Name:	Class Model
Author:	macavaneys
Version:	1.0
Created:	12/11/2013 9:13:50 AM
Updated:	12/16/2013 9:08:43 AM

WavPlayer

- `-` BUFFER_SIZE :int = 64000 {readOnly}
- `-` filename :String

- `+` play() :void
- `+` WavPlayer(String)

JComponent

GameCharacterImage

- `-` image :BufferedImage
- `+` IMAGESIZE :int = Game.BOARDSIZE/… {readOnly}
- `-` serialVersionUID :long = 1L {readOnly}

- `+` GameCharacterImage(String)
- `-` loadImage(String) :void
- `+` paintComponent(Graphics) :void
- `-` transformImage(BufferedImage) :Image

+player

~image

JFrame

GameBoard

- `-` board :GameBoard = null
- `~` characters :List<GameCharacter> = new ArrayList<G… {readOnly}
- `~` grid :GameGrid
- `-` serialVersionUID :long = 1L {readOnly}
- `-` timeRemainingLabel :JLabel

- `~` addCharacter(GameCharacter, int, int) :void
- `~` createBoard(String) :void
- `-` createUI(String) :void
- `~` displayTime(int) :void
- `-` GameBoard(String)
- `~` removeCharacter(GameCharacter) :void

~grid

«static»
GameGrid

- `-` cells :List<GameCharacter> (▯▯)

- `~` addAt(GameCharacter, int, int) :boolean
- `-` GameGrid(int)
- `~` getCharactersAt(int, int) :List<GameCharacter>
- `~` remove(GameCharacter) :boolean
- `~` removeAt(GameCharacter, int, int) :boolean

~characters

0..*

-cells

0..*

GameCharacter

- `~` currentPos :Point
- `~` image :GameCharacterImage
- `~` moveValid :boolean
- `#` name :String
- `-` player :WavPlayer
- `~` r :Runnable = new Runnable() …
- `~` score :int
- `-` serialNumber :int
- `-` serialNumberSequence :int = 1
- `-` soundThread :Thread
- `~` targetPos :Point

«encapsulated behavior»
- `~` moveBehavior :MoveBehavior

- `+` distanceTo(GameCharacter) :double
- `+` *finishMove(List<GameCharacter>) :void*
- `+` GameCharacter(String, String, String)
- `+` getSerialNumber() :int
- `+` makeSound() :void
- `+` planMove(List<GameCharacter>) :void

«property get»
- `+` getCurrentColumn() :int
- `+` getCurrentRow() :int
- `#` getImage() :GameCharacterImage
- `+` getName() :String
- `+` getScore() :int
- `#` getTargetColumn() :int
- `#` getTargetRow() :int

«property set»
- `#` setCurrentPosition(int, int) :void
- `+` setMoveBehavior(MoveBehavior.behaviorType) :void
- `#` setScore(int) :void
- `#` setTargetPosition(int, int) :void

«enumeration»
behaviorType

NONE
RANDOM
SMART
GREEDY
SUPER_SMART

~moveBehavior

«strategy interface»
MoveBehavior

- `+` planMove(List<GameCharacter>) :void

-gc

-gc

-gc

«concrete strategy»
StationaryMoveBehavior

- `-` gc :GameCharacter

- `+` planMove(List<GameCharacter>) :void
- `+` StationaryMoveBehavior(GameCharacter)

«concrete strategy»
RandomMoveBehavior

- `-` gc :GameCharacter
- `-` RANDOM_INCREMENT :int = 3 {readOnly}

- `+` planMove(List<GameCharacter>) :void
- `+` RandomMoveBehavior(GameCharacter)

«concrete strategy»
GreedyMoveBehavior

- `-` gc :GameCharacter

- `+` GreedyMoveBehavior(GameCharacter)
- `+` planMove(List<GameCharacter>) :void

S…

- `-` gc :GameC…
- `-` MINIMUM_T…

- `+` planMove(Lis…
- `+` SmartMoveB…

«strategy interface»
TagBehavior

+ tag(GameCharacter, GameCharacter) :int

-tagBehavior / TagBehavior

«concrete strategy»
StandardTagBehavior

+ tag(GameCharacter, GameCharacter) :int

«concrete strategy»
AdvancedTagBehavior

~ GREEDY_FLOWER_NAME :String = "rose" {readOnly}
- random :Random
- RANDOM_BEHAVIOR_CHANGE_PROBABILITY :float = 0.5f {readOnly}
~ RANDOM_FLOWER_NAME :String = "nightshade" {readOnly}
~ SMART_FLOWER_NAME :String = "daisy" {readOnly}
- STEAL_POINTS_PROBABILITY :float = 0.2f {readOnly}
- STEAL_POINTS_RATE :float = 0.5f {readOnly}

«encapsulated behavior»
- standardTagBehavior :TagBehavior

+ AdvancedTagBehavior()
- stealPoints(GameCharacter, GameCharacter) :int
+ tag(GameCharacter, GameCharacter) :int
- tagGreedyFlower(GameCharacter, GameCharacter) :void
- tagRandomFlower(GameCharacter, GameCharacter) :boolean
- tagSmartFlower(GameCharacter, GameCharacter) :void

Game

~ ADVANCED_GAME_MODE_TIME :int = (int)(GAME_TIME… {readOnly}
~ BOARDSIZE :int = 500 {readOnly}
~ GAME_TIME :int = 100 {readOnly}
~ GRIDSIZE :int = 20 {readOnly}
~ INITIAL_POINTS :int = 25 {readOnly}
~ MAX_MOVE_DISTANCE :int = 3 {readOnly}
~ timeCounter :int
~ TIMER_PAUSE_TIME :int = 2 {readOnly}
~ TURN_POINTS :int = 1 {readOnly}

«encapsulated behavior»
- tagBehavior :TagBehavior

- createBees() :void
- createFlowers() :void
- executeMove(GameCharacter, double) :void
+ getCharacters() :List<GameCharacter>
+ main(String[]) :void
- runGame() :void
+ tag(GameCharacter, GameCharacter) :int
+ validateMove(GameCharacter) :void

Flower

+ finishMove(List<GameCharacter>) :void
- Flower(String, String)

«factory method»
+ createFlower(String, String) :Flower

Bee

- Bee(String, String, String)
+ finishMove(List<GameCharacter>) :void

«factory method»
+ createBee(String, String, String, MoveBehavior.behaviorType) :Bee

---

-gc

-gc

«concrete strategy»
SmartMoveBehavior

haracter
TARGET_FLOWER_POINTS :int = 3

st<GameCharacter>) :void
Behavior(GameCharacter)

«concrete strategy»
SuperSmartMoveBehavior

+ DANGEROUS_FLOWER_NAMES :String ([]) = new String[] {A… {readOnly}
- gc :GameCharacter

- isDangerous(GameCharacter) :boolean
+ planMove(List<GameCharacter>) :void
+ SuperSmartMoveBehavior(GameCharacter)