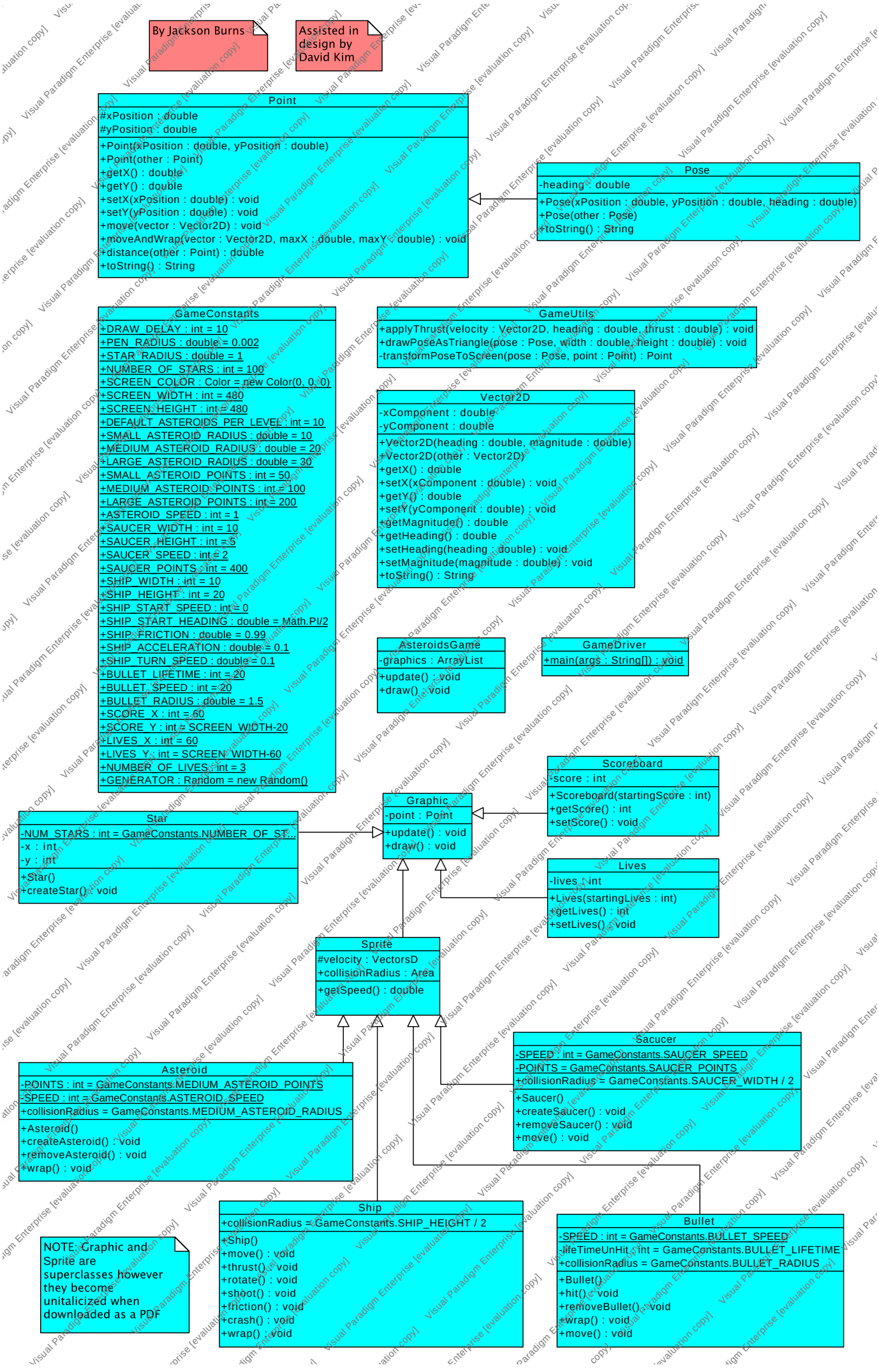
By Jackson Burns
Assisted in design by David Kim
Point
#xPosition : double
#yPosition : double
+Point(xPosition : double, yPosition : double)
+Point(other : Point)
+getX() : double
+getY() : double
+setX(xPosition : double) : void
+setY(yPosition : double) : void
+move(vector : Vector2D) : void
+moveAndWrap(vector : Vector2D, maxX : double, maxY : double) : void
+distance(other : Point) : double
+toString() : String
Pose
-heading : double
+Pose(xPosition : double, yPosition : double, heading : double)
+Pose(other : Pose)
+toString() : String
GameConstants
+DRAW_DELAY : int = 10
+PEN_RADIUS : double = 0.002
+STAR_RADIUS : double = 1
+NUMBER_OF_STARS : int = 100
+SCREEN_COLOR : Color = new Color(0, 0, 0)
+SCREEN_WIDTH : int = 480
+SCREEN_HEIGHT : int = 480
+DEFAULT_ASTEROIDS_PER_LEVEL : int = 10
+SMALL_ASTEROID_RADIUS : double = 10
+MEDIUM_ASTEROID_RADIUS : double = 20
+LARGE_ASTEROID_RADIUS : double = 30
+SMALL_ASTEROID_POINTS : int = 50
+MEDIUM_ASTEROID_POINTS : int = 100
+LARGE_ASTEROID_POINTS : int = 200
+ASTEROID_SPEED : int = 1
+SAUCER_WIDTH : int = 10
+SAUCER_HEIGHT : int = 5
+SAUCER_SPEED : int = 2
+SAUCER_POINTS : int = 400
+SHIP_WIDTH : int = 10
+SHIP_HEIGHT : int = 20
+SHIP_START_SPEED : int = 0
+SHIP_START_HEADING : double = Math.PI/2
+SHIP_FRICTION : double = 0.99
+SHIP_ACCELERATION : double = 0.1
+SHIP_TURN_SPEED : double = 0.1
+BULLET_LIFETIME : int = 20
+BULLET_SPEED : int = 20
+BULLET_RADIUS : double = 1.5
+SCORE_X : int = 60
+SCORE_Y : int = SCREEN_WIDTH-20
+LIVES_X : int = 60
+LIVES_Y : int = SCREEN_WIDTH-60
+NUMBER_OF_LIVES : int = 3
+GENERATOR : Random = new Random()
GameUtils
+applyThrust(velocity : Vector2D, heading : double, thrust : double) : void
+drawPoseAsTriangle(pose : Pose, width : double, height : double) : void
-transformPoseToScreen(pose : Pose, point : Point) : Point
Vector2D
-xComponent : double
-yComponent : double
+Vector2D(heading : double, magnitude : double)
+Vector2D(other : Vector2D)
+getX() : double
+setX(xComponent : double) : void
+getY() : double
+setY(yComponent : double) : void
+getMagnitude() : double
+getHeading() : double
+setHeading(heading : double) : void
+setMagnitude(magnitude : double) : void
+toString() : String
AsteroidsGame
-graphics : ArrayList
+update() : void
+draw() : void
GameDriver
+main(args : String[]) : void
Scoreboard
-score : int
+Scoreboard(startingScore : int)
+getScore() : int
+setScore() : void
Graphic
-point : Point
+update() : void
+draw() : void
Star
-NUM_STARS : int = GameConstants.NUMBER_OF_ST...
-x : int
-y : int
+Star()
+createStar() : void
Lives
-lives : int
+Lives(startingLives : int)
+getLives() : int
+setLives() : void
Sprite
#velocity : VectorsD
+collisionRadius : Area
+getSpeed() : double
Sacucer
-SPEED : int = GameConstants.SAUCER_SPEED
-POINTS = GameConstants.SAUCER_POINTS
+collisionRadius = GameConstants.SAUCER_WIDTH / 2
+Saucer()
+createSaucer() : void
+removeSaucer() : void
+move() : void
Asteroid
-POINTS : int = GameConstants.MEDIUM_ASTEROID_POINTS
-SPEED : int = GameConstants.ASTEROID_SPEED
+collisionRadius = GameConstants.MEDIUM_ASTEROID_RADIUS
+Asteroid()
+createAsteroid() : void
+removeAsteroid() : void
+wrap() : void
Ship
+collisionRadius = GameConstants.SHIP_HEIGHT / 2
+Ship()
+move() : void
+thrust() : void
+rotate() : void
+shoot() : void
+friction() : void
+crash() : void
+wrap() : void
Bullet
-SPEED : int = GameConstants.BULLET_SPEED
-lifeTimeUnHit : int = GameConstants.BULLET_LIFETIME
+collisionRadius = GameConstants.BULLET_RADIUS
+Bullet()
+hit() : void
+removeBullet() : void
+wrap() : void
+move() : void
NOTE: Graphic and Sprite are superclasses however they become unitalicized when downloaded as a PDF