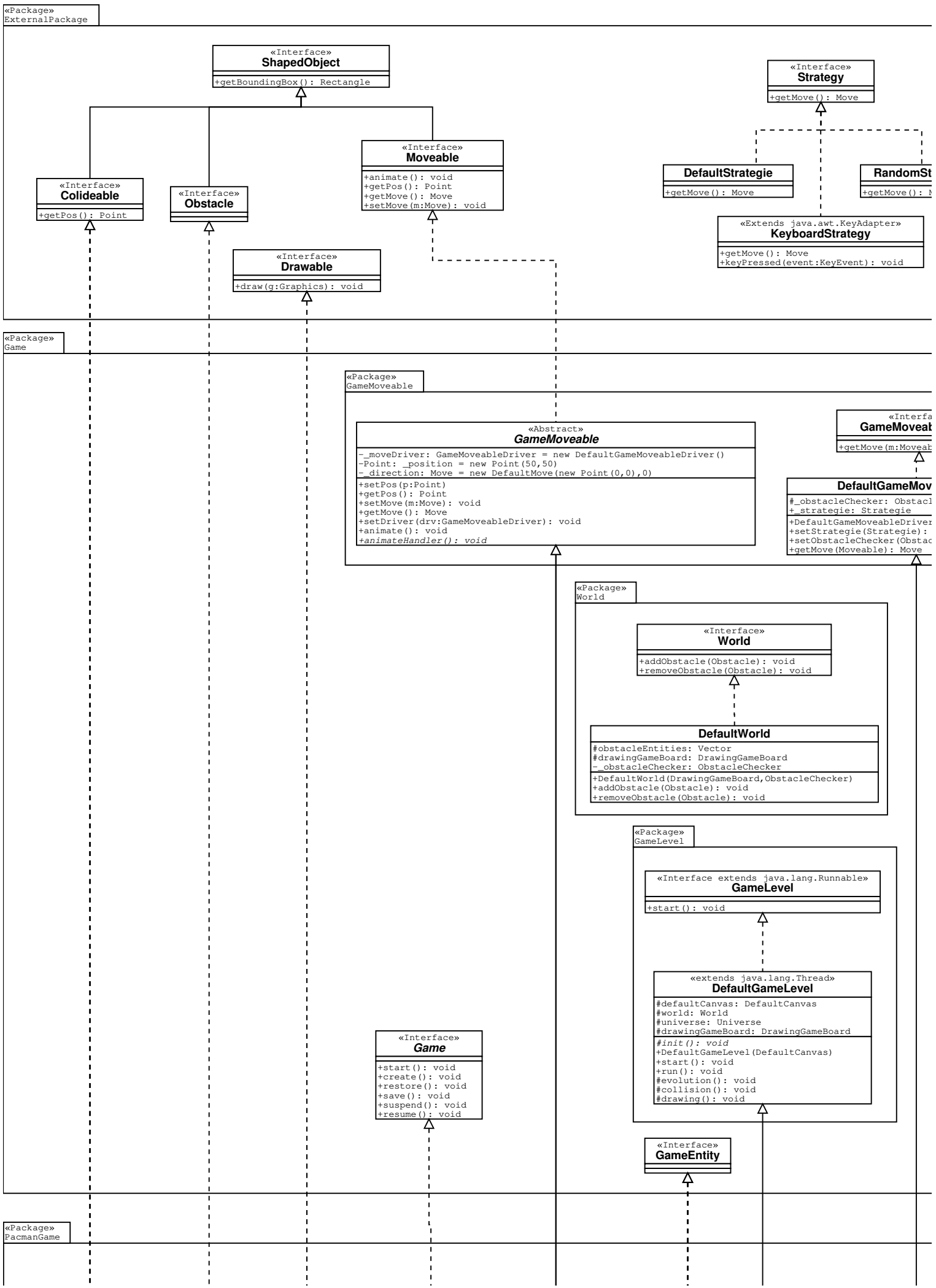
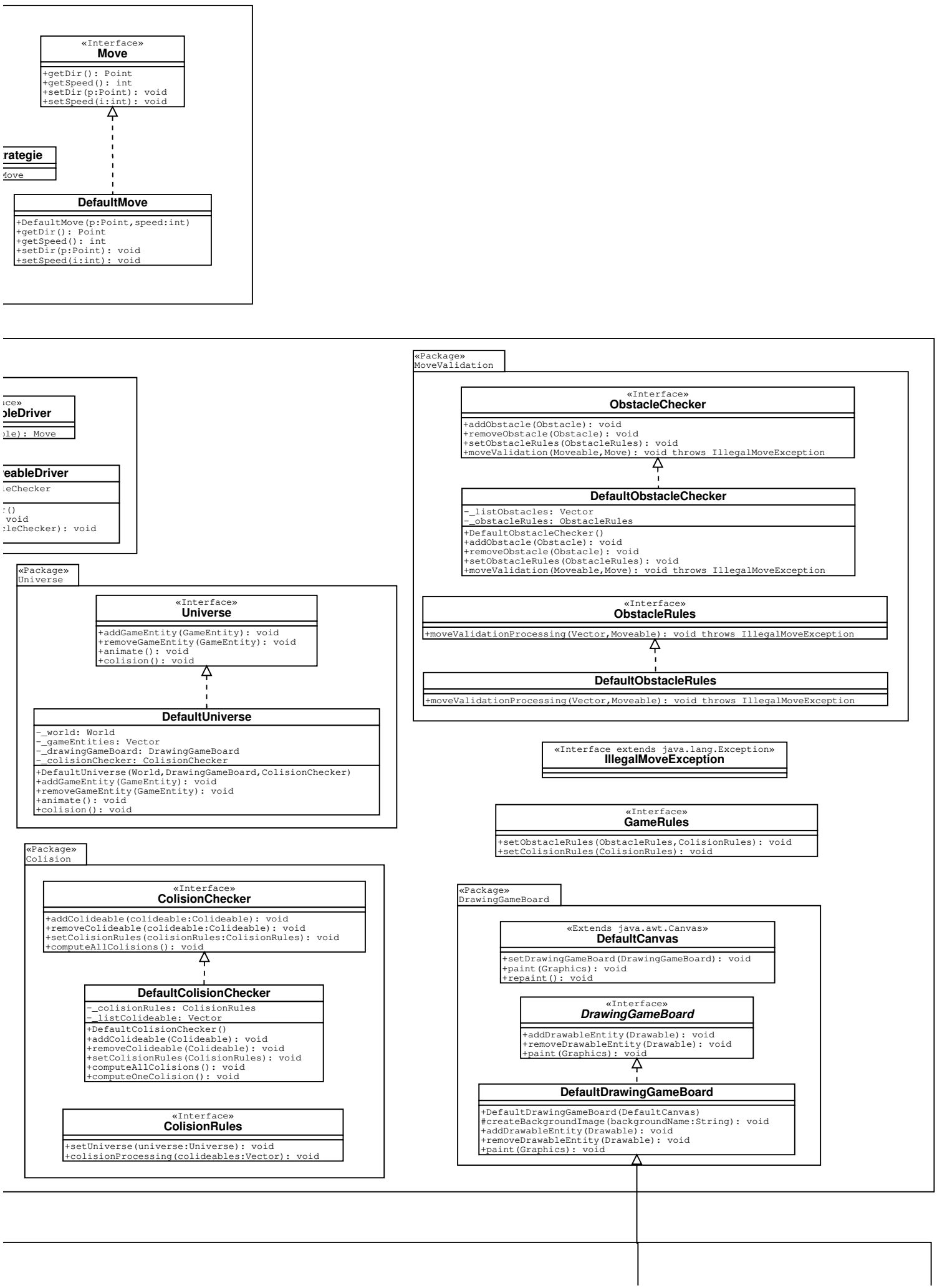
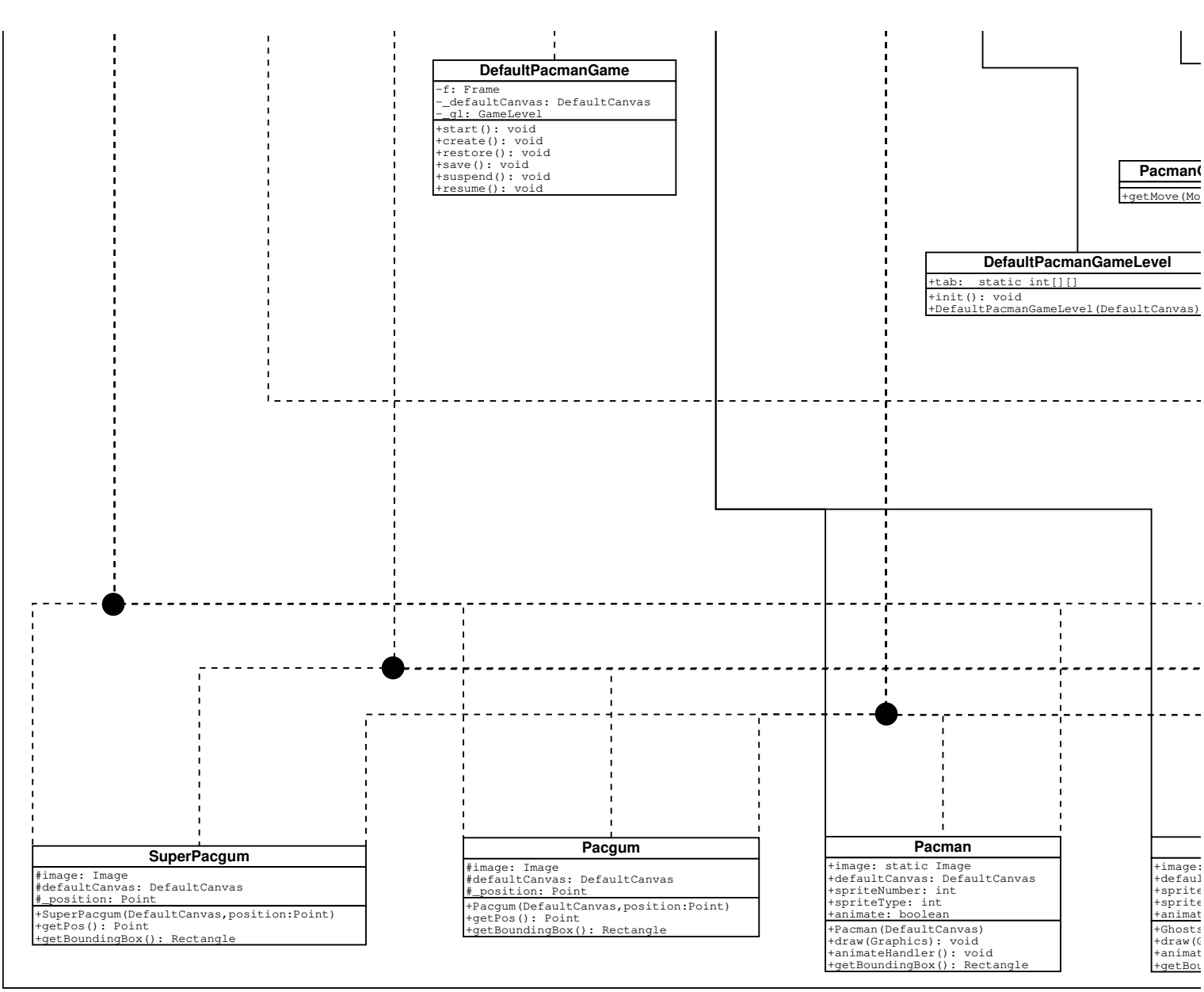
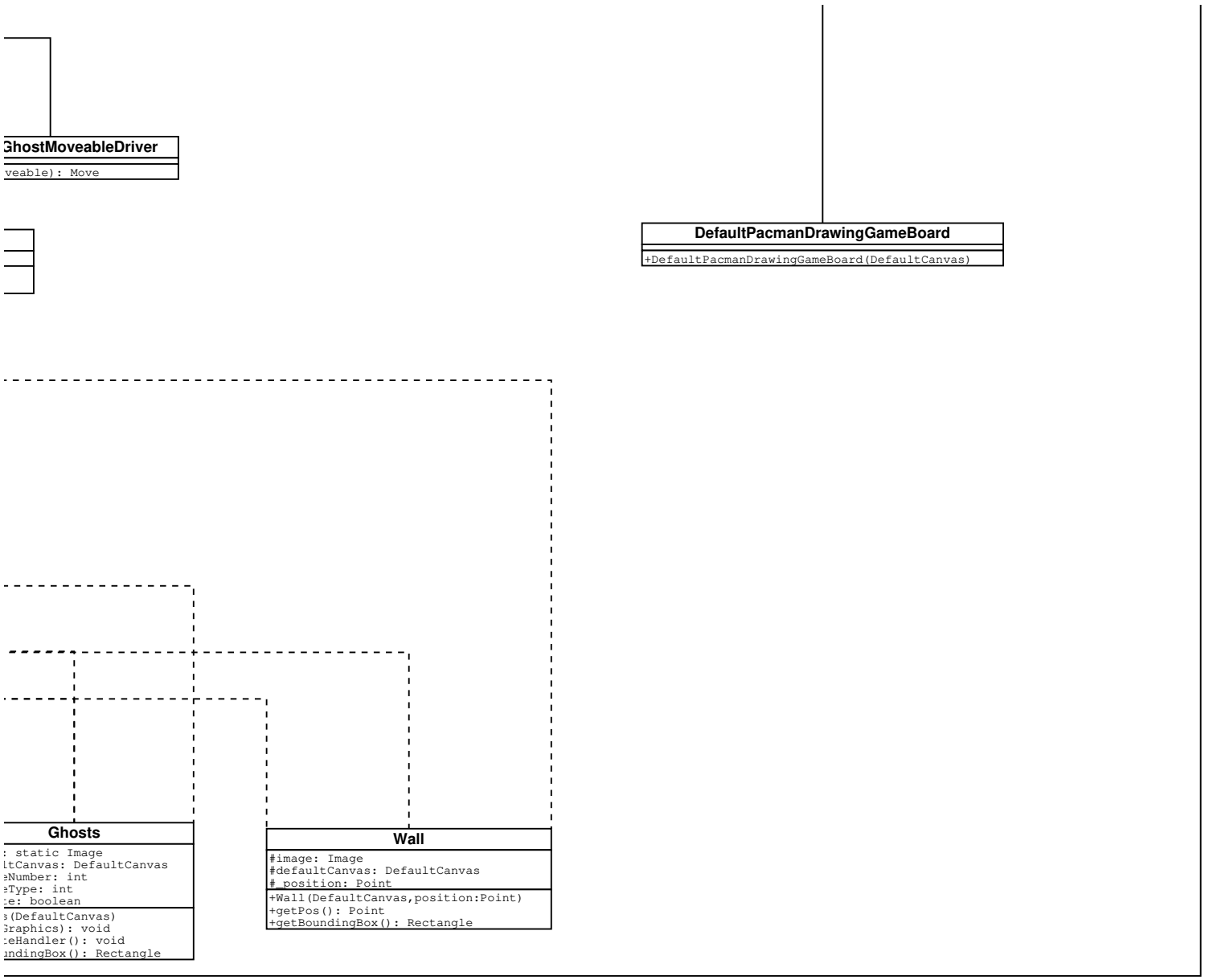

«Package»
ExternalPackage
«Interface»
ShapedObject
+getBoundingBox(): Rectangle
«Interface»
Strategy
+getMove(): Move
«Interface»
Moveable
+animate(): void
+getPos(): Point
+getMove(): Move
+setMove(m:Move): void
«Interface»
Colideable
+getPos(): Point
«Interface»
Obstacle
DefaultStrategie
+getMove(): Move
RandomSt
+getMove(): M
«Extends java.awt.KeyAdapter»
KeyboardStrategy
+getMove(): Move
+keyPressed(event:KeyEvent): void
«Interface»
Drawable
+draw(g:Graphics): void
«Package»
Game
«Package»
GameMoveable
«Abstract»
GameMoveable
-_moveDriver: GameMoveableDriver = new DefaultGameMoveableDriver()
-Point: _position = new Point(50,50)
-_direction: Move = new DefaultMove(new Point(0,0),0)
+setPos(p:Point)
+getPos(): Point
+setMove(m:Move): void
+getMove(): Move
+setDriver(drv:GameMoveableDriver): void
+animate(): void
+animateHandler(): void
«Interfa
GameMoveab
+getMove(m:Moveab
DefaultGameMov
#_obstacleChecker: Obstacl
+_strategie: Strategie
+DefaultGameMoveableDriver
+setStrategie(Strategie):
+setObstacleChecker(Obstac
+getMove(Moveable): Move
«Package»
World
«Interface»
World
+addObstacle(Obstacle): void
+removeObstacle(Obstacle): void
DefaultWorld
#obstacleEntities: Vector
#drawingGameBoard: DrawingGameBoard
-_obstacleChecker: ObstacleChecker
+DefaultWorld(DrawingGameBoard,ObstacleChecker)
+addObstacle(Obstacle): void
+removeObstacle(Obstacle): void
«Package»
GameLevel
«Interface extends java.lang.Runnable»
GameLevel
+start(): void
«extends java.lang.Thread»
DefaultGameLevel
#defaultCanvas: DefaultCanvas
#world: World
#universe: Universe
#drawingGameBoard: DrawingGameBoard
#init(): void
+DefaultGameLevel(DefaultCanvas)
+start(): void
+run(): void
#evolution(): void
#collision(): void
#drawing(): void
«Interface»
Game
+start(): void
+create(): void
+restore(): void
+save(): void
+suspend(): void
+resume(): void
«Interface»
GameEntity
«Package»
PacmanGame


«Interface»
Move
+getDir(): Point
+getSpeed(): int
+setDir(p:Point): void
+setSpeed(i:int): void
DefaultMove
+DefaultMove(p:Point,speed:int)
+getDir(): Point
+getSpeed(): int
+setDir(p:Point): void
+setSpeed(i:int): void
«Package»
MoveValidation
«Interface»
ObstacleChecker
+addObstacle(Obstacle): void
+removeObstacle(Obstacle): void
+setObstacleRules(ObstacleRules): void
+moveValidation(Moveable,Move): void throws IllegalMoveException
DefaultObstacleChecker
-_listObstacles: Vector
-_obstacleRules: ObstacleRules
+DefaultObstacleChecker()
+addObstacle(Obstacle): void
+removeObstacle(Obstacle): void
+setObstacleRules(ObstacleRules): void
+moveValidation(Moveable,Move): void throws IllegalMoveException
«Interface»
ObstacleRules
+moveValidationProcessing(Vector,Moveable): void throws IllegalMoveException
DefaultObstacleRules
+moveValidationProcessing(Vector,Moveable): void throws IllegalMoveException
«Package»
Universe
«Interface»
Universe
+addGameEntity(GameEntity): void
+removeGameEntity(GameEntity): void
+animate(): void
+colision(): void
DefaultUniverse
-_world: World
-_gameEntities: Vector
-_drawingGameBoard: DrawingGameBoard
-_colisionChecker: ColisionChecker
+DefaultUniverse(World,DrawingGameBoard,ColisionChecker)
+addGameEntity(GameEntity): void
+removeGameEntity(GameEntity): void
+animate(): void
+colision(): void
«Interface extends java.lang.Exception»
IllegalMoveException
«Interface»
GameRules
+setObstacleRules(ObstacleRules,ColisionRules): void
+setColisionRules(ColisionRules): void
«Package»
Colision
«Interface»
ColisionChecker
+addColideable(colideable:Colideable): void
+removeColideable(colideable:Colideable): void
+setColisionRules(colisionRules:ColisionRules): void
+computeAllColisions(): void
DefaultColisionChecker
-_colisionRules: ColisionRules
-_listColideable: Vector
+DefaultColisionChecker()
+addColideable(Colideable): void
+removeColideable(Colideable): void
+setColisionRules(ColisionRules): void
+computeAllColisions(): void
+computeOneColision(): void
«Interface»
ColisionRules
+setUniverse(universe:Universe): void
+colisionProcessing(colideables:Vector): void
«Package»
DrawingGameBoard
«Extends java.awt.Canvas»
DefaultCanvas
+setDrawingGameBoard(DrawingGameBoard): void
+paint(Graphics): void
+repaint(): void
«Interface»
DrawingGameBoard
+addDrawableEntity(Drawable): void
+removeDrawableEntity(Drawable): void
+paint(Graphics): void
DefaultDrawingGameBoard
+DefaultDrawingGameBoard(DefaultCanvas)
#createBackgroundImage(backgroundName:String): void
+addDrawableEntity(Drawable): void
+removeDrawableEntity(Drawable): void
+paint(Graphics): void


DefaultPacmanGame
-f: Frame
-_defaultCanvas: DefaultCanvas
-_gl: GameLevel
+start(): void
+create(): void
+restore(): void
+save(): void
+suspend(): void
+resume(): void
Pacman(
+getMove(Mo
DefaultPacmanGameLevel
+tab: static int[][]
+init(): void
+DefaultPacmanGameLevel(DefaultCanvas)
SuperPacgum
#image: Image
#defaultCanvas: DefaultCanvas
#_position: Point
+SuperPacgum(DefaultCanvas,position:Point)
+getPos(): Point
+getBoundingBox(): Rectangle
Pacgum
#image: Image
#defaultCanvas: DefaultCanvas
#_position: Point
+Pacgum(DefaultCanvas,position:Point)
+getPos(): Point
+getBoundingBox(): Rectangle
Pacman
+image: static Image
+defaultCanvas: DefaultCanvas
+spriteNumber: int
+spriteType: int
+animate: boolean
+Pacman(DefaultCanvas)
+draw(Graphics): void
+animateHandler(): void
+getBoundingBox(): Rectangle


GhostMoveableDriver
veable): Move
DefaultPacmanDrawingGameBoard
+DefaultPacmanDrawingGameBoard(DefaultCanvas)
Ghosts
: static Image
ltCanvas: DefaultCanvas
eNumber: int
eType: int
ze: boolean
s(DefaultCanvas)
Graphics): void
zeHandler(): void
undingBox(): Rectangle
Wall
#image: Image
#defaultCanvas: DefaultCanvas
#_position: Point
+Wall(DefaultCanvas,position:Point)
+getPos(): Point
+getBoundingBox(): Rectangle