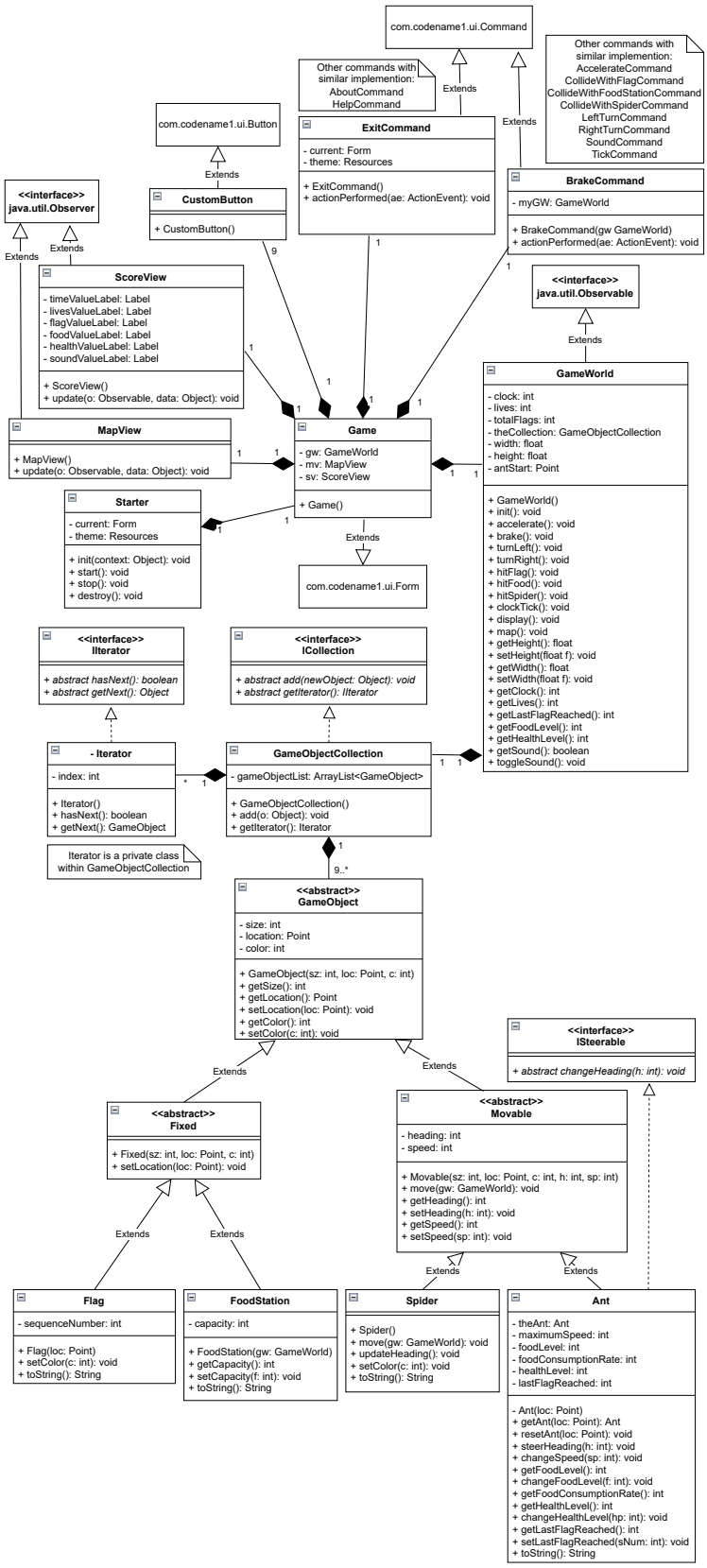
com.codename1.ui.Command
Other commands with similar implemention:
AboutCommand
HelpCommand
Extends
Extends
Other commands with similar implemention:
AccelerateCommand
CollideWithFlagCommand
CollideWithFoodStationCommand
CollideWithSpiderCommand
LeftTurnCommand
RightTurnCommand
SoundCommand
TickCommand
com.codename1.ui.Button
Extends
ExitCommand
- current: Form
- theme: Resources
+ ExitCommand()
+ actionPerformed(ae: ActionEvent): void
BrakeCommand
- myGW: GameWorld
+ BrakeCommand(gw GameWorld)
+ actionPerformed(ae: ActionEvent): void
<<interface>>
java.util.Observer
Extends
Extends
CustomButton
+ CustomButton()
9
1
1
ScoreView
- timeValueLabel: Label
- livesValueLabel: Label
- flagValueLabel: Label
- foodValueLabel: Label
- healthValueLabel: Label
- soundValueLabel: Label
+ ScoreView()
+ update(o: Observable, data: Object): void
<<interface>>
java.util.Observable
Extends
GameWorld
- clock: int
- lives: int
- totalFlags: int
- theCollection: GameObjectCollection
- width: float
- height: float
- antStart: Point
+ GameWorld()
+ init(): void
+ accelerate(): void
+ brake(): void
+ turnLeft(): void
+ turnRight(): void
+ hitFlag(): void
+ hitFood(): void
+ hitSpider(): void
+ clockTick(): void
+ display(): void
+ map(): void
+ getHeight(): float
+ setHeight(float f): void
+ getWidth(): float
+ setWidth(float f): void
+ getClock(): int
+ getLives(): int
+ getLastFlagReached(): int
+ getFoodLevel(): int
+ getHealthLevel(): int
+ getSound(): boolean
+ toggleSound(): void
MapView
+ MapView()
+ update(o: Observable, data: Object): void
Game
- gw: GameWorld
- mv: MapView
- sv: ScoreView
+ Game()
Starter
- current: Form
- theme: Resources
+ init(context: Object): void
+ start(): void
+ stop(): void
+ destroy(): void
Extends
com.codename1.ui.Form
<<interface>>
IIterator
+ abstract hasNext(): boolean
+ abstract getNext(): Object
<<interface>>
ICollection
+ abstract add(newObject: Object): void
+ abstract getIterator(): IIterator
- Iterator
- index: int
+ Iterator()
+ hasNext(): boolean
+ getNext(): GameObject
GameObjectCollection
- gameObjectList: ArrayList<GameObject>
+ GameObjectCollection()
+ add(o: Object): void
+ getIterator(): Iterator
Iterator is a private class within GameObjectCollection
9..*
<<abstract>>
GameObject
- size: int
- location: Point
- color: int
+ GameObject(sz: int, loc: Point, c: int)
+ getSize(): int
+ getLocation(): Point
+ setLocation(loc: Point): void
+ getColor(): int
+ setColor(c: int): void
<<interface>>
ISteerable
+ abstract changeHeading(h: int): void
Extends
Extends
<<abstract>>
Fixed
+ Fixed(sz: int, loc: Point, c: int)
+ setLocation(loc: Point): void
<<abstract>>
Movable
- heading: int
- speed: int
+ Movable(sz: int, loc: Point, c: int, h: int, sp: int)
+ move(gw: GameWorld): void
+ getHeading(): int
+ setHeading(h: int): void
+ getSpeed(): int
+ setSpeed(sp: int): void
Extends
Extends
Extends
Extends
Flag
- sequenceNumber: int
+ Flag(loc: Point)
+ setColor(c: int): void
+ toString(): String
FoodStation
- capacity: int
+ FoodStation(gw: GameWorld)
+ getCapacity(): int
+ setCapacity(f: int): void
+ toString(): String
Spider
+ Spider()
+ move(gw: GameWorld): void
+ updateHeading(): void
+ setColor(c: int): void
+ toString(): String
Ant
- theAnt: Ant
- maximumSpeed: int
- foodLevel: int
- foodConsumptionRate: int
- healthLevel: int
- lastFlagReached: int
- Ant(loc: Point)
+ getAnt(loc: Point): Ant
+ resetAnt(loc: Point): void
+ steerHeading(h: int): void
+ changeSpeed(sp: int): void
+ getFoodLevel(): int
+ changeFoodLevel(f: int): void
+ getFoodConsumptionRate(): int
+ getHealthLevel(): int
+ changeHealthLevel(hp: int): void
+ getLastFlagReached(): int
+ setLastFlagReached(sNum: int): void
+ toString(): String