## **Utilities:**

# DrawUtility - paint: Paint

- DrawUtility()
- + drawRectangle(coords: int[], color: int)
- + drawLines(coords: int[], color: int, width: int)
- + drawString(text: String, x: int, y: int, color: int, size: int): void
- + drawBitmap(bitmap: Bitmap, x: int, y: int): void

## Session

## - prefs: SharedPreferences

- Session()
- + setupSession(context: Context)
- checkSession(context: Context)
- + setUser(username: String, password: String): void
- + clearUser(): void
- + getUser(): String[]
- + setMusic(id: int): void
- + getMusic(context: Context): int
- + setBackground(id: int); void
- + getBackground(context: Context): int
- + setTheme(on: boolean): void
- + getTheme(context: Context): boolean

### FileStorage

- fileName: String
- ALL USERS FILE NAME:
- String
- DATA SEPARATOR: String
- FileStorage()
- + saveUserData(dirPath: String, userToSave: UserAccount): void
- + getUserData(dirPath: String): UserAccount
- removeInventoryItems(jsonData: String): String
- + saveUserToFile(dirPath: String, name: String, password: String): boolean
- + getAllUsersFromFile(context: Context): List<UserAccount>
- warnException(e: Exception)

#### HandleCustomization

- themeOn: boolean
- player: MediaPlayer
- HandleCustomization()
- + setActivityBackground(context: Context, window: Window): void
- getBackgroundDrawable(context: Context): int
- + setGameBackground(context: Context, layout: View): void
- setDefaultBackground(view: View): void
- + startMusic(context: Context): void
- + pauseMusic(context: Context): void + getPixelLabelColor(): int

## Achievements & Statistics:

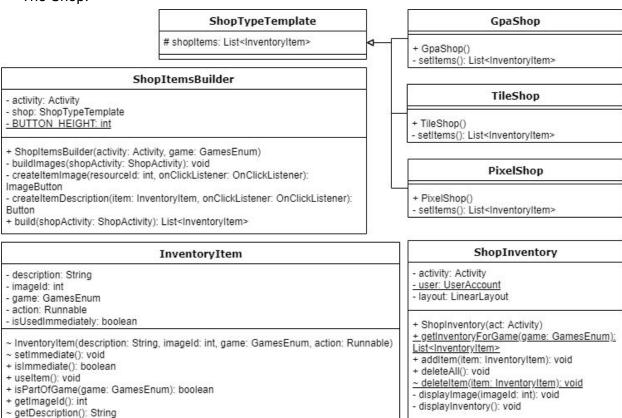
## Achievements

- maxGpaAchieve: boolean
- threeGpaAchieve: boolean
- \_failGpaAchieve: boolean
- numPixelTaps: int
- numPixelTapsFiftyCompleted: boolean
- \_numPixelTapsTwoHundredCompleted: boolean
- \_twentySecondsOrLessPixel: boolean
- \_numRotateTaps: int
- \_numRotateTapsFiftyCompleted: boolean
- \_numRotateTapsTwoHundredCompleted: boolean
- twentySecondsOrLessRotate: boolean
- Achievements()
- + \_gpaChecker(gpa: double): void
- ~ \_checkTime(gamesEnum: GamesEnum, elapsedTime: long): void
- + isMaxGpaAchieve(): boolean
- + \_isThreeGpaAchieve(): boolean
- + isFailGpaAchieve(): boolean
- + \_getNumPixelTaps(): int
- + \_setNumPixelTaps(): void
- + \_isTwentySecondsOrLessPixel(): boolean
- + \_setTwentySecondsOrLessPixel(): void
- + \_getNumRotateTaps(): int
- + \_setNumRotateTaps(): void
- + \_isTwentySecondsOrLessRotate(): boolean
- + \_setTwentySecondsOrLessRotate(): void
- + \_isNumPixelTapsTwoHundredCompleted(): boolean
- + \_isNumRotateTapsFiftyCompleted(): boolean
- + \_isNumRotateTapsTwoHundredCompleted(): boolean
- + isNumPixelTapsFiftyCompleted(): boolean

#### Statistics

- \_startTime: long
- \_startTimePixel: long
- \_startTimeRotate: long
- \_timeElapsedSeconds: long
- -\_totalScore: int
- \_counterClicks: int
- Statistics()
- + \_start(): void
- + \_startTimeByGame(gamesEnum: GamesEnum): void
- + \_end(): void
- + \_endTimeByGame(gamesEnum: GamesEnum): void
- \_addRandomScore(): int
- + \_sumTotalScore(): void
- + \_printStats(): String + clickEvent(): void
- + \_totalClicks(): int
- + \_printStats(): String

## The Shop:



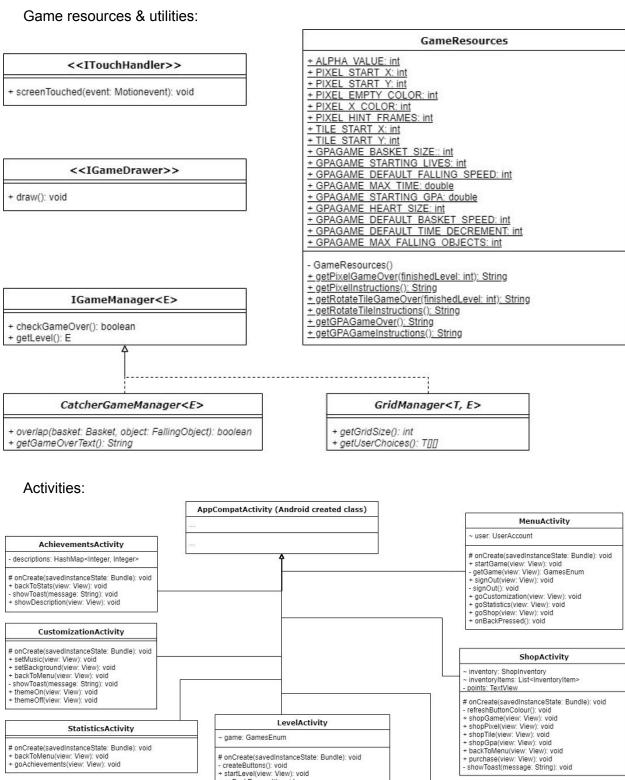
## Users:

## UserAccount

- name: String
- password: String
- pixelUnlocked: int
- tileUnlocked: int
- inventoryItems: List<InventoryItem>
- userPoints: int
- + UserAccount(name: String, password: String)
- ~ checkCreds(name: String, password: String): boolean
- ~ sameUsername(name: String): boolean
- + addUnlocked(game: GamesEnum): void
- + getUnlocked(game: GamesEnum): int
- + getName(): String
- + addInventoryItem(item: InventoryItem): void
- + removeInventoryItem(item: InventoryItem): void
- + getInventoryItems(): List<InventoryItem>
- + addUserPoints(pointsToAdd: int): void
- + getUserPoints(): int

## UserAccountManager

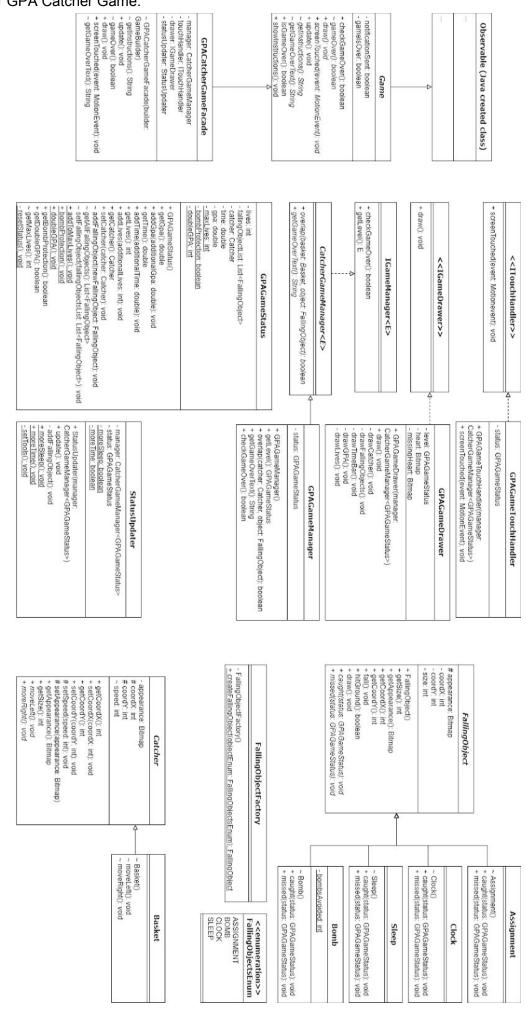
- + currentUser: UserAccount
- users: List<UserAccount>
- userFileDirectory: String
- context: Context
- + UserAccountManager(context: Context)
- + signIn(username: String, password: String): UserAccount
- + signUp(username: String, password: String): boolean
- ~ \_updateUser(): void
- + \_signOut(): void
- restorePreviousUsers(): void
- saveUserToFile(name: String, password: String, user: UserAccount): boolean
- + notSignedIn(): boolean



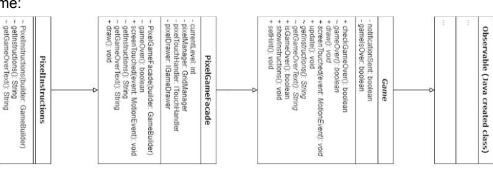
# + onBackPressed(): void GameActivity implements Observer LoginActivity panel: Panel currentGame: GamesEnum level: int IntentExtras userManager: UserAccountManager ~ gameEnum: String password: EditText levelNumber: String # onCreate(savedInstanceState: Bundle): void # onStart(): void - setupGame(): void # onCreate(savedInstanceState: Bundle): void + signIn(view: View); void setCurrentGame(): void setHintButtons(game: GamesEnum): void pauseButtonPressed(view: View): void + signUp(view: View): void + onBackPressed(): void resumePressed(view: View); void showOverlay(text: String, gameIsOver: boolean); void - update(observable: Observable, o: Object); void Observer (Java created class) + onBackPressed(): void + backToMain(view: View): void + backToLevels(view: View): void

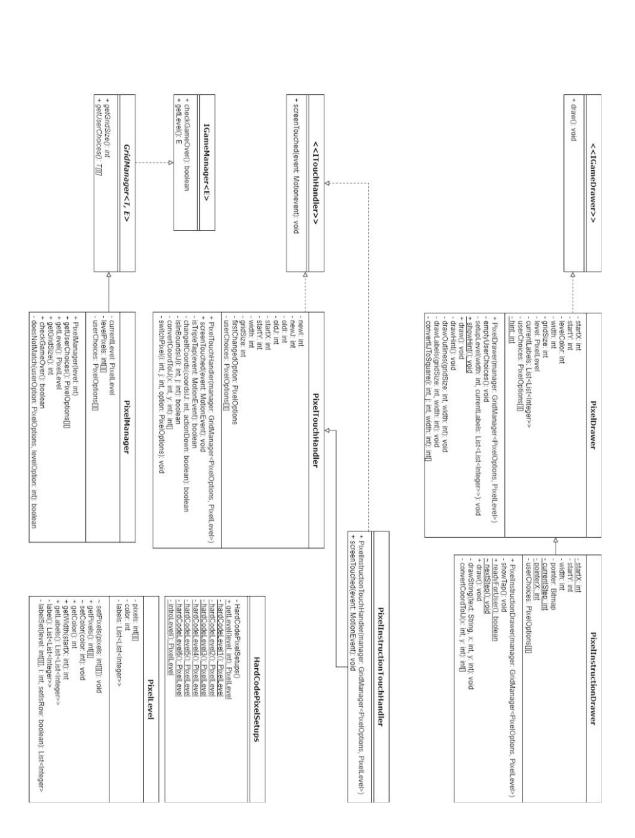
## Our Rotate Tile Game: + update(): void - get/nstructions(): String - get/GameOverText(): String + isGameOver(): boolean + showInstructions(): void -- gameOver(): boolean -- screenTouched(event MotionEvent): void -- getInstructions(): String --- getGameOverText(): String --- getGameOverText(): String ---- draw(): void + screenTouched(event: MotionEvent): void + draw0: void - gameOver(): boolean RotateTileGameFacade(builder: GameBuilder) -manager: GridManager -touchHandler: iTouchHandler -drawer: IGameDrawer + setHint(): void notificationSent: boolean gameIsOver: boolean Observable (Java created class) RotateTileGameFacade + checkGameOver(): boolean + getLevel(): E Motionevent): void - pointer: Bitmap - currentStep: int - userChoices: Tile[][] - length: int + Rotate TileInstructionDrawer(manager GridManager<Tile, TileLevel>) pointerX: int pointerY: int width: int draw(): void getGridSize(): int screenTouched(event <<ITouchHandler>> drawString(text: String, x: int, y: int): void nextStep(): void <<IGameDrawer>> GridManager<T, E> IGameManager<E> RotateTileInstructionDrawer + gett.evel(): Tilet.evel - gett.devel(): Tilet.evel - gett.devel(): tilets(lim): void - tandomize Tilets(lim): void - tandomize Tilets(lim): boolean - add.evel(): void - gameOver(x: int, y: int, exits: mt): boolean - calculateNewCoord(coord: int, exit: int, isY: int): int - aEInd(x: int, y: int; exits: mt)): boolean - valid(x: int, y: int; boolean - allStageList\_List<TileLevel> - currentLevel: TileLevel - userChoices: Tile[]] - levelToPlay: int TileLevel>) + screenTouched(event MotionEvent); void - screenTouched(x.int, y.int); void + setFreeTile(tile: char); void - gridSize: int - width: int - startX: int - startY: int - userChoices: Tile[II] - hintTile: char - startX: int - startY: int - startEndPipe: Tile + RotateTileDrawer(manager, GridManager<Tile,TileLevel>) - createStartEndTile(); void + draw(); void + TileManager(levelToPlay: int) - loadStage(difficulty: DifficultyEnum, gridLength: int): void + getUserChoices(): Tile[II] - gridSize: int - userChoices: Tile[][] width: int RotateTileTouchHandler(manager: GridManager<Tile - pastLocation: int[]] - gridLength: int - difficulty: DifficultyEnum - path ToFinish: List</ri> - FindSouthon(start int]], end int], gridLength, int, difficulty. DifficultyEnum) - isOnFinishGrid(position: int]]): boolean - path(location: int]], pastLocation: int]], steps: int) boolean - isDifficultEnough(steps: int), boolean - isDifficultEnough(steps: int), boolean - hasDeenToBefore(position: int]]): boolean - outORange(location: int]): boolean - calculateNewCoord(coord: int], move: int): void - gelPathToFinish(), List-integer> RotateTileTouchHandler RotateTileDrawer FindSolution Tile(exits: intt], originalimage: Bitmap) resize(newDimension: int); void gelExit(s), intl gelExit(s) intl ge + setTiles(files: char(III)) void + setTiles(Tiles(Tiles(III)) void - setTiles(Tiles(Tiles(III)) void - getTiles(Voidh(startX: int): int - getCiles(Xize(): int) - convertChar(O'Tile(input char(III)): Tile(III - resizeUserChoices(width: int): void EASY MEDIUM HARD <<enumeration>> DifficultyEnum + visited: boolean + rotatedImage: Bitmap - originalImage: Bitmap - exits: int[] - rotation: Rotation <<Cloneable>> (Java created class) level: Tile[]] tileWidth: int rotateBitmap(angle: float): void TileLevel Tile + RandomLevelCeneration(pulder, RotateStageBuilder) - makeStage(start.int[]. end.int[]). void - makePath(entry.int.lastExit.int). void - addPipe(diff.int). Tile - fillBlanks(). void - fillBlanks(). void - caliculateNewCoord(cord: int[], mover.int): void setGridLength(gridLength: int): RotateStageBuilder setDifficulty(difficulty: DifficultyEnum): RotateStageBuilder + makeLevel(): TileLevel stage: Tile[][] gridLength: int difficulty: DifficultyEnum pathToFinish: List<Integer> gridLength: int difficulty: DifficultyEnum construct(): TileLevel RandomLevelGeneration RotateStageBuilder ANY - value: char - map: Map + yalueOf(tileType: char): TileEnum + getValue(): char ~ LTile() ~ TTile() - CrossTile() StraightTile() <<enumeration>> TileEnum StraightTile CrossTile Tile LTile NORTH EAST SOUTH WEST - value: int ~ Rotation(value: int) +\_getRandom(): Rotation +\_valueOf(degrees: int): Rotation + getValue(): int TileFactory() +\_createTile(tileToCreate: char): Tile +\_createTile(tileToCreate: TileEnum): Tile <<enumeration>> Rotation TileFactory

# Our GPA Catcher Game:

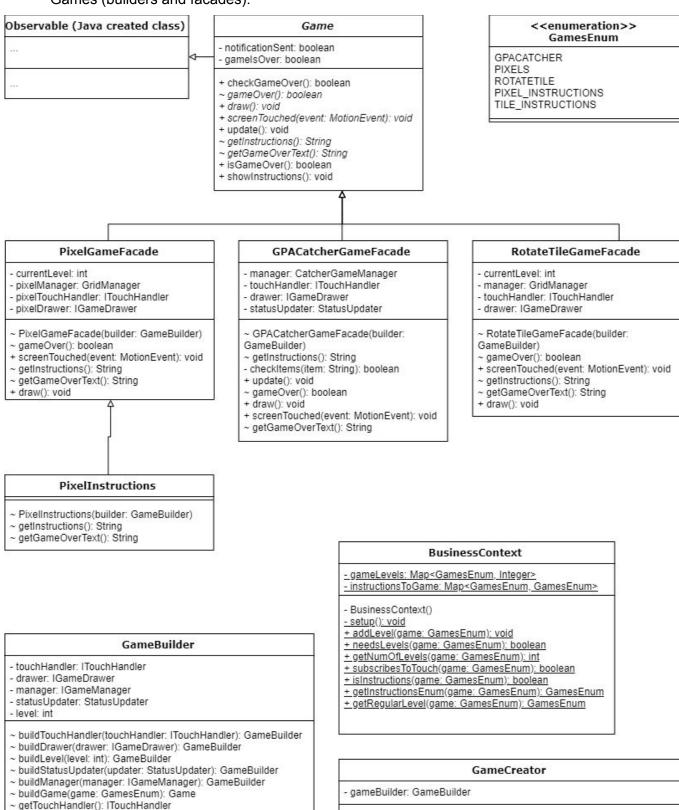


# Our Pixel Game:





## Games (builders and facades):



- + createGame(gameToCreate: GamesEnum): Game
- + createGame(gameToCreate: GamesEnum, level: int): Game
- createTileInstructions(gameToCreate: GamesEnum, level: int): Game createPixeIInstructions(gameToCreate: GamesEnum, level: int): Game

- ~ getLevel(): int
- ~ getStatusUpdater(): StatusUpdater

~ getDrawer(): IGameDrawer

~ getManager(): IGameManager