COLOR SWITCH : CLASS DIAGRAM

Main

+ main(String[]): void

Game

- runGame: RunGame

RunGame

- exit_Status: boolean
- list_of_load_games: ArrayList<LoadGame>
- default_stateClass: StateClass

+ displayLoadGame() : void
+ chooseLoadGame() : void
+ resumeGame(LoadGame): void
+ startGame(StateClass) : void
+ displayGUI() : void
+ returnedBackToMainMenu() : void
+ Exit() : void
+ print() : void
+ serialize() : void
+ deserialize() : ArrayList<LoadGame>

Options_StopClass {abstract}

loadGame: LoadGame
runGame: RunGame

+ getCurrentStateOfGame(): StateClass
+ saveStateOfLoadGame(StateClass): void
+ *RunLoadGame(LoadGame):* void
+ exitToMainMenu(): void
+ getLoadGame(): LoadGame
+ getRunGame(): RunGame

«interface»
java.io.Serializable

RunningGame

- loadGame: LoadGame
- runGame: RunGame
- is_hit: boolean = false
- ball: Ball
- score: Score
- state: StateClass
- pauseGame: PauseGame

+ isHit(): boolean
+ ifHit(): void
+ getState(): StateClass
+ pauseGame(): void
+ getScore: Score

StopDueToHitClass

- score: Score

+ RunLoadGame(LoadGame): void
+ canBeContinued(): boolean
+ getLifes(): int
+ continueGame(): void

PauseGame

+ RunLoadGame(LoadGame): void
+ exitToMainMenuWithSaving(): void
+ restartGame(): void

LoadGame

arrayList_of_loadGames: ArrayList<LoadGame>=new ArrayList<LoadGame>()
- state: StateClass

+ getStateClass(): StateClass
+ saveState(StateClass): void
+ addLoadGame(LoadGame): void
+ getListOfLoadGames(): ArrayList<LoadGame>

ColorSwitch

- xPosition: float
- yPosition: float
- color: Color
- color_arraylist: ArrayList<Color>

+ addColor(Color): void
+ getRandomColor(): Color
+ getXPosition(): float
+ getYPosition(): float
+ setXPosition(float): void
+ setYPosition(float): void

Ball

- color: Color
- RADIUS: float
- VELOCITY: float
- xPosition: float
- yPosition: float

+ getColor(): Color
+ getRadius(): float
+ getVelocity(): float
+ getXPosition(): float
+ getYPosition(): float
+ setXPosition(float): void
+ setYPosition(float): void
+ setColor(Color): void
+ move(): void

StateClass

- obstacles_arrayList: ArrayList<Obstacle>
- ball: Ball
- score: Score
- starArrayList: ArrayList<Star>

+ getObstacleArraylist(): ArrayList<Obstacle>
+ setScore(Score): void
+ getBall(): Ball
+ getStarArrayList(): ArrayList<Star>
+ getScore(): Score
+ getDefaultState(): StateClass
+ getCurrentState(): StateClass
+ setObstacleArraylist(ArrayList<Obstacle>): void
+ setBall(Ball): void
+ setScore(Score): void

Score

- INITIAL_SCORE: int
- currentScore: int = 0
- finalScore: int = 0

+ getIntialScore(): int
+ getFinalScore(): int
+ getCurrentScore(): int
+ setCurrentScore(int): void
+ toString(): String
+ displayScore(): void

Star

- xPosition: float
- yPosition: float
- Size: float

+ getXPosition(): float
+ getYPosition(): float
+ getSize(): float
+ setXPosition(float): void
+ setYPosition(float): void
+ setSize(float): void

Obstacle{abstract}

XCoordinate: float
YCoordinate: float

 # Rotate():void
+ getX(): float
+ getY(): float
+ setX(float): void
+ setY(float): void

Obstacle1

Rotate(): void

Obstacle2

Rotate(): void

Obstacle3

Rotate(): void

Obstacle4

Rotate(): void

0..*
1