Shooter Game Over

<<Interface>>
Shooter Game Over View

~ nextLevelAppear(): void
~ nextLevelGone(): void
~ startMusic(): void
~ stopMusic(): void
~ setGameText(message: String): void
~ startNewLevel(): void
~ backToMenu(): void

Use

Shooter Game Over Presenter

- shooterGameOverLogic: ShooterGameOverLogic
- shooterGameOverView: ShooterGameOverView

~ ShooterGameOverPresenter(shootergameoverlogic: ShooterGameOverLogic, shootergameoverview: ShooterGameOverView)
~ checkNextLevelAppear(): void
~ checkMusicStart(): void
~ setUpGameMessage(): void
~ checkMusicStop(): void
~ handleNextLevel(): void
~ handleBackToMain(): void

Shooter Game Over

- next: Button
- message: TextView
- shooterGameStatus: ShooterGameStatusFacade
- shooterGameOverPresenter: ShooterGameOverPresenter
- backToMain: Button
- backToMenu: Button

onCreate(savedInstanceState Bundle): void
onResume(): void
onPause(): void
+ onClick(): void

Shooter Game Over Logic

- shooterGameStatus: ShooterGameStatusFacade
- shooterCrossLevelManager: ShooterCrossLevelManager
- level: int
- musicFinish: boolean
- gameSuccess: boolean
- point: int
- levelFinish: boolean

~ ShooterGameOverLogic(shooterGameStatus: ShooterGameStatusFacade)
~ checkNextLevelAppear(): boolean
~ getText(): String
~ setMusicFinish(finish: boolean): void
~ getMusicFinish(): boolean
~ levelUpGameStatus(): void

shooter plane game

<<Interface>>
Shooter Game Interface

~ startMusic(): void
~ stopMusic(): void
~ dismissDialog(): void
~ openDialog(): void
~ makeBonusWinToast(): void
~ makeBonusLoseToast(): void

Use

Shooter Plane Game Logic

- shooterGameStatus: ShooterGameStatusFacade
- context: Context
- shooterGameView: ShooterGameView
- bonusOpen: boolean
- start: int

~ ShooterPlaneGameLogic(shooterGameStatus: ShooterGameStatusFacade, shooterGameView: ShooterGameView)
~ shouldMusicStop(): boolean
- saveGameState(): void
~ addBonusPoint(): void
~ handleOnResume(): void
~ handleOnPause(): void
~ isBonusOpen(): boolean
- setBonusOpen(): void
~ handleActivateBonusGame(): void
~ handleCancelBonus(): void

Class

~ gameView: shooterGameView
~ shooterGameStatus: ShooterGameStatusFacade
~ shooterGamePresenter: ShooterGamePresenter
- dialog: BonusLevelDialog

onCreate(savedInstanceState Bundle): void
onResume(): void
onPause(): void
+ onClick(): void
+ activateBonusGame(): void
+ openDialog(): void

Shooter Game Presenter

- shooterGameInterface: ShooterGameInterface
- shooterPlaneGameLogic: ShooterPlaneGameLogic

~ ShooterGamePresenter(shooterGameInterface: ShooterGameInterface, shooterPlaneGameLogic: ShooterPlaneGameLogic)
~ handleOnResume(): void
~ handleOnPause(): void
~ activateBonusGame(): void
~ handleBonusGameResult(): void
~ cancelBonus(): void

Shooter Game View

Shooter Bitmap Manager

- context: Context
- shooterGameStatus: ShooterGameStatusFacade
- shooterPlaneBullets: List<ShooterPlaneBullet>
- shooterBonuses: List<ShooterBonus>
- healthAids: List<ShooterHealthAid>
- pointBuffs: LIst<ShooterPointBuff>
- shooterEnemyBullets: List<ShooterEnemyBullet>
- enemyExplosions: List<ShooterEnemyExplosion>
- planeExplosions: List<ShooterPlaneExplosion>
- shooterEnemies: List<ShooterEnemy>
- plane: ShooterPlane

~ ShooterBitmapManager(context: Context,
shooterGameStatus ShooterGameStatusFacade)
- loadManager(): void
~ loadBitmap(): void
- loadExplosion(): void
- loadListGameObject(): void

Shooter Draw Item Manager

- shooterGameStatus: ShooterGameStatusFacade
- shooterPlaneBullets: List<ShooterPlaneBullet>
- shooterBonuses: List<ShooterBonus>
- healthAids: List<ShooterHealthAid>
- pointBuffs: LIst<ShooterPointBuff>
- shooterEnemyBullets: List<ShooterEnemyBullet>
- enemyExplosions: List<ShooterEnemyExplosion>
- planeExplosions: List<ShooterPlaneExplosion>
- shooterEnemies: List<ShooterEnemy>
- plane: ShooterPlane
- scorePaint: Paint
- levelPaint: Paint
- healthPaint: Paint
- canvas: Canvas

~ ShooterDrawItemManager(
shooterGameStatus ShooterGameStatusFacade)
~ setCanvas(canvas: Canvas): void
- setUpManager(): void
~ draw(): void
- setPaint(): void
- drawText(): void
- drawGameObject(): void
- drawExplosion(): void

Shooter Collision Manager

- context: Context
- level: int
- shooterGameStatus: ShooterGameStatusFacade
- shooterPlaneBullets: List<ShooterPlaneBullet>
- shooterBonuses: List<ShooterBonus>
- healthAids: List<ShooterHealthAid>
- pointBuffs: LIst<ShooterPointBuff>
- shooterEnemyBullets: List<ShooterEnemyBullet>
- enemyExplosions: List<ShooterEnemyExplosion>
- planeExplosions: List<ShooterPlaneExplosion>
- shooterEnemies: List<ShooterEnemy>
- plane: ShooterPlane
- sp: SoundPool

~ ShooterCollisionManager(context: Context,
shooterGameStatus ShooterGameStatusFacade,
sp SoundPool)
- setUpManager(): void
~ handleCollision(): void
- planeEnemyCollide(): void
- checkEnemyPlaneCollide(): boolean
- bullet1EnemyCollide(): void
- hitEnemy(): int
- planeSpecialItemCollision(): void
- planeSpecialItemCheck(): boolean
- bullet2PlaneCollide(): void
- bonusPlaneCollide(): void

Shooter Game View

~ enemyDown: int
~ bulletLoad: int
~ background: Bitmap
~ shooterCollisionManager: ShooterCollisionManager
~ loadItemManager: ShooterLoadItemManager
~ drawItemManager: ShooterDrawItemManager
~ bitmapManager: ShooterBitmapManager
~ plane: ShooterPlane
~ rect: Rect
~ context: Context
~ countDownTimer: CountDownTimer
~ handler: Handler
~ runnable: Runnable

+ ShooterGameView(context: Context)
+ setShooterGameStatus(
shooterGameStatus: ShooterGameStatusFacade): void
- setSoundPool(): void
- startTimer(): void
~ setBackground(): void
# onDraw(canvas Canvas): void
+ onTouchEvent(event MotionEvent): void
~ onViewFinish(finish: boolean): void
+ setActivityFinish(): void

Shooter Load Item Manager

- shooterGameStatus: ShooterGameStatusFacade
- shooterPlaneBullets: List<ShooterPlaneBullet>
- shooterBonuses: List<ShooterBonus>
- healthAids: List<ShooterHealthAid>
- pointBuffs: LIst<ShooterPointBuff>
- shooterEnemyBullets: List<ShooterEnemyBullet>
- enemyExplosions: List<ShooterEnemyExplosion>
- planeExplosions: List<ShooterPlaneExplosion>
- shooterEnemies: List<ShooterEnemy>

- setUpManager(): void
~ loadItem(): void
- updateSpecialItem(): void
- updateEnemy(): void
- updateEnemyBullet(): void
- updatePlaneBullet(): void
- updateBonuses(): void