STICK HERO UML CLASS

Player

- noOfCherries: int
- highestScore: int
- score: int
- position: int
- savedGames: Map<Long, Game>

+ tapSticks(): void
+ increaseScore(): void
+ getNoOfCherries(): int
+ setNoOfCherries(): int
+ getHighScore(): int
+ setHighScore(): int
+ getScore(): int
+ setScore(): int
+ getPosition(): int
+ saveGame(): void
+ deleteGame(); void
+ getSavedGames():Map<Long, Game>

java.lang.exception

GameNotFoundException

<<throws>>

Main

- activeGame: Game
- cherry: int

+ startGame(): void
+ exit(): void
+ openSavedGames(): void
+ getCherry(): int
+ updateCherry(int Cherry): void
+ getActiveGame(): Game
+ setActiveGame(Game g): void
+ about(): void
+ viewStatistics(): void
+ tapStickSound(): void
+ backgroundMusic(): void
+ gameOverSound(): void
+ fallingSound(): void
+ pickingCherrySound(): void

Game

- player: Player
- score: int
- charecter: Character
- cherry: ArrayList<Cherry>
- platform: list<Platform>
- gameOver(): boolean
- dateTime(): String

- generateCherry() : Cherry
- generatePlatform() : Platform
+ playGame() : void
+ exitGame(): void
+ gameOver(): void
+ Sound() : void
+ pause(): void
+ resume(); void
+ updateScore(); void
+ savedGames(): void
+ setGameOver(): boolean
+ setDateTime(String DateTimr):void
+ getDateTime(): int

Hero

- position: int

+ getPosition(); int
+ setPosition(); int
+ moveHero(); void

Stick

- length: int

+ rotateSpeed(): void
+ increaseStick(): void
+ rotate(): void
+ hasCollided(Stick s): boolean

<<interface>> Serializable

Cherry

- x: float
- y: float

+ setX(): float
+ getX(float x): void
+ setY(): float
+ getY(float y): void

Platform

- x: float
- y: float
- width: int
- height: int

+ setX(): float
+ getX(float x): void
+ setY(): float
+ getY(float y): void
+ getWidth(): int
+ setWidth():
+ getHeight(): int
+ setHeight():

<<interface>> Rotatable

+ rotate(): void

<<interface>> Collideable

+ hasCollided(Stick s): boolean

Pandillapelly Harshvardhini - 2022345
Priyanshu Yadav - 2022383