

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
```

HorizontalFollowMonster

```
- xVelocity : float
```

```
~ HorizontalFollowMonster(initialX:int,
speed:int, gameHeight:int, gameWidth:int,
size:int)
~ update(playerX:int) : void
```

MonsterActivity

```
# onCreate(savedInstanceState:Bundle) : void
```

CircularMoveMonster

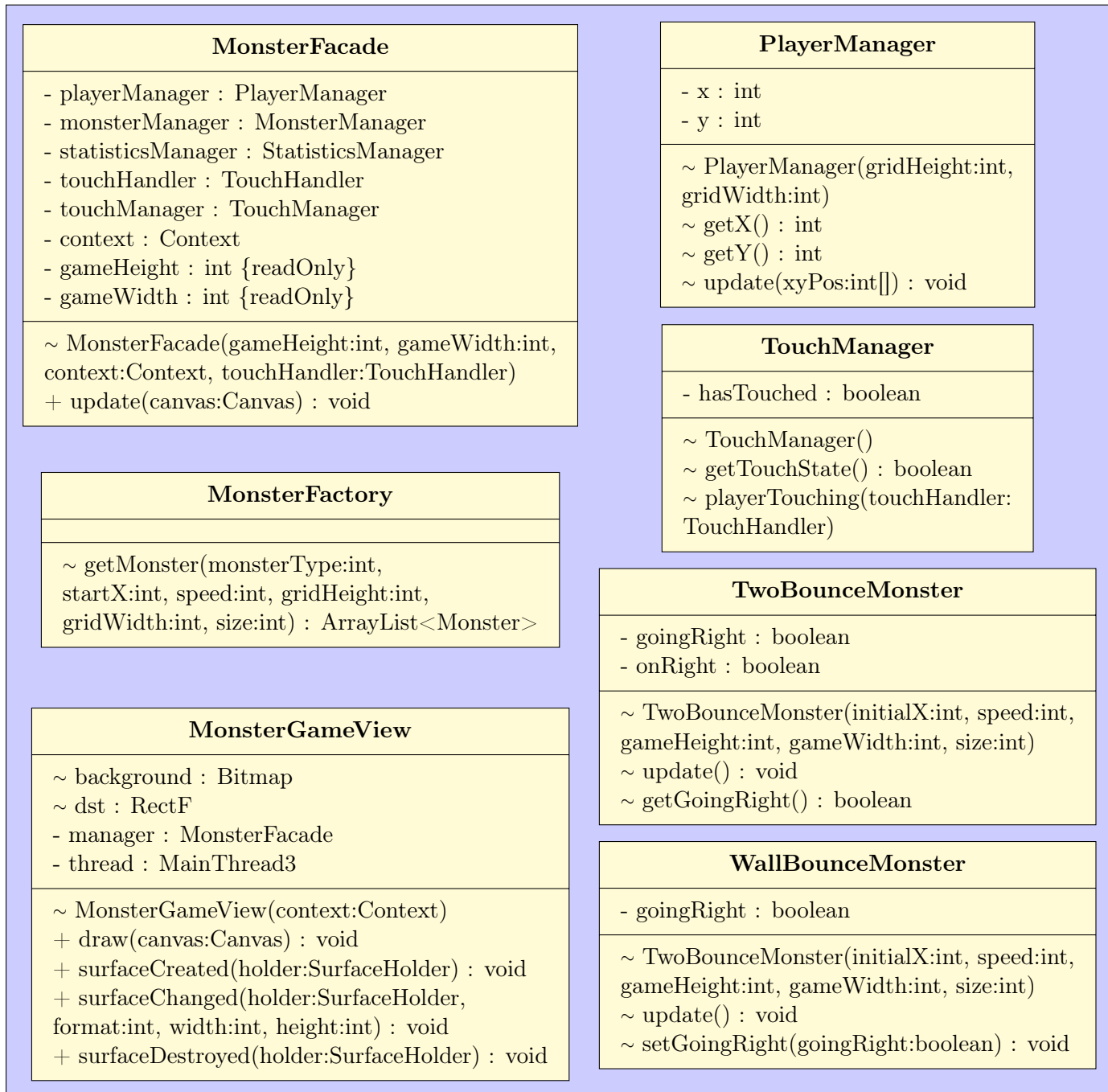
```
- radius : int
- noCircleX : int {readOnly}
- noCircleY : int
```

```
~ CircularMoveMonster(speed:int,
gameHeight:int, gameWidth:int, size:int)
~ update(frame: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
```



+ **MainActivity** extends AppCompatActivity
+ fields
- constructors
+ methods

+ **Login** extends AppCompatActivity
- fields
- username:EditText
- password:EditText
- message:TextView
- constructors
- methods
onCreate (savedInstanceState: Bundle):void
- openNextActivity(user:String):void
- openHowToPlay():void

+ **HowToPlay** extends AppCompatActivity
- fields
- constructors
- methods
onCreate (savedInstanceState: Bundle):void

+ **Registration** extends AppCompatActivity
- fields
- username:EditText
- password:EditText
- name:EditText
- email:EditText
- constructors
- methods
onCreate (savedInstanceState: Bundle):void

+ **ActivityManager**
- fields
- constructors
- methods

+ **OptionalSaveStatistics** extends AppCompatActivity
- fields
- constructors
- methods
onCreate (savedInstanceState: Bundle):void
- nextActivity():void

+ **Leaderboard** extends AppCompatActivity
- fields
- constructors
- methods
onCreate (savedInstanceState: Bundle):void
- setLayoutContent (usernames:ArrayList<String>, statisticsStrings:ArrayList<String>, statisticsNames:String[], linearLayout:LinearLayout):void

```

+ StatisticsWriter
  implements Observer
- fields -----
- roundStatistics: int[]
- gameOrder: int[]
- gamesPlayed: int
- final totalGames: int
- final statisticsNames: String[]
- context: Context
- usernames: ArrayList<String>
- currentUser: int
- activityClasses: ArrayList<Class>
- numberOfUsers: int
+ gameClasses: ArrayList<Class>
+ loading: boolean
- constructors -----
+ StatisticsWriter (totalGames: int, statisticsNames: String[], gameClasses: ArrayList<Class>, numberOfUsers: int)
- methods -----
+ update (obs: Observable, gameStatisticsObject: Object): void
- writeStatistics (username: String): void
- writeTemporaryStatistics (username: String): void
+ loadStatistics (username: String): void
+ writeScoreboardStatistics (): void
- writeScoreboardStatistics (username: String): void
+ getLastGame (username: String): int
+ getGames (username: String): ArrayList<Class>
+ getUserSavedState (username: String): ArrayList<Object>
+ addUsername (username: String): void
+ getCurrentUsername (): String
+ getStatisticsNames (): String[]
+ getUsernames (): ArrayList<String>
+ getActivityClasses (): ArrayList<Class>
+ getContext (): Context
+ setContext (context: Context): void
+ getNumberOfUsers (): int
+ getTotalGames (): int

```

```

+ StatisticsDrawer
- fields -----
- constructors -----
- methods -----

```

```

+ StatisticsManager extends Observable
  implements Observer
- fields -----
- lives: int
- score: int
+ startTime: long
- context: Context
- statisticsNames: String[]
- hasSentStatistics: boolean
- drawLoadingBackground: boolean
- constructors -----
+ StatisticsManager (context: Context)
- methods -----
+ update (obs: Observable, statisticObject: Object): void
- gameOver (): void
+ update (canvas: Canvas, screenHeight: int, screenWidth: int): void
- getStatistics (): int[]
- getStatisticsNames (): String[]

```


+ **TouchHandler**

[-] fields

- x:int
- y:int
- holding:boolean

[-] constructors

[-] methods

- + setPosition (arg:MotionEvent):void
- + getPosition ():int[]
- ~ setHolding(holding:boolean):void
- + getHolding():boolean

+ **MainThread** extends Thread

[-] fields

- gameView :GameView
- final surfaceHolder:SurfaceHolder
- + touchHandler:TouchHandler
- isRunning:boolean

[-] constructors

- + MainThread (surfaceHolder:SurfaceHolder, gameView: GameView)

[-] methods

- + run():void
- + setRunning(isRunning:boolean):void

+ **GameView** extends *SurfaceView* implements *SurfaceHolder.Callback*

[-] fields

[-] constructors

- + GameView (context:Context)

[-] methods

- + draw (canvas:Canvas):void

+ **GameFacade**

[-] fields

[-] methods

- + update (canvas:Canvas):void

