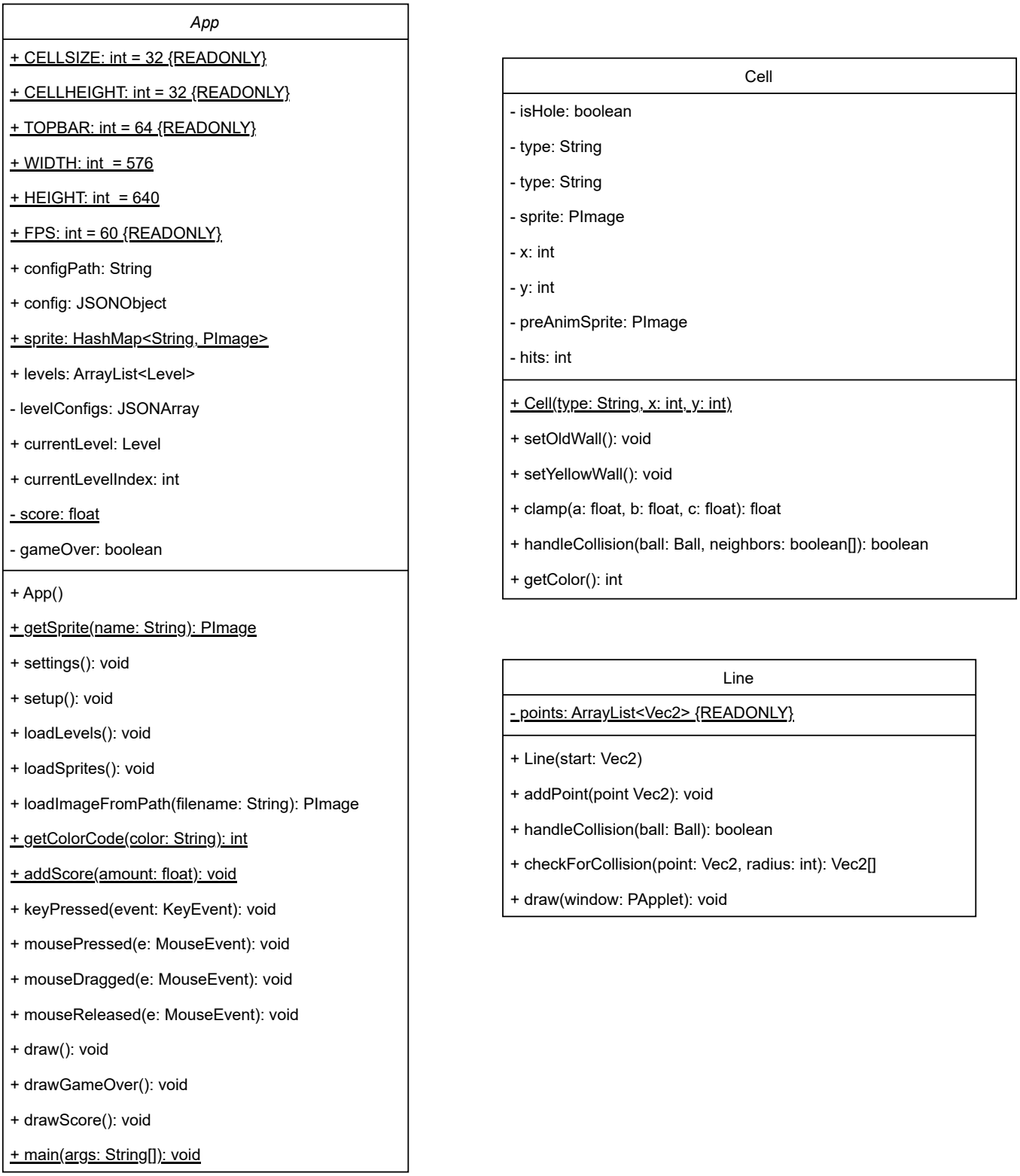
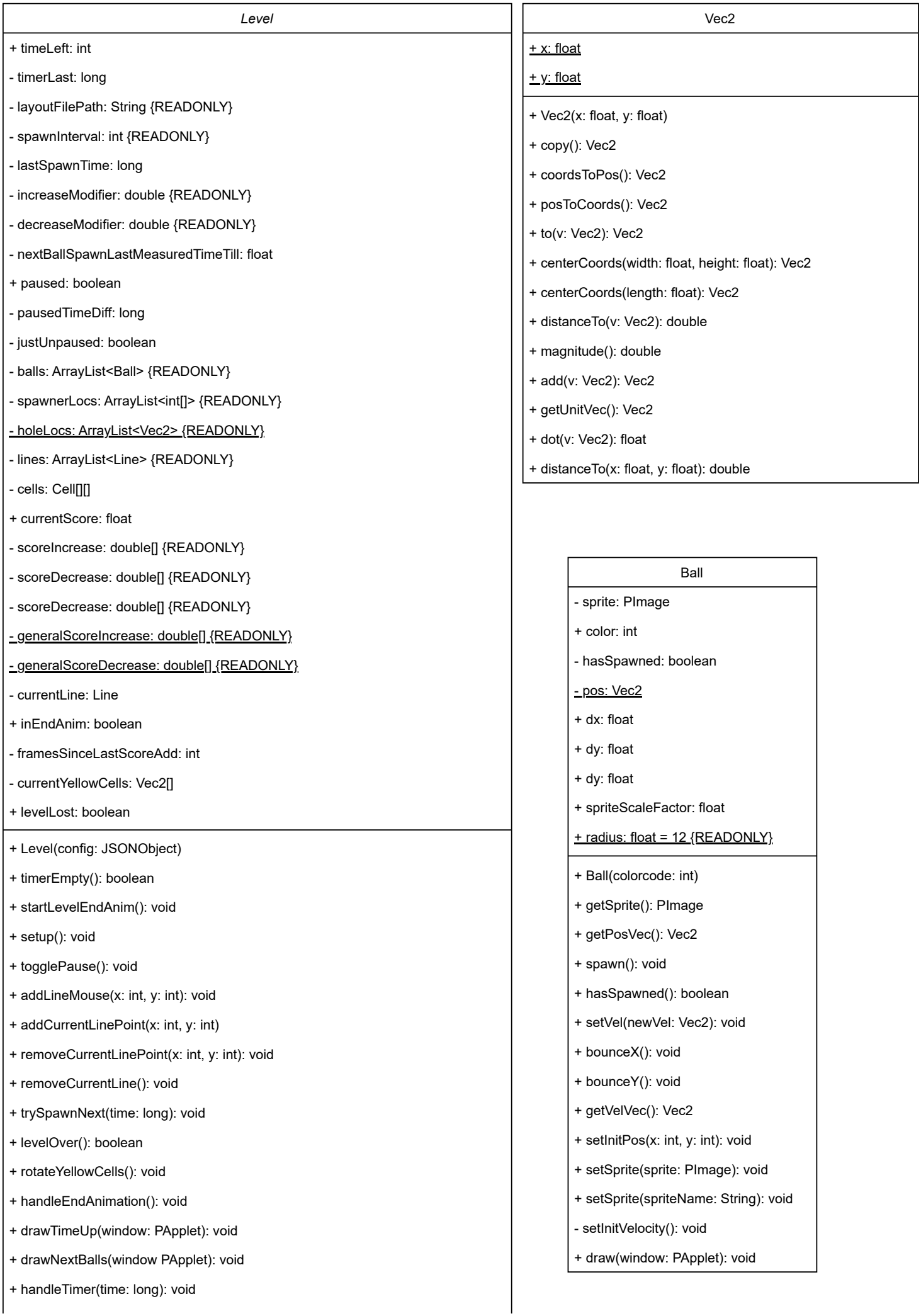

App
+ CELLSIZE: int = 32 {READONLY}
+ CELLHEIGHT: int = 32 {READONLY}
+ TOPBAR: int = 64 {READONLY}
+ WIDTH: int = 576
+ HEIGHT: int = 640
+ FPS: int = 60 {READONLY}
+ configPath: String
+ config: JSONObject
+ sprite: HashMap<String, PImage>
+ levels: ArrayList<Level>
- levelConfigs: JSONArray
+ currentLevel: Level
+ currentLevelIndex: int
- score: float
- gameOver: boolean
+ App()
+ getSprite(name: String): PImage
+ settings(): void
+ setup(): void
+ loadLevels(): void
+ loadSprites(): void
+ loadImageFromPath(filename: String): PImage
+ getColorCode(color: String): int
+ addScore(amount: float): void
+ keyPressed(event: KeyEvent): void
+ mousePressed(e: MouseEvent): void
+ mouseDragged(e: MouseEvent): void
+ mouseReleased(e: MouseEvent): void
+ draw(): void
+ drawGameOver(): void
+ drawScore(): void
+ main(args: String[]): void
Cell
- isHole: boolean
- type: String
- type: String
- sprite: PImage
- x: int
- y: int
- preAnimSprite: PImage
- hits: int
+ Cell(type: String, x: int, y: int)
+ setOldWall(): void
+ setYellowWall(): void
+ clamp(a: float, b: float, c: float): float
+ handleCollision(ball: Ball, neighbors: boolean[]): boolean
+ getColor(): int
Line
- points: ArrayList<Vec2> {READONLY}
+ Line(start: Vec2)
+ addPoint(point Vec2): void
+ handleCollision(ball: Ball): boolean
+ checkForCollision(point: Vec2, radius: int): Vec2[]
+ draw(window: PApplet): void


Level
+ timeLeft: int
- timerLast: long
- layoutFilePath: String {READONLY}
- spawnInterval: int {READONLY}
- lastSpawnTime: long
- increaseModifier: double {READONLY}
- decreaseModifier: double {READONLY}
- nextBallSpawnLastMeasuredTimeTill: float
+ paused: boolean
- pausedTimeDiff: long
- justUnpaused: boolean
- balls: ArrayList<Ball> {READONLY}
- spawnerLocs: ArrayList<int[]> {READONLY}
- holeLocs: ArrayList<Vec2> {READONLY}
- lines: ArrayList<Line> {READONLY}
- cells: Cell[][]
+ currentScore: float
- scoreIncrease: double[] {READONLY}
- scoreDecrease: double[] {READONLY}
- scoreDecrease: double[] {READONLY}
- generalScoreIncrease: double[] {READONLY}
- generalScoreDecrease: double[] {READONLY}
- currentLine: Line
+ inEndAnim: boolean
- framesSinceLastScoreAdd: int
- currentYellowCells: Vec2[]
+ levelLost: boolean
+ Level(config: JSONObject)
+ timerEmpty(): boolean
+ startLevelEndAnim(): void
+ setup(): void
+ togglePause(): void
+ addLineMouse(x: int, y: int): void
+ addCurrentLinePoint(x: int, y: int)
+ removeCurrentLinePoint(x: int, y: int): void
+ removeCurrentLine(): void
+ trySpawnNext(time: long): void
+ levelOver(): boolean
+ rotateYellowCells(): void
+ handleEndAnimation(): void
+ drawTimeUp(window: PApplet): void
+ drawNextBalls(window PApplet): void
+ handleTimer(time: long): void
Vec2
+ x: float
+ y: float
+ Vec2(x: float, y: float)
+ copy(): Vec2
+ coordsToPos(): Vec2
+ posToCoords(): Vec2
+ to(v: Vec2): Vec2
+ centerCoords(width: float, height: float): Vec2
+ centerCoords(length: float): Vec2
+ distanceTo(v: Vec2): double
+ magnitude(): double
+ add(v: Vec2): Vec2
+ getUnitVec(): Vec2
+ dot(v: Vec2): float
+ distanceTo(x: float, y: float): double
Ball
- sprite: PImage
+ color: int
- hasSpawned: boolean
- pos: Vec2
+ dx: float
+ dy: float
+ dy: float
+ spriteScaleFactor: float
+ radius: float = 12 {READONLY}
+ Ball(colorcode: int)
+ getSprite(): PImage
+ getPosVec(): Vec2
+ spawn(): void
+ hasSpawned(): boolean
+ setVel(newVel: Vec2): void
+ bounceX(): void
+ bounceY(): void
+ getVelVec(): Vec2
+ setInitPos(x: int, y: int): void
+ setSprite(sprite: PImage): void
+ setSprite(spriteName: String): void
- setInitVelocity(): void
+ draw(window: PApplet): void

+ drawText(window: PApplet): void

+ drawLines(window PApplet): void

+ handleLinesCollision(ball: Ball): void

+ drawBalls(window PApplet): void

+ addScore(adjustment: float): void

+ handleHole(ball: Ball): Vec2

+ getNeighborsArr(x: int, y: int): boolean[]

+ drawCells(window: PApplet): void

+ setCells(): void

+ setBallInit(colorCode: char, x: int, y: int): void

+ adjustScoreAmounts(): void

+ setScoreAmounts(increase: JSONObject, decrease: JSONObject): void