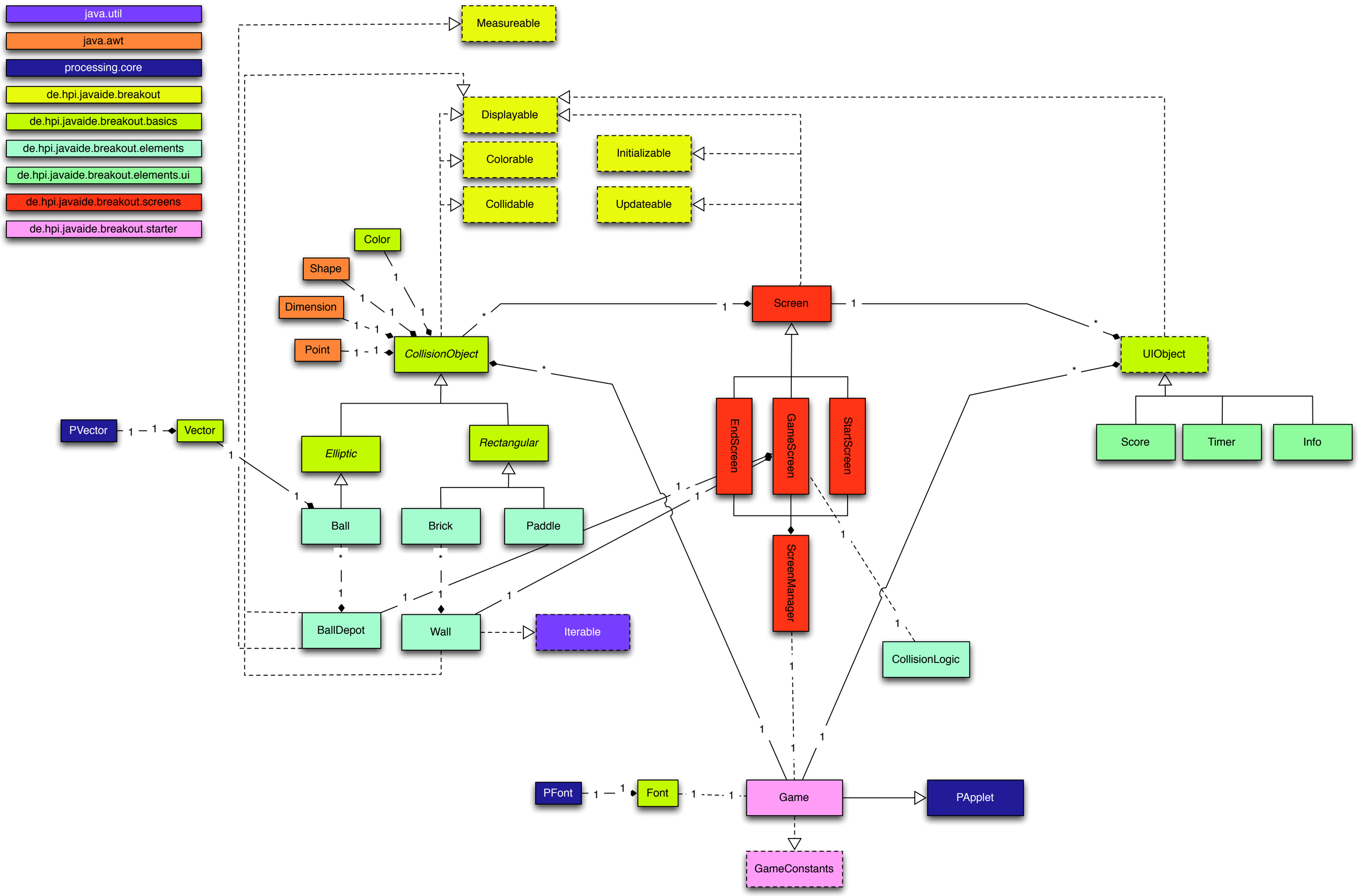

Brick

```
...
+ display(): void
+ nextStatus(): void
+ isDead(): boolean
...
```

Ball

```
...
+ display(): void
+ move(): void
...
```

Paddle

```
...
+ display(): void
+ move(): void
...
```

Wall

```
- wall: ArrayList<Brick>
...
+ removeBrick(): void
+ Iterator<Brick>(): boolean
+ display(): void
+ iterator(): Iterator<Brick>
- buildWall(): void
...
```

BallDepot

```
- balls: ArrayList<Ball>
...
+ dispense(): Ball
+ isEmpty(): boolean
+ display(): void
...
```

CollisionLogic

```
+ checkCollision(game: Game, ball: Ball, paddle: Paddle, wall: Wall): void
- ...
```

Info

```
- content: String
+ display(): void
+ update(input: String ): void
...
```

Score

```
- score: int
+ display(): void
+ update(input: String ): void
...
```

Timer

```
- time: int
- seconds: int
+ display(): void
+ update(input: String ): void
...
```

StartScreen

```
- instance: Screen
- game: Game
- infoBox: UIObject
...
+ getInstance(game: Game): Screen
+ init(): void
+ update(): void
+ display(): void
+ handleKeyPressed(key: String): void
+ handleMouseDragged(): void
+ increaseScore(i: int): void
```

GameScreen

```
- instance: Screen
- game: Game
- ballDepot: BallDepot
- currentBall: Ball
- wall: Wall
- score: Score
- timer: Timer
...
+ getInstance(game: Game): Screen
+ init(): void
+ update(): void
+ display(): void
+ handleKeyPressed(key: String): void
+ handleMouseDragged(): void
+ increaseScore(i: int): void
```

EndScreen

```
- instance: Screen
- game: Game
- infoBox: UIObject
...
+ getInstance(game: Game): Screen
+ init(): void
+ update(): void
+ display(): void
+ handleKeyPressed(key: String): void
+ handleMouseDragged(): void
+ increaseScore(i: int): void
```