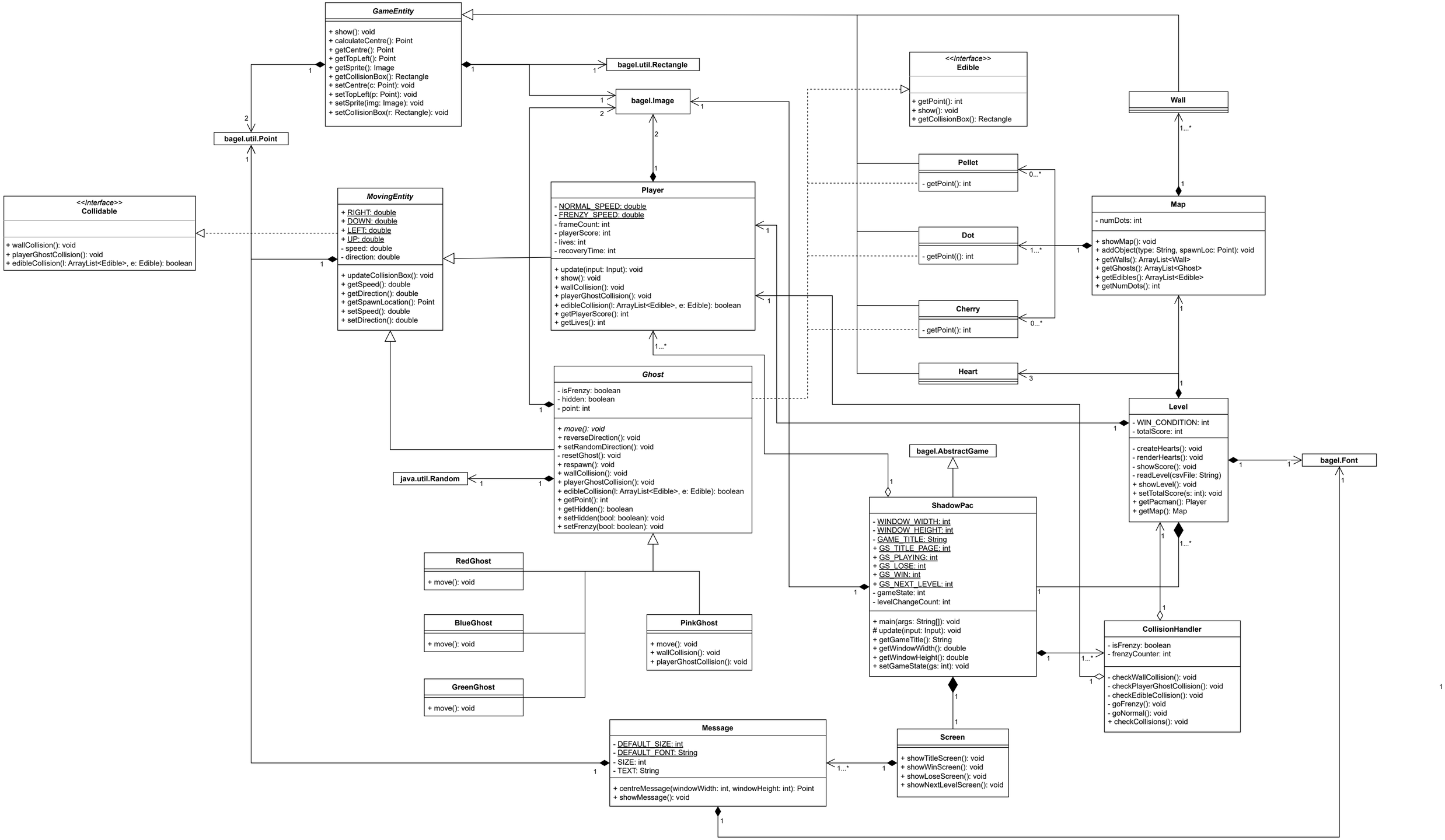
GameEntity
+ show(): void
+ calculateCentre(): Point
+ getCentre(): Point
+ getTopLeft(): Point
+ getSprite(): Image
+ getCollisionBox(): Rectangle
+ setCentre(c: Point): void
+ setTopLeft(p: Point): void
+ setSprite(img: Image): void
+ setCollisionBox(r: Rectangle): void
bagel.util.Rectangle
bagel.Image
bagel.util.Point
<<Interface>>
Edible
+ getPoint(): int
+ show(): void
+ getCollisionBox(): Rectangle
Wall
Pellet
- getPoint(): int
Dot
- getPoint(): int
Cherry
- getPoint(): int
Heart
Map
- numDots: int
+ showMap(): void
+ addObject(type: String, spawnLoc: Point): void
+ getWalls(): ArrayList<Wall>
+ getGhosts(): ArrayList<Ghost>
+ getEdibles(): ArrayList<Edible>
+ getNumDots(): int
<<Interface>>
Collidable
+ wallCollision(): void
+ playerGhostCollision(): void
+ edibleCollision(l: ArrayList<Edible>, e: Edible): boolean
MovingEntity
+ RIGHT: double
+ DOWN: double
+ LEFT: double
+ UP: double
- speed: double
- direction: double
+ updateCollisionBox(): void
+ getSpeed(): double
+ getDirection(): double
+ getSpawnLocation(): Point
+ setSpeed(): double
+ setDirection(): double
Player
- NORMAL_SPEED: double
- FRENZY_SPEED: double
- frameCount: int
- playerScore: int
- lives: int
- recoveryTime: int
+ update(input: Input): void
+ show(): void
+ wallCollision(): void
+ playerGhostCollision(): void
+ edibleCollision(l: ArrayList<Edible>, e: Edible): boolean
+ getPlayerScore(): int
+ getLives(): int
Ghost
- isFrenzy: boolean
- hidden: boolean
- point: int
+ move(): void
+ reverseDirection(): void
+ setRandomDirection(): void
- resetGhost(): void
+ respawn(): void
+ wallCollision(): void
+ playerGhostCollision(): void
+ edibleCollision(l: ArrayList<Edible>, e: Edible): boolean
+ getPoint(): int
+ getHidden(): boolean
+ setHidden(bool: boolean): void
+ setFrenzy(bool: boolean): void
java.util.Random
Level
- WIN_CONDITION: int
- totalScore: int
- createHearts(): void
- renderHearts(): void
- showScore(): void
- readLevel(csvFile: String)
+ showLevel(): void
+ setTotalScore(s: int): void
+ getPacman(): Player
+ getMap(): Map
bagel.Font
bagel.AbstractGame
ShadowPac
- WINDOW_WIDTH: int
- WINDOW_HEIGHT: int
- GAME_TITLE: String
+ GS_TITLE_PAGE: int
+ GS_PLAYING: int
+ GS_LOSE: int
+ GS_WIN: int
+ GS_NEXT_LEVEL: int
- gameState: int
- levelChangeCount: int
+ main(args: String[]): void
# update(input: Input): void
+ getGameTitle(): String
+ getWindowWidth(): double
+ getWindowHeight(): double
+ setGameState(gs: int): void
CollisionHandler
- isFrenzy: boolean
- frenzyCounter: int
- checkWallCollision(): void
- checkPlayerGhostCollision(): void
- checkEdibleCollision(): void
- goFrenzy(): void
- goNormal(): void
+ checkCollisions(): void
RedGhost
+ move(): void
BlueGhost
+ move(): void
GreenGhost
+ move(): void
PinkGhost
+ move(): void
+ wallCollision(): void
+ playerGhostCollision(): void
Message
- DEFAULT_SIZE: int
- DEFAULT_FONT: String
- SIZE: int
- TEXT: String
+ centreMessage(windowWidth: int, windowHeight: int): Point
+ showMessage(): void
Screen
+ showTitleScreen(): void
+ showWinScreen(): void
+ showLoseScreen(): void
+ showNextLevelScreen(): void