ObstacleComponent
`<<Java Class>>`
ObstacleComponent

-color: Paint
-node: Node

+setColor(Paint):void
+getColor()
+getNode()
+setNode(Node):void

GameState
`<<Java Class>>`
GameState

-serialVersionUID: long
-DURATION: int
-ANGLE: int
-MAX_NUMBER_OBJ: int
-RESTART_SCORE: int
-score: Integer
-starValues: ArrayList<Integer>
-isSavedGame: boolean
-translateOffset: double
-maxDisplacement: double
-trackTranslate: double
-firstObstacleY: double
-translateHalfway: double
-lastTapNs: double
-pausedNow: long
-hasEnded: boolean
-secondsTicker: double
-starCount: int
-prevStarValue: int
-starsCrossed: int
-colorChangersCrossed: int
-scoreCountNode: Text
-scene: Scene

+GameState(Gameplay)
+setGameplay(Gameplay):void
+setSaved(boolean):void
+setup():void
+initNodes(AnchorPane,Circle,Text):void
+getObstacles():ArrayList<Obstacle>
+getScore():Integer
+setScore(Integer):void
+getStarCount():int
+getPrevStarValue():int
+setStarCount(int):void
+setPrevStarValue(int):void
+pauseState():void
+playState(Boolean):void
+updateState(double):void
+restoreState():void
+addObstacle(Obstacle):void
+addStar(Star):void
+addColorChanger(ColorChanger):void
+checkForCollision():void
-collisionWithObstacle(Obstacle):void
-collisionWithStar(Star):void
-collisionWithColorChanger(ColorChanger):void
-getTransitions(ArrayList<Circle>)

GameObject
`<<Java Class>>`
GameObject

#node: Node
#WIDTH: double
#HEIGHT: double
#SCENE_WIDTH: double
#SCENE_HEIGHT: double
#INITIAL_OFFSET_Y: double

+setPosition(Point):void
+getPosition():Point
+getNode():Node
+setNode(Node):void
+isColliding(Ball):Boolean

Settings
`<<Java Class>>`
Settings

-SOUND_ON: Boolean

+Settings()
+setSoundOn(Boolean):void
+isSoundOn():Boolean

Sounds
`<<Java Class>>`
Sounds

-BOUNCE_FILE: String
-COLLISION_FILE: String
-SCORE_FILE: String

+Sounds()
-play(String):void
+bounce():void
+collision():void
+score():void

Stat
`<<Java Enumeration>>`
Stat

+GAME_COUNT: Stat
+TOTAL_SCORE: Stat
+HIGHSCORE: Stat
+REVIVAL_COUNT: Stat
+SAVED_COUNT: Stat
+TOTAL_TIME: Stat

Physics
`<<Java Class>>`
Physics

+Physics()
+velocity(double,double):double
+acceleration(double,double):double
+displacement(double,double,double):double

Constants
`<<Java Class>>`
Constants

+Constants()

Utils
`<<Java Class>>`
Utils

+Utils()
+rotate(Node,int,int)
+getAbsoluteY(Node):double
+getBounds(Node):Bounds
+intersects(Node,Node):Boolean
+intersects(Bounds,Bounds):Boolean
+isInside(Node,Node):Boolean
+isInside(Bounds,Bounds):Boolean
+loadObject(String):Node
+getComponents(Obstacle,ObservableList<Node>):ArrayList<ObstacleComponent>
+deleteNode(Node):void

Obstacle (Constants group)
`<<Java Class>>`
Obstacle

+CIRCLE: String
+GEARS: String
+SQUARE: String
+BAR: String

Scene
`<<Java Class>>`
Scene

+START_MENU: String
+PAUSE_MENU: String
+END_MENU: String
+GAMEPLAY: String
+LEADERBOARD: String
+SAVED_GAMES: String
+GAME_CARD: String
+SETTINGS: String
+HELP: String
+RESTART_DIALOG: String
+STATISTICS: String

DataFiles
`<<Java Class>>`
DataFiles

+ROOT: String
+COLOR_SWITCH: String

App
`<<Java Class>>`
App

+App()
+main(String[]):void
+deserializeColorSwitch():ColorSwitch

ColorSwitch
`<<Java Class>>`
ColorSwitch

+serialVersionUID: long
-startMenu: StartMenu

+ColorSwitch()
+getSavedGames():ArrayList<SavedGame>
+addSavedGame(SavedGame):void
+_init():void
+getStats():Statistics
+serialize():void
+start(Stage):void

SavedGamesScene
`<<Java Class>>`
SavedGamesScene

-startMenu: StartMenu
-savedGameCards: ArrayList<AnchorPane>
-lastCardY: double
-scrollPane: ScrollPane
-vbox: VBox

+display():void
+goBack():void
+addGame(SavedGame,int):void
+setScrollPane(ScrollPane):void
+SavedGamesScene(StartMenu)

StatisticsScene
`<<Java Class>>`
StatisticsScene

-startMenu: StartMenu

+StatisticsScene(StartMenu)
+display():void

SettingsScene
`<<Java Class>>`
SettingsScene

-startMenu: StartMenu
-controller: SettingsController

+display():void
+goBack():void

Gameplay
`<<Java Class>>`
Gameplay

-paused: boolean
-stage: Stage
-scene: Scene
-endMenu: EndMenu
-gpController: GameplayController
-renderLoop: AnimationTimer
-isSavedGame: Boolean

+Gameplay(SavedGame,Stage)
+Gameplay(Stage)
-commonSetup(Stage):void
+display():void
+getState():GameState
+getScene():Scene
+getStage():Stage
+getController():GameplayController
+getPaused():Boolean
+playGame(Boolean):void
+pauseGame():void
+saveGame():void
+endGame():void
+serializeGameState(SavedGame):void
+deserializeGameState(SavedGame):GameState

Obstacle
`<<Java Class>>`
Obstacle

#OBSTACLES: String[]
-FIXED_GAP: double
-lastObstacleY: double
-obstacleBaseY: double
#INITIAL_PERIOD: double
#lastTaskNs: double
#expectedNextTaskNs: double

#getDelay(double,double):long
+getShape():ObstacleShape
+getObstacleBaseY():double
+setLastObstacleY(double):void
+positionSelf(Boolean,Node):void
+move():void
+stop():void
+pause():void
+play(double):void

Track
`<<Java Class>>`
Track

-SCORE_THRESHOLD: int
-starCount: int
-prevStarValue: int

+isColliding(Ball):Boolean
+shiftUp(double):double
+addObstacle(ObstacleShape):void

Ball
`<<Java Class>>`
Ball

-JUMP_SIZE: int
-MAX_SCORE: int
-jumpDuration: double
-velocity: double
-DOWNWARD_ACCN: double

+getColor():Colors
+setColor(Colors):void
+updateVelocity(int):void
+reset():void
+move(double,double):double
+isColliding(Ball):Boolean

Star
`<<Java Class>>`
Star

-value: Integer
-YELLOW: Paint

+getValue():Integer
+setValue(int):void
+position(Point):void
+isColliding(Ball):Boolean

ColorChanger
`<<Java Class>>`
ColorChanger

+isColliding(Ball):Boolean

IScene
`<<Java Interface>>`
IScene

+display():void

SavedGame
`<<Java Class>>`
SavedGame

-serialVersionUID: long
-label: String
-gameStateFile: String
-timestamp: String
-score: String

+getLabel():String
+getScore():String
+getGameStateFile():String
+getTimestamp():String

Statistics
`<<Java Class>>`
Statistics

-gameCount: int
-totalScore: int
-highscore: int
-revivalCount: int
-savedCount: int
-totalTime: int

+Statistics()
+getStat(Stat):int
+setStat(Stat,int,boolean):void

SquareObstacle
`<<Java Class>>`
SquareObstacle

-ANGLE: int
-quarterTimer: Timer

-getTimerTask():TimerTask
+move():void
+pause():void
+play(double):void
+isColliding(Ball):Boolean
+stop():void

CircleObstacle
`<<Java Class>>`
CircleObstacle

-ANGLE: int
-transition: RotateTransition
-outerBoundingBox: Rectangle
-innerBoundingBox: Rectangle
-ring: Group
-quarterTimer: Timer
-timerCount: int
-colors: ArrayList<Paint>
-topColor: Paint
-bottomColor: Paint

-getTimerTask():TimerTask
+move():void
+pause():void
+play(double):void
+isColliding(Ball):Boolean
+stop():void

BarObstacle
`<<Java Class>>`
BarObstacle

-TRANSITION_BY_X: double
-transition: TranslateTransition

+move():void
+pause():void
+play(double):void
+isColliding(Ball):Boolean
+stop():void

GearsObstacle
`<<Java Class>>`
GearsObstacle

-ANGLE: int
-transitionLeft: RotateTransition
-transitionRight: RotateTransition
-quarterTimer: Timer
-timerCount: int
-colors: ArrayList<Paint>
-middleColor: Paint
-criticalRegion: Rectangle
-leftGear: Group
-rightGear: Group

-getMiddleColorTimerTask():TimerTask
-getNullerTimerTask():TimerTask
+move():void
+pause():void
+play(double):void
+isColliding(Ball):Boolean
+stop():void

ObstacleShape
`<<Java Enumeration>>`
ObstacleShape

+CIRCLE: ObstacleShape
+BAR: ObstacleShape
+SQUARE: ObstacleShape
+GEARS: ObstacleShape

+ObstacleShape()

Colors
`<<Java Enumeration>>`
Colors

-TURQUOISE: Colors
-MAGENTA: Colors
-ORANGE: Colors
-DARK_BLUE: Colors
+colorCode: String

-Colors(String)

Point
`<<Java Class>>`
Point

-serialVersionUID: long
-x: Double
-y: Double

+getX():double
+getY():double
+setX(double):void
+setY(double):void
+set(double,double):void

Relationship labels:
#components 0..*
-currentState 0..1
+game 0..1
-gameplay 0..1
-obstacle 0..1
-obstacles 0..*
-gameState 0..1
-gameTrack 0..1
-ball 0..1
-stars 0..*
-colorChangers 0..*
-savedGames 0..*
-stats 0..1
#shape 0..1
-obstacleShapes 0..*
-ballColor 0..1
-ballColor 0..1
#position 0..1
#OFFSET 0..1
-rotationPivot 0..1

UML Use Case Diagram

Player

File system

<<system>>

0..1

Start New Game   0..*

Display Saved Games   0..*

Help   0..*

Change Settings   0..*

View Statistics   0..*

Exit   0..1

1

Load Saved Game

<<include>>

Play Game

<<include>>   0..*

Hit Obstacle component of **diffferent** color

<<extend>>

<<extend>>   <<extend>>

<<include>>   <<extend>>

Resume Game

Pause Game

<<extend>>

<<extend>>

Save Game   0..*

Exit to Main Menu

<<extend>>

<<extend>>

End Game

<<include>>

<<extend>>

Update total points   0..*

Start New Game

Hit Star   0..*

<<include>>

Increase Points

View Statistics   0..*

Get Initial Data   1..*

1