ PlayerManager(gridHeight:int, gridWidth:int, numberOfLanes:int)
~ update(canvas:Canvas, context:Context, touchHandler:TouchHandler) : void
- draw(canvas:Canvas, context:Context) : void
~ getLocation() : int[]
~ getDimenseions() : int[]
- setXPosition(x:int) : void
CatchingActivity
onCreate(savedInstanceState:Bundle) : void
IngredientCard
- x : int {readOnly}
- y : int {readOnly}
- width : int {readOnly}
- height : int {readOnly}
- ingredientFactory : IngredientFactory
- wantedIngredient : String
- screenWidth : int
~ IngredientCard(gridWidth:int, ingredientType:String)
~ draw(canvas:Canvas, context:Context) : void
~ setRandomWanted(ingredient:String) : void
~ getWanted() : String


MazeGame
FoodFactory
~ makeFood(foodType:String, playerLocation:int[], maze:String[][], foods:ArrayList<Food>) : Food
- createNewLocation(playerLocation:int[], maze:String[][], foods:ArrayList<Food>) : int[]
- isOpenCell(location:int[], mazeAppearance:String[][], playerLocation:int[], foods:ArrayList<Food>) : boolean
FoodManager
- foods : ArrayList<Food>
- tileSize : int
- maze : String[][]
- foodFactory : FoodFactory
~ FoodManager(tileSize:int, maze:String[][], playerLocation: int[])
~ update(playerLocation:int[], canvas:Canvas, context:Context) :int
~ draw(canvas:Canvas, context:Context) : void
MazeManager
- mazeAppearance : String[][]
- tileSize : int
- allBackground : Bitmap
- randomMaze : int
- dimension : int
~ MazeManager(context:Context, screenWidth:int, screenHeight:int)
~ getTileSize() : int
~ playerStart() : int[]
- makeMaze(context:Context) : String[][]
- combineMaze(context:Context, background:Bitmap, width:int, height:int) : Bitmap
- combineNonMoving(context:Context, width:int, height:int) : Bitmap
~ draw(canvas:Canvas, width:int, height:int) : void
~ getMazeAppearance() : String[][]
«abstract»Food
- location : int[]
~ Food(location:int[])
~ getLocation() : int[]
~ getEaten() : int
~ draw(canvas:Canvas, context:Context) : void
MazeActivity
onCreate(savedInstanceState: Bundle) : void
GoodFood
~ GoodFood(location:int[])
~ getEaten() : int
~ draw(canvas:Canvas, context:Context, tileSize:int) : void
BadFood
~ BadFood(location:int[])
~ getEaten() : int
~ draw(canvas:Canvas, context:Context, tileSize:int) : void


MazeGame
MazeFacade
- foodManager : FoodManager
- statisticsManager : StatisticsManager
- joystickManager : JoystickManager
- mazeManager : MazeManager
- playerManager : PlayerManager
- screenHeight : int
- screenWidth : int
~ context : Context
- touchHandler : TouchHandler
~ MazeFacade(height:int, width:int, cnt:Context, touchHandler:TouchHandler)
~ update(canvas:Canvas) : void
JoystickManager
- radius : int
- positionX : int
- positionY : int
- direction : String
- centerX : int
- centerY : int
~ JoystickManager()
~ update(th:TouchHandler, canvas:Canvas, context:Context) : void
~ getDirection() : String
- calAngle(x:float, y:float) : double
- calDirection(angle:double) : String
- setPosition(x:int, y:int) : void
- draw(canvas:Canvas, context:Context) : void
PlayerManager
- healthPoints : double
- x : int
- y : int
- direction : String
- tileSize : int
~ PlayerManager(tileSize:int, startingPosition:int[])
~ getLocation() : int[]
- getDirection() : String
~ setDirection(newDirection:String) : void
~ addHealthPoints(change:double) : void
~ update(maze:String[][], canvas:Canvas, context:Context) : void
- checkWall(mazeAppearance:String[][]) : boolean
- isOpenCell(location:int[], mazeAppearance:String[][]) : boolean
~ draw(canvas:Canvas, context:Context) : void
MazeFactory
~ makeMaze(n:int) : int[]
MazeGameView
~ manager : MazeFacade
- thread : MainThread
~ MazeGameView(context:Context)
+ draw(canvas:Canvas) : void
+ surfaceCreated(holder: SurfaceHolder) : void
+ surfaceChanged(holder: SurfaceHolder, format:int, width:int, height:int) : void
+ surfaceDestroyed(holder: SurfaceHolder) : void


MonsterGame
Monster
x : int
y : int
speed : int
- gameHeight : int
~ gameWidth : int
# size : int {readOnly}
~ Monster(x:int, speed:int, gameHeight:int, gameWidth:int, size:int)
~ getYPosition() : int
~ getXPosition() : int
~ setYPosition(y:int) : void
~ isPassed() : boolean
# draw(canvas:Canvas, context:Context) : void
CircularMoveMonster
- radius : int
- noCircleX : int {readOnly}
- noCircleY : int
~ CircularMoveMonster(speed:int, gameHeight:int, gameWidth:int, size:int)
~ update(frame:int) : void
HorizontalFollowMonster
- xVelocity : float
~ HorizontalFollowMonster(initialX:int, speed:int, gameHeight:int, gameWidth:int, size:int)
~ update(playerX:int) : void
MonsterManager
- gameHeight : int
gameWidth : int
- monsterSize : int
- lastSpawnFrame : long
- frame : int
- speed : int
- monsterFactory : MonsterFactory
- monsters : ArrayList<Monster>
~ MonsterManager(gameHeight:int, gameWidth:int, monsterSize:int)
~ update(playerX:int, playerY:int, canvas:Canvas, context:Context) : void
- updateSpeed() : void
- deleteOffscreenMonsters() : void
- moveMonsters(playerX:int) : void
- checkCollision(playerX:int, playerY:int canvas:Canvas, context:Context) : void
- drawMonsters(canvas:Canvas, context:Context) : void
- createMonster() : void
- createWallBounceInPlaceOfTwoBouncing(monster:Monster, mon:int) : void
MonsterActivity
onCreate(savedInstanceState:Bundle) : void


MonsterFacade
- playerManager : PlayerManager
- monsterManager : MonsterManager
- statisticsManager : StatisticsManager
- touchHandler : TouchHandler
- touchManager : TouchManager
- context : Context
- gameHeight : int {readOnly}
- gameWidth : int {readOnly}
~ MonsterFacade(gameHeight:int, gameWidth:int, context:Context, touchHandler:TouchHandler)
+ update(canvas:Canvas) : void
PlayerManager
- x : int
- y : int
~ PlayerManager(gridHeight:int, gridWidth:int)
~ getX() : int
~ getY() : int
~ update(xyPos:int[]) : void
TouchManager
- hasTouched : boolean
~ TouchManager()
~ getTouchState() : boolean
~ playerTouching(touchHandler: TouchHandler)
MonsterFactory
~ getMonster(monsterType:int, startX:int, speed:int, gridHeight:int, gridWidth:int, size:int) : ArrayList<Monster>
TwoBounceMonster
- goingRight : boolean
- onRight : boolean
~ TwoBounceMonster(initialX:int, speed:int, gameHeight:int, gameWidth:int, size:int)
~ update() : void
~ getGoingRight() : boolean
MonsterGameView
~ background : Bitmap
~ dst : RectF
- manager : MonsterFacade
- thread : MainThread3
~ MonsterGameView(context:Context)
+ draw(canvas:Canvas) : void
+ surfaceCreated(holder:SurfaceHolder) : void
+ surfaceChanged(holder:SurfaceHolder, format:int, width:int, height:int) : void
+ surfaceDestroyed(holder:SurfaceHolder) : void
WallBounceMonster
- goingRight : boolean
~ TwoBounceMonster(initialX:int, speed:int, gameHeight:int, gameWidth:int, size:int)
~ update() : void
~ setGoingRight(goingRight:boolean) : void


FoodGame
GameAdmin
<<interface>>
GameFacade
<<abstract>>
GameView
MainThread
TouchHandler
Statistics
StatisticsDrawer
StatisticsManager
StatisticsWriter
CatchingGame
CatchingActivity
CatchingFacade
CatchingGameView
Ingredient
IngredientCard
IngredientFactory
IngredientManager
PlayerManager
MazeGame
Food
BadFood
GoodFood
FoodFactory
MazeActivity
FoodManager
JoystickManager
MazeManager
MazeGameView
MazeFacade
PlayerManager
MazeFactory
MonsterGame
Monster
TwoBounceMonster
WallBounceMonster
CircularMoveMonster
HorizontalFollowMonster
MonsterFacade
MonsterFactory
MonsterGameView
MonsterManager
MonsterActivity
Activities
ActivityManager
HowToPlay
MainActivity
Leaderboard
Login
Registration
OptionalSaveStatistics
Observable