Blood Red Skies

ViewController — Uses Singleton
- instance: ViewController
- + switchView(String): void

Model View Controller Pattern

MenuView
- + playButton: HTML <button>
- + viewController: ViewController
- + goToGame(): void

KeyboardHandler
- - wKeyDown: Boolean
- - aKeyDown: Boolean
- - sKeyDown: Boolean
- - dKeyDown: Boolean
- - rKeyDown: Boolean
- - spaceKeyDown: Boolean
- + handle(parent): void

GUI
- + healthImg: File
- + healthSprite: Sprite
- + timeImg: File
- + timeSprite: Sprite
- + ammoImg: File
- + ammoSprite: Sprite
- + scoreImg: File
- + scoreSprite: Sprite
- + healthTextStyle: PIXI.TextStyle
- + healthText: PIXI.Text
- + timeTextStyle: PIXI.TextStyle
- + timeText: PIXI.Text
- + ammoTextStyle: PIXI.TextStyle
- + ammoText: PIXI.Text
- + scoreTextStyle: PIXI.TextStyle
- + scoreText: PIXI.Text
- + updateHealthText(health): void
- + updateTimeText(time): void
- + updateAmmoText(ammo): void
- + updateScoreText(score): void

GameView
- + viewController: ViewController
- + audioPlayer: AudioPlayer
- + keyboardHandler: KeyboardHandl
- - mouseDown: Boolean
- - width: Float
- - height: Float
- + gameOver: Boolean
- + nonPlayerSprites: Sprite [ ]
- + ground: Ground
- + player: Player
- + bullets: Bullet [ ]
- + powerups: Powerup [ ]
- + monsters: Monster [ ]
- + gui: GUI
- + health: Int
- + time: Int
- + score: Int
- + highscore: Int
- + guns: Gun [ ]
- + currentGun: Gun
- + moveSpeed: Int
- + gotoGameOverView(): void
- + load(): void
- + stopGame(): void
- + clean(): void
- + reset(): void
- - loop(): void
- + increaseTime(): void
- + increaseScore(): void
- + bloodSplatter(): void
- + removeObject(): void
- - mouseDownListener(parent): void
- + shoot(): void
- + scrollGuns(): void
- + moveCamera(): void

GameOverView
- + viewController: ViewController
- + retryButton: HTML <button>
- + postButton: HTML <button>
- + scoreText: HTML <div>
- + highScoreText: HTML <div>
- + load(): void
- + gotoGameView(): void
- + gotoSubmitView(): void

SubmitView
- + viewController: ViewController
- + nameField: HTML <div>
- + submitButton: HTML <button>
- + randomUsername: String
- + load(): void
- + gotoLeaderboardView(): void
- + submitScore(): void
- + generateUsername(): void

LeaderboardView
- + viewController: ViewController
- + backButton: HTML <button>
- + table: HTML <div>
- + rows: (HTML <div>) [ ]
- + scoreCols: (HTML <div>) [ ]
- + nameCols: (HTML <div>) [ ]
- + load(): void
- + gotoMenuView(): void
- - createTable(): void
- - createRows(): void
- - createCols(): void
- - async populateTable(): void

Player
- - gameView: GameView
- + x: Float
- + y: Float
- + width: Int
- + width: Int
- + height: Int
- + radius: Int
- - img: String
- + sprite: Sprite
- + mouseAngle: Float
- - rotate(sprite, parent): void
- + takeDamage(): void

AudioPlayer
- + soundsDictionary: Dict
- + play(String): void
- + stop(String): void

Monster
- + width: Float
- + height: Float
- + radius: Float
- + frames: Sprite
- + sprite: Sprite
- - img: String
- - tintImg: String
- + tintSprite: Sprite
- + dead: Boolean
- + health: Int
- + speed: Int
- + attackingRunSpeed: Int
- + waitingToMove: Boolean
- + targettingPlayer: Boolean
- + targettingDistance: Int
- + maxWanderingDistance: Int
- + attackSpeed: Int
- + canAttack: Boolean
- + player: Player
- + respawn(): void
- + setPosition(): void
- + move(): void
- + hitTest(x, y): Boolean
- + hitTest(x, y): Boolean
- + hitTestCircle(x, y): Boolean
- + waitToMove(time): void
- + stopWaiting(parent): void
- + takeDamage(amount): void
- + die(): void
- + distanceTo(otherObject): void
- + rotateTowards(otherObject): void
- + attack(player): void
- + readyToAttack(parent): void

Bullet
- + width: Float
- + height: Float
- - img: String
- + sprite: Sprite
- + type: String
- + setAttributes(gunType): void
- + move(): void

Blood
- + width: Float
- + height: Float
- - img: String
- + sprite: Sprite
- + fade(): void

Gun <Abstract>
- - gameView: GameView
- + type: String
- + gunLength: Int
- + bulletSpeed: Int
- + bulletDamage: Int
- + bulletSize: Int
- + owned: Boolean
- + ammo: Int
- + autoReload: Bool
- + canReload: Bool
- + reloadSpeed: Int
- + resetGun(): void
- + shoot(): void
- + reload(): void
- + readyGun(): void

Strategy Pattern

PistolStrategy
- + type: String
- + gunLength: Int
- + bulletSpeed: Int
- + bulletDamage: Int
- + bulletSize: Int
- + owned: Boolean
- + ammo: Int
- + autoReload: Bool
- + reloadSpeed: Int
- + resetGun(): void
- + shoot(): void

RifleStrategy
- + type: String
- + gunLength: Int
- + bulletSpeed: Int
- + bulletDamage: Int
- + bulletSize: Int
- + owned: Boolean
- + ammo: Int
- + autoReload: Bool
- + reloadSpeed: Int
- + resetGun(): void
- + shoot(): void

ShotgunStrategy
- + type: String
- + gunLength: Int
- + bulletSpeed: Int
- + bulletDamage: Int
- + bulletSize: Int
- + owned: Boolean
- + ammo: Int
- + autoReload: Bool
- + reloadSpeed: Int
- + resetGun(): void
- + shoot(): void

TreeFactory
- - treeTypes: TreeType [ ]
- + getTreeType(name): treeType

Forest
- - factory: Factory
- + plantTree(x, y, name): Tree
- + draw(gameView)

AcaciaTreeType
- + name: String
- + color: String
- + texture: File
- + draw(gameView, x, y):void

PalmTreeType
- + name: String
- + color: String
- + texture: File
- + draw(gameView, x, y):void

TreeType<Abstract>
- + name: String
- + color: String
- + texture: File
- + draw(gameView, x, y): void

Tree
- + x: Float
- + y: Float
- + type: TreeType
- + draw(gameView): void

Ground
- + width: Float
- + height: Float
- - img: String
- + sprite: Sprite

Flyweight Pattern