NHF – Pac-Man

A programozás alapjai 3

Ábel András Attila

QUMFKD

Tartalom

Rövid szöveges ismertetés

A program az ikonikus Pac-Man videójátékot valósítja meg, amiben a játékos egy sárga karaktert irányít egy labirintusban úgy, hogy közben elkerüli az őt üldöző négy szellemet.

A játékos feladata a pályán megjelenő pontok elfogyasztása. A pontok mellett Pac-Man a pályán véletlenszerűen megjelenő nagyobb golyókat is el tudja fogyasztani, ami ideiglenesen lehetővé teszi neki, hogy megegye az őt üldöző szellemeket is. Ekkor a megevett szellem visszamegy a „házba”. A játék akkor ér véget, ha Pac-Mant elkapja valamelyik őt üldöző szellem, vagy megeszi az összes pontot a pályán

Pac-Man folyamatosan mozog a labirintuson belül amíg falba nem ütközik, de erre csak négy irányban képes: felfelé, lefelé, balra és jobbra. Ezt a játékos a nyilakkal irányítani.

A játék kezdetekor két szellem kezdi el üldözni a játékost, a másik kettő a „házban” van és egyesével hagyják el azt az idő múlásával. A négy szellem mind más stratégiát alkalmaz a játékos elkapására:

- Blinky a legagresszívebb szellem. Mindig követi Pac-Man aktuális pozícióját.
- Pinky ahelyett, hogy követné Pac-Mant inkább próbál elékerülni.
- Inky viszonylag kiszámíthatatlan kiszámíthatatlanul mozog: megpróbálja Blinkyt követni és amikor elér egy bizonyos távolságot tőle, megpróbál a másik oldalra kerülni, ami miatt néha váratlanul cselekszik.
- Clyde bizonyos távolságig megközelíti Pac-Man-t, de ha túl közel kerül hozzá, akkor elmenekül.

Use-case-ek

A program indításakor a főmenü jelenik meg, itt a játékos két lehetőség közül választhat.

Ha a szintkészítőt lehetőséget választja, akkor egy 28x31-es méretű „mátrix” fog megjelenni előtte tele pontokkal. A felhasználó ezután a menü panelen megjelenő legördülő menükből választhatja ki, hogy éppen mit szeretne a pályára helyezni. Az elkészült, vagy esetleg félkész pályát a felhasználó elmentheti, vagy be is tölthet egy másikat. Mentésre csak akkor van lehetőség, ha a pálya játszható állapotban van. A pálya csak akkor játszható, hogyha Pac-Man és mind a 4 szellem is pályára került.

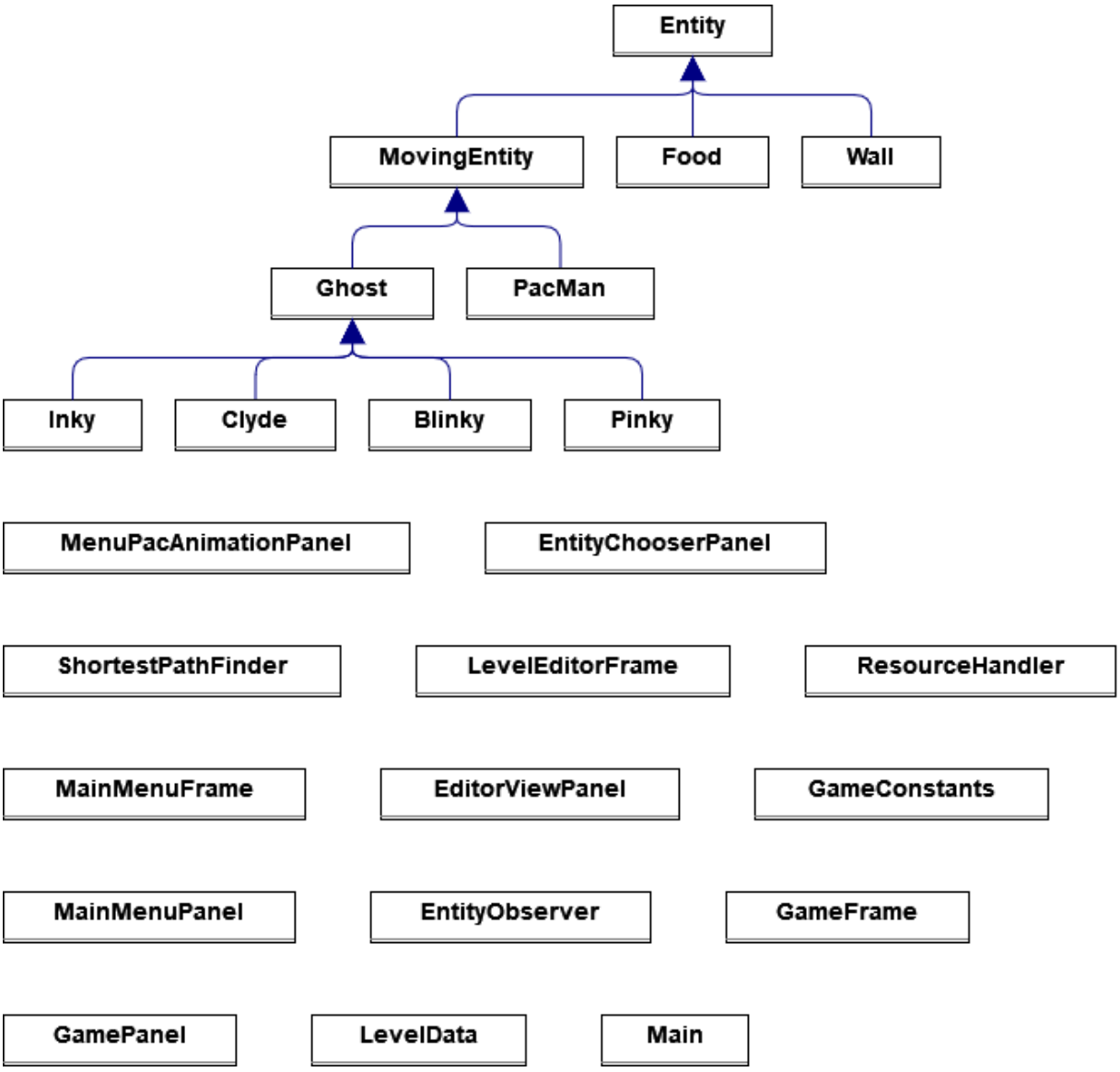
Ha a játék lehetőségét választja akkor ki kell választania egy tetszőleges szintet a gépről, amit a program betölt. A játék akkor indul, ha a felhasználó valamelyik nyíl segítségével elindítja Pac-Mant. A játékmenet az előzőleg leírtak szerint történik.

Megoldási ötlet

A programot a Java programozási nyelv segítségével fogom megvalósítani. A grafikus kezelőfelületet a Swing könyvtárral, a fájlok kezelését az ObjectOutputStream osztállyal, a teszteket pedig a JUnit 4 használatával fogom megvalósítani.

Programozói dokumentáció

Osztálydiagram

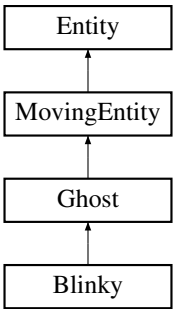


Class Documentation

3.1 Blinky Class Reference

This class represents Blinky.

Inheritance diagram for Blinky:



Public Member Functions

- Blinky (int x, int y)

  Constructor for Blinky.
- BufferedImage getSprite ()

  Gets the sprite for Blinky.
- void **update** ()

  Moves Blinky.

Public Member Functions inherited from Ghost

- boolean isEdible ()

  Checks if the ghost is edible.
- void eatenBy (PacMan pacMan)

  Tries to eat the ghost, fails if the ghost is not frightened.
- void update (Point newLocation, String newDirection, boolean canEatGhosts)

  Updates the ghost's location, direction, and frightened state.
- boolean isDead ()

  Checks if the ghost is dead.

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Additional Inherited Members

Protected Member Functions inherited from Ghost

- Ghost (int x, int y)

  Constructor for the ghost.

Protected Member Functions inherited from MovingEntity

- MovingEntity (int x, int y)

  Constructor for the moving entity.
- void **getNextAnimation** ()

  Sets the sprite to the next one in the animation.

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes inherited from Ghost

- boolean **isFrightened**

  True if the ghost is frightened, false otherwise.
- boolean **isDead**

  True if the ghost is dead, false otherwise.
- Point **pacManLocation**

  The current location of PacMan.
- String **pacManDirection**

  The current direction of PacMan.
- int **deathTimer**

  The number of frames the ghost is dead for.

Protected Attributes inherited from MovingEntity

- int **speedX**

  Stores the speed on the x axis of the entity.
- int **speedY**

  Stores the speed on the y axis of the entity.
- String **currentDirection** = NEUTRAL

  Stores the current direction of the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

Static Protected Attributes inherited from MovingEntity

- static final String **NEUTRAL** = "neutral"

  Sprite constant.
- static final String **LEFT_1** = "left_1"

  Sprite constant.
- static final String **LEFT_2** = "left_2"

  Sprite constant.
- static final String **RIGHT_1** = "right_1"

  Sprite constant.
- static final String **RIGHT_2** = "right_2"

  Sprite constant.
- static final String **UP_1** = "up_1"

  Sprite constant.
- static final String **UP_2** = "up_2"

  Sprite constant.
- static final String **DOWN_1** = "down_1"

  Sprite constant.
- static final String **DOWN_2** = "down_2"

  Sprite constant.

3.1.1 Detailed Description

This class represents Blinky.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Blinky()

```
Blinky (
            int x,
            int y )
```

Constructor for Blinky.

Parameters

x	x-coordinate
y	y-coordinate

3.1.3 Member Function Documentation

3.1.3.1 getSprite()

```
BufferedImage getSprite ( )
```

Gets the sprite for Blinky.

Returns

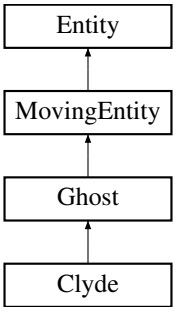
current sprite

Reimplemented from Ghost.

3.2 Clyde Class Reference

This class represents Clyde.

Inheritance diagram for Clyde:



Public Member Functions

- Clyde (int x, int y)

  Constructor for Clyde.
- BufferedImage getSprite ()

  Gets the sprite for Clyde.
- void **update** ()

  Moves Clyde.

Public Member Functions inherited from Ghost

- boolean isEdible ()

  Checks if the ghost is edible.
- void eatenBy (PacMan pacMan)

  Tries to eat the ghost, fails if the ghost is not frightened.
- void update (Point newLocation, String newDirection, boolean canEatGhosts)

  Updates the ghost's location, direction, and frightened state.
- boolean isDead ()

  Checks if the ghost is dead.

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Additional Inherited Members

Protected Member Functions inherited from Ghost

- Ghost (int x, int y)

  Constructor for the ghost.

Protected Member Functions inherited from MovingEntity

- MovingEntity (int x, int y)

  Constructor for the moving entity.
- void **getNextAnimation** ()

  Sets the sprite to the next one in the animation.

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes inherited from Ghost

- boolean **isFrightened**

  True if the ghost is frightened, false otherwise.
- boolean **isDead**

  True if the ghost is dead, false otherwise.
- Point **pacManLocation**

  The current location of PacMan.
- String **pacManDirection**

  The current direction of PacMan.
- int **deathTimer**

  The number of frames the ghost is dead for.

Protected Attributes inherited from MovingEntity

- int **speedX**

  Stores the speed on the x axis of the entity.
- int **speedY**

  Stores the speed on the y axis of the entity.
- String **currentDirection** = NEUTRAL

  Stores the current direction of the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

Static Protected Attributes inherited from MovingEntity

- static final String **NEUTRAL** = "neutral"
  Sprite constant.
- static final String **LEFT_1** = "left_1"
  Sprite constant.
- static final String **LEFT_2** = "left_2"
  Sprite constant.
- static final String **RIGHT_1** = "right_1"
  Sprite constant.
- static final String **RIGHT_2** = "right_2"
  Sprite constant.
- static final String **UP_1** = "up_1"
  Sprite constant.
- static final String **UP_2** = "up_2"
  Sprite constant.
- static final String **DOWN_1** = "down_1"
  Sprite constant.
- static final String **DOWN_2** = "down_2"
  Sprite constant.

3.2.1 Detailed Description

This class represents Clyde.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Clyde()

```
Clyde (
            int x,
            int y )
```

Constructor for Clyde.

Parameters

x	x-coordinate
y	y-coordinate

3.2.3 Member Function Documentation

3.2.3.1 getSprite()

```
BufferedImage getSprite ( )
```

Gets the sprite for Clyde.

Returns

current sprite

Reimplemented from Ghost.

3.3 EditorViewPanel Class Reference

This class is responsible for the visual representation and logic of the level editor.

Inherits JPanel.

Public Member Functions

- **EditorViewPanel** ()
  Constructor for the level editor view panel.
- LevelData getLevelData ()
- void loadEntities (ArrayList< ArrayList< Entity > > entities, HashMap< String, Point > locations)
  Loads the level into the level editor.

3.3.1 Detailed Description

This class is responsible for the visual representation and logic of the level editor.

3.3.2 Member Function Documentation

3.3.2.1 getLevelData()

```
LevelData getLevelData ( )
```

Returns

the level data of the level editor

3.3.2.2 loadEntities()

```
void loadEntities (
            ArrayList< ArrayList< Entity > > entities,
            HashMap< String, Point > locations )
```

Loads the level into the level editor.

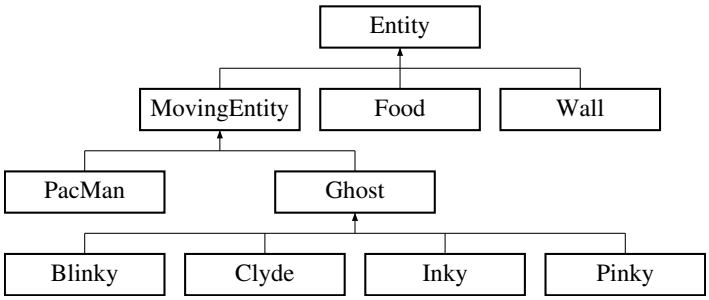
Parameters

entities	the entities to be loaded
locations	the locations of the ghosts and PacMan

3.4 Entity Class Reference

Abstract class for all entities in the game.

Inheritance diagram for Entity:



Public Member Functions

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- abstract BufferedImage getSprite ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- boolean isEdible ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void **update** ()

  Moves the entity.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Protected Member Functions

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

3.4.1 Detailed Description

Abstract class for all entities in the game.

Entities are objects that are drawn on the screen and can be interacted with.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Entity()

```
Entity (
            int x,
            int y )  [protected]
```

Constructor for the entity.

Parameters

x	The x coordinate of the entity.
y	The y coordinate of the entity.

3.4.3 Member Function Documentation

3.4.3.1 addObserver()

```
void addObserver (
            EntityObserver observer )
```

Adds an observer to the entity.

Parameters

observer	The observer to be added.

3.4.3.2 draw()

```
void draw (
            Graphics g )
```

Draws the entity on the screen.

Parameters

g	Graphics object used to draw the entity.

3.4.3.3 eatenBy()

```
void eatenBy (
            PacMan pacMan )
```

Called when the entity is eaten by PacMan.

Parameters

pacMan	PacMan

Reimplemented in Food.

3.4.3.4 getHeight()

```
int getHeight ( )
```

Returns

The height of the entity.

3.4.3.5 getSprite()

```
abstract BufferedImage getSprite ( )  [abstract]
```

Returns

The sprite of the entity.

Reimplemented in Blinky, Clyde, Ghost, Inky, Pinky, PacMan, Food, and Wall.

3.4.3.6 getWidth()

```
int getWidth ( )
```

Returns

The width of the entity.

3.4.3.7 getX()

```
int getX ( )
```

Returns

The x coordinate of the entity.

3.4.3.8 getY()

```
int getY ( )
```

Returns

The y coordinate of the entity.

3.4.3.9 isEdible()

```
boolean isEdible ( )
```

Returns

Whether the entity is edible.

Reimplemented in Ghost, PacMan, and Food.

3.4.3.10 isTraversableByGhosts()

```
boolean isTraversableByGhosts ( )
```

Returns

Whether the entity is traversable by ghosts.

Reimplemented in Wall.

3.4.3.11 isTraversableByPacMan()

```
boolean isTraversableByPacMan ( )
```

Returns

Whether the entity is traversable by PacMan.

Reimplemented in Wall.

3.4.3.12 setSprite()

```
void setSprite (
            String spriteName )
```

Sets the sprite of the entity.

Parameters

spriteName	The name of the sprite.

3.5 EntityChooserPanel Class Reference

This class is responsible for everything concerning the entity chooser panel.

Inherits JPanel.

Public Member Functions

- **EntityChooserPanel** ()
  Constructor for the entity chooser panel.

3.5.1 Detailed Description

This class is responsible for everything concerning the entity chooser panel.

3.6 EntityObserver Class Reference

This class is used to observe the state of the game.

Public Member Functions

- **EntityObserver** ()
  Constructor for the entity observer.
- void addScore (int score)
  Adds the given score to the current score.
- int getScore ()
- void setFoodCount (int foodCount)
  Sets the food count.
- int getFoodCount ()
- void **incrementEaten** ()
  Increments the number of eaten food.
- boolean shouldInkyLeaveHome ()
  Determines whether Blinky should leave home.
- boolean shouldClydeLeaveHome ()
  Determines whether Pinky should leave home.

3.6.1 Detailed Description

This class is used to observe the state of the game.

3.6.2 Member Function Documentation

3.6.2.1 addScore()

```
void addScore (
            int score )
```

Adds the given score to the current score.

Parameters

score	The score to be added

3.6.2.2 getFoodCount()

```
int getFoodCount ( )
```

Returns

The food count

3.6.2.3 getScore()

```
int getScore ( )
```

Returns

The current score

3.6.2.4 setFoodCount()

```
void setFoodCount (
            int foodCount )
```

Sets the food count.

Parameters

foodCount	The food count

3.6.2.5 shouldClydeLeaveHome()

```
boolean shouldClydeLeaveHome ( )
```

Determines whether Pinky should leave home.

Returns

True if more than a third of the food has been eaten, false otherwise

3.6.2.6 shouldInkyLeaveHome()

```
boolean shouldInkyLeaveHome ( )
```

Determines whether Blinky should leave home.

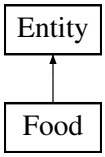
Returns

True if more than 30 food has been eaten, false otherwise

3.7 Food Class Reference

This class represents a food entity.

Inheritance diagram for Food:



Public Member Functions

- Food (int x, int y)
  Constructor for the food entity.
- BufferedImage getSprite ()
  Gets the sprite of the food entity.
- boolean isEdible ()
  Checks if the food entity is edible.
- void eatenBy (PacMan pacMan)
  Eats the food entity.

Public Member Functions inherited from Entity

- void draw (Graphics g)
  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)
  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()
  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()
  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- void **update** ()
  Moves the entity.
- void addObserver (EntityObserver observer)
  Adds an observer to the entity.

Additional Inherited Members

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

3.7.1 Detailed Description

This class represents a food entity.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 Food()

```
Food (
            int x,
            int y )
```

Constructor for the food entity.

Parameters

x	The x coordinate of the food entity
y	The y coordinate of the food entity

3.7.3 Member Function Documentation

3.7.3.1 eatenBy()

```
void eatenBy (
            PacMan pacMan )
```

Eats the food entity.

Parameters

pacMan	The PacMan that eats the food entity

Reimplemented from Entity.

3.7.3.2 getSprite()

```
BufferedImage getSprite ( )
```

Gets the sprite of the food entity.

Returns

The sprite of the food entity

Reimplemented from Entity.

3.7.3.3 isEdible()

```
boolean isEdible ( )
```

Checks if the food entity is edible.

Returns

Always true

Reimplemented from Entity.

3.8 GameConstants Class Reference

This class stores the constants used in the game.

Static Public Attributes

- static final int **ROW_COUNT** = 31
  The number of rows in the map.
- static final int **COLUMN_COUNT** = 28
  The number of columns in the map.
- static final int **CELL_SIZE** = 22
  The size of each cell in the map.
- static final int **TIMER_DELAY** = 150
  The delay between each frame of the game.
- static final String **PACMAN** = "pacman"
  PacMan constant.
- static final String **BLINKY** = "blinky"
  Blinky constant.
- static final String **INKY** = "inky"
  Inky constant.
- static final String **PINKY** = "pinky"
  Pinky constant.
- static final String **CLYDE** = "clyde"
  Clyde constant.

3.8.1 Detailed Description

This class stores the constants used in the game.

3.9 GameFrame Class Reference

GameFrame class.

Inherits JFrame.

Public Member Functions

- **GameFrame** ()
  Constructor.

3.9.1 Detailed Description

GameFrame class.

3.10 GamePanel Class Reference

This class is responsible for the visual representation of the game.

Inherits JPanel.

Public Member Functions

- **GamePanel** ()
  Constructor.
- Dimension getPreferredSize ()
- void paint (Graphics g)

3.10.1 Detailed Description

This class is responsible for the visual representation of the game.

3.10.2 Member Function Documentation

3.10.2.1 getPreferredSize()

```
Dimension getPreferredSize ( )
```

Returns

the preferred size of the panel

3.10.2.2 paint()

```
void paint (
            Graphics g )
```

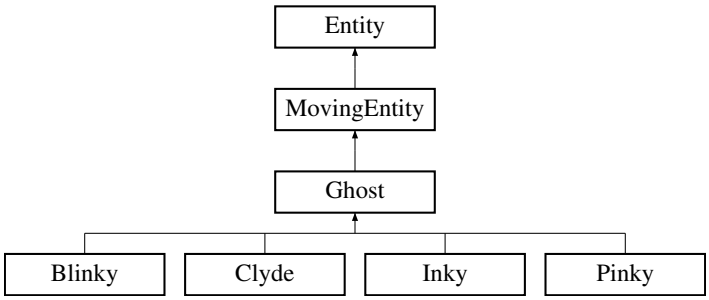
Parameters

g	the `Graphics` context in which to paint

3.11 Ghost Class Reference

This class represents an abstract ghost.

Inheritance diagram for Ghost:



Public Member Functions

- abstract BufferedImage getSprite ()

  Gets the sprite of the ghost.
- boolean isEdible ()

  Checks if the ghost is edible.
- void eatenBy (PacMan pacMan)

  Tries to eat the ghost, fails if the ghost is not frightened.
- void update (Point newLocation, String newDirection, boolean canEatGhosts)

  Updates the ghost's location, direction, and frightened state.
- boolean isDead ()

  Checks if the ghost is dead.

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.

- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void **update** ()

  Moves the entity.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Protected Member Functions

- Ghost (int x, int y)

  Constructor for the ghost.

Protected Member Functions inherited from MovingEntity

- MovingEntity (int x, int y)

  Constructor for the moving entity.
- void **getNextAnimation** ()

  Sets the sprite to the next one in the animation.

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes

- boolean **isFrightened**

  True if the ghost is frightened, false otherwise.
- boolean **isDead**

  True if the ghost is dead, false otherwise.
- Point **pacManLocation**

  The current location of PacMan.
- String **pacManDirection**

  The current direction of PacMan.
- int **deathTimer**

  The number of frames the ghost is dead for.

Protected Attributes inherited from MovingEntity

- int **speedX**

  Stores the speed on the x axis of the entity.
- int **speedY**

  Stores the speed on the y axis of the entity.
- String **currentDirection** = NEUTRAL

  Stores the current direction of the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

Additional Inherited Members

Static Protected Attributes inherited from MovingEntity

- static final String **NEUTRAL** = "neutral"

  Sprite constant.
- static final String **LEFT_1** = "left_1"

  Sprite constant.
- static final String **LEFT_2** = "left_2"

  Sprite constant.
- static final String **RIGHT_1** = "right_1"

  Sprite constant.
- static final String **RIGHT_2** = "right_2"

  Sprite constant.
- static final String **UP_1** = "up_1"

  Sprite constant.
- static final String **UP_2** = "up_2"

  Sprite constant.
- static final String **DOWN_1** = "down_1"

  Sprite constant.
- static final String **DOWN_2** = "down_2"

  Sprite constant.

3.11.1 Detailed Description

This class represents an abstract ghost.

3.11.2 Constructor & Destructor Documentation

3.11.2.1 Ghost()

```
Ghost (
            int x,
            int y )  [protected]
```

Constructor for the ghost.

Parameters

x	x-coordinate
y	y-coordinate

3.11.3 Member Function Documentation

3.11.3.1 eatenBy()

```
void eatenBy (
            PacMan pacMan )
```

Tries to eat the ghost, fails if the ghost is not frightened.

Parameters

pacMan	the PacMan that eats the ghost

3.11.3.2 getSprite()

```
abstract BufferedImage getSprite ( )  [abstract]
```

Gets the sprite of the ghost.

Returns

the sprite of the ghost

Reimplemented from Entity.

Reimplemented in Blinky, Clyde, Inky, and Pinky.

3.11.3.3 isDead()

```
boolean isDead ( )
```

Checks if the ghost is dead.

Returns

true if the ghost is dead, false otherwise

3.11.3.4 isEdible()

```
boolean isEdible ( )
```

Checks if the ghost is edible.

Returns

true if the ghost is frightened, false otherwise

Reimplemented from Entity.

3.11.3.5 update()

```
void update (
            Point newLocation,
            String newDirection,
            boolean canEatGhosts )
```

Updates the ghost's location, direction, and frightened state.

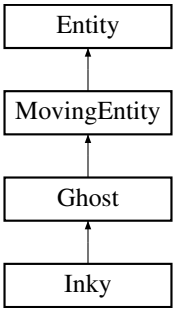
Parameters

newLocation	the new location of the ghost
newDirection	the new direction of the ghost
canEatGhosts	true if PacMan can eat the ghost, false otherwise

3.12 Inky Class Reference

This class represents Inky.

Inheritance diagram for Inky:



Public Member Functions

- Inky (int x, int y)

  Constructor for Inky.

- BufferedImage getSprite ()

  Gets the sprite for Inky.
- void **update** ()

  Moves Inky.
- void setObservedBlinky (Blinky observedBlinky)

  Sets the observed Blinky.

Public Member Functions inherited from Ghost

- boolean isEdible ()

  Checks if the ghost is edible.
- void eatenBy (PacMan pacMan)

  Tries to eat the ghost, fails if the ghost is not frightened.
- void update (Point newLocation, String newDirection, boolean canEatGhosts)

  Updates the ghost's location, direction, and frightened state.
- boolean isDead ()

  Checks if the ghost is dead.

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Additional Inherited Members

Protected Member Functions inherited from Ghost

- Ghost (int x, int y)

  Constructor for the ghost.

Protected Member Functions inherited from MovingEntity

- MovingEntity (int x, int y)

  Constructor for the moving entity.
- void **getNextAnimation** ()

  Sets the sprite to the next one in the animation.

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes inherited from Ghost

- boolean **isFrightened**

  True if the ghost is frightened, false otherwise.
- boolean **isDead**

  True if the ghost is dead, false otherwise.
- Point **pacManLocation**

  The current location of PacMan.
- String **pacManDirection**

  The current direction of PacMan.
- int **deathTimer**

  The number of frames the ghost is dead for.

Protected Attributes inherited from MovingEntity

- int **speedX**

  Stores the speed on the x axis of the entity.
- int **speedY**

  Stores the speed on the y axis of the entity.
- String **currentDirection** = NEUTRAL

  Stores the current direction of the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

Static Protected Attributes inherited from MovingEntity

- static final String **NEUTRAL** = "neutral"

  Sprite constant.
- static final String **LEFT_1** = "left_1"

  Sprite constant.
- static final String **LEFT_2** = "left_2"

  Sprite constant.
- static final String **RIGHT_1** = "right_1"

  Sprite constant.
- static final String **RIGHT_2** = "right_2"

  Sprite constant.
- static final String **UP_1** = "up_1"

  Sprite constant.
- static final String **UP_2** = "up_2"

  Sprite constant.
- static final String **DOWN_1** = "down_1"

  Sprite constant.
- static final String **DOWN_2** = "down_2"

  Sprite constant.

3.12.1 Detailed Description

This class represents Inky.

3.12.2 Constructor & Destructor Documentation

3.12.2.1 Inky()

```
Inky (
            int x,
            int y )
```

Constructor for Inky.

Parameters

x	x-coordinate
y	y-coordinate

3.12.3 Member Function Documentation

3.12.3.1 getSprite()

```
BufferedImage getSprite ( )
```

Gets the sprite for Inky.

Returns

the sprite for Inky

Reimplemented from Ghost.

3.12.3.2 setObservedBlinky()

```
void setObservedBlinky (
            Blinky observedBlinky )
```

Sets the observed Blinky.

Parameters

observedBlinky	the observed Blinky

3.13 LevelData Class Reference

This class is used to store a level's data.

Inherits Serializable.

Public Member Functions

- LevelData (ArrayList< ArrayList< Entity > > entities, HashMap< String, Point > locations, int foodCount)
  Constructor for the level data.
- ArrayList< ArrayList< Entity > > getEntities ()
- HashMap< String, Point > getLocations ()
- int getFoodCount ()

3.13.1 Detailed Description

This class is used to store a level's data.

3.13.2 Constructor & Destructor Documentation

3.13.2.1 LevelData()

```
LevelData (
            ArrayList< ArrayList< Entity > > entities,
            HashMap< String, Point > locations,
            int foodCount )
```

Constructor for the level data.

Parameters

entities	The entities in the level
locations	The locations of the ghosts and PacMan
foodCount	The number of food in the level

3.13.3 Member Function Documentation

3.13.3.1 getEntities()

```
ArrayList< ArrayList< Entity > > getEntities ( )
```

Returns

The entities in the level

3.13.3.2 getFoodCount()

```
int getFoodCount ( )
```

Returns

The number of food in the level

3.13.3.3 getLocations()

```
HashMap< String, Point > getLocations ( )
```

Returns

The locations of the ghosts and PacMan

3.14 LevelEditorFrame Class Reference

This class is responsible for everything concerning the level editor.

Inherits JFrame.

Public Member Functions

- **LevelEditorFrame** ()

  Constructor for the level editor frame.

Static Public Member Functions

- static String getCurrentSprite ()
- static String getCurrentEntityType ()
- static void setCurrentSprite (String spriteName)

  Set the current sprite that is selected in the level editor.
- static void setCurrentEntityType (String entityType)

  Set the current entity type that is selected in the level editor.

3.14.1 Detailed Description

This class is responsible for everything concerning the level editor.

It is a JFrame that contains a EntityChooserPanel and an EditorViewPanel. It also contains a save button and a load button.

3.14.2 Member Function Documentation

3.14.2.1 getCurrentEntityType()

```
static String getCurrentEntityType ( )  [static]
```

Returns

the current entity type that is selected in the level editor

3.14.2.2 getCurrentSprite()

```
static String getCurrentSprite ( )  [static]
```

Returns

the current sprite that is selected in the level editor

3.14.2.3 setCurrentEntityType()

```
static void setCurrentEntityType (
            String entityType )  [static]
```

Set the current entity type that is selected in the level editor.

Parameters

entityType	the type of the entity that is selected

3.14.2.4 setCurrentSprite()

```
static void setCurrentSprite (
            String spriteName )  [static]
```

Set the current sprite that is selected in the level editor.

Parameters

spriteName	the name of the sprite that is selected

3.15 Main Class Reference

The entry point of the program.

Static Public Member Functions

- static void main (String[ ] args)
  The entry point of the program.
- static void setDisplayedFrame (JFrame f)
  Sets the frame to be displayed.

Static Public Attributes

- static final Logger **logger** = Logger.getLogger("Error")
  The logger for the program.

3.15.1 Detailed Description

The entry point of the program.

3.15.2 Member Function Documentation

3.15.2.1 main()

```
static void main (
            String[] args )  [static]
```

The entry point of the program.

Parameters

args	The command line arguments

3.15.2.2 setDisplayedFrame()

```
static void setDisplayedFrame (
            JFrame f )  [static]
```

Sets the frame to be displayed.

Parameters

f	The frame to be displayed

3.16 MainMenuFrame Class Reference

The main window of the application.

Inherits JFrame.

Public Member Functions

- **MainMenuFrame** ()
  Constructor.

3.16.1 Detailed Description

The main window of the application.

3.17 MainMenuPanel Class Reference

The visual representation of the main menu.

Inherits JPanel.

Public Member Functions

- **MainMenuPanel** ()
  Constructor.

3.17.1 Detailed Description

The visual representation of the main menu.

3.18 MenuPacAnimationPanel Class Reference

This class is responsible for the Pac-Man animation in the main menu.

Inherits JPanel.

Public Member Functions

- **MenuPacAnimationPanel** ()
  Constructor.
- Dimension getPreferredSize ()
- void paint (Graphics g)
  Paints Pac-Man and the food entities.

3.18.1 Detailed Description

This class is responsible for the Pac-Man animation in the main menu.

3.18.2 Member Function Documentation

3.18.2.1 getPreferredSize()

```
Dimension getPreferredSize ( )
```

Returns

the preferred size of the panel

3.18.2.2 paint()

```
void paint (
            Graphics g )
```

Paints Pac-Man and the food entities.

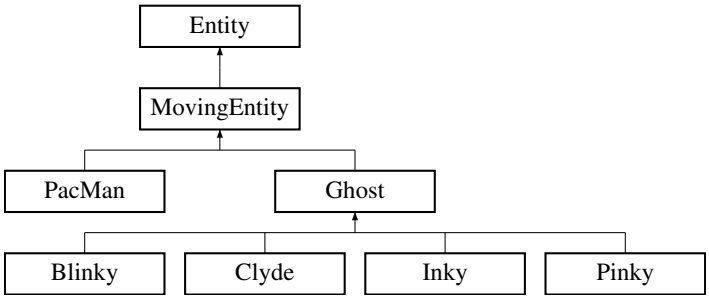
Parameters

g	the `Graphics` context in which to paint

3.19 MovingEntity Class Reference

Abstract class for all moving entities in the game.

Inheritance diagram for MovingEntity:



Protected Member Functions

- MovingEntity (int x, int y)

  Constructor for the moving entity.
- void **getNextAnimation** ()

  Sets the sprite to the next one in the animation.

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes

- int **speedX**

  Stores the speed on the x axis of the entity.
- int **speedY**

  Stores the speed on the y axis of the entity.
- String **currentDirection** = NEUTRAL

  Stores the current direction of the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

Static Protected Attributes

- static final String **NEUTRAL** = "neutral"

  Sprite constant.
- static final String **LEFT_1** = "left_1"

  Sprite constant.
- static final String **LEFT_2** = "left_2"

  Sprite constant.
- static final String **RIGHT_1** = "right_1"

  Sprite constant.
- static final String **RIGHT_2** = "right_2"

  Sprite constant.
- static final String **UP_1** = "up_1"

  Sprite constant.
- static final String **UP_2** = "up_2"

  Sprite constant.
- static final String **DOWN_1** = "down_1"

  Sprite constant.
- static final String **DOWN_2** = "down_2"

  Sprite constant.

Additional Inherited Members

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- abstract BufferedImage getSprite ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- boolean isEdible ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void **update** ()

  Moves the entity.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

3.19.1 Detailed Description

Abstract class for all moving entities in the game.

3.19.2 Constructor & Destructor Documentation

3.19.2.1 MovingEntity()

```
MovingEntity (
            int x,
            int y )  [protected]
```

Constructor for the moving entity.

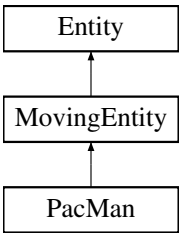
Parameters

x	The x coordinate of the entity.
y	The y coordinate of the entity.

3.20 PacMan Class Reference

This class represents the PacMan entity.

Inheritance diagram for PacMan:



Public Member Functions

- PacMan (int x, int y)

  Constructor for PacMan.
- void **update** ()

  Updates PacMan's location and checks if he is facing a ghost or not If he is, he either eats the ghost or dies If he is facing a cell that is traversable, he moves to that cell.
- void menuMove (int screenWidth)

  Used for the menu screen.
- void setInitialDirection (String direction)

  Sets the initial direction of PacMan (only used for the menu screen).
- BufferedImage getSprite ()

  Sets the current sprite of PacMan based on his previous sprite and the direction he is facing.
- boolean isEdible ()
- boolean isDead ()
- void keyPressed (KeyEvent e)

  Sets the current direction of PacMan based on the key pressed.
- void addObserver (Ghost ghost)

  Adds a ghost to the list of observers.
- void setCanEatGhosts (boolean canEatGhosts)

  Sets whether PacMan can eat ghosts or not.
- boolean canEatGhosts ()

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Additional Inherited Members

Protected Member Functions inherited from MovingEntity

- MovingEntity (int x, int y)

  Constructor for the moving entity.
- void **getNextAnimation** ()

  Sets the sprite to the next one in the animation.

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes inherited from MovingEntity

- int **speedX**

  Stores the speed on the x axis of the entity.
- int **speedY**

  Stores the speed on the y axis of the entity.
- String **currentDirection** = NEUTRAL

  Stores the current direction of the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

Static Protected Attributes inherited from MovingEntity

- static final String **NEUTRAL** = "neutral"

  Sprite constant.
- static final String **LEFT_1** = "left_1"

  Sprite constant.
- static final String **LEFT_2** = "left_2"

  Sprite constant.
- static final String **RIGHT_1** = "right_1"

  Sprite constant.
- static final String **RIGHT_2** = "right_2"

  Sprite constant.
- static final String **UP_1** = "up_1"

  Sprite constant.
- static final String **UP_2** = "up_2"

  Sprite constant.
- static final String **DOWN_1** = "down_1"

  Sprite constant.
- static final String **DOWN_2** = "down_2"

  Sprite constant.

3.20.1 Detailed Description

This class represents the PacMan entity.

3.20.2 Constructor & Destructor Documentation

3.20.2.1 PacMan()

```
PacMan (
            int x,
            int y )
```

Constructor for PacMan.

Parameters

x	x-coordinate
y	y-coordinate

3.20.3 Member Function Documentation

3.20.3.1 addObserver()

```
void addObserver (
            Ghost ghost )
```

Adds a ghost to the list of observers.

Parameters

ghost	the ghost to be added

3.20.3.2 canEatGhosts()

```
boolean canEatGhosts ( )
```

Returns

whether PacMan can eat ghosts or not

3.20.3.3 getSprite()

```
BufferedImage getSprite ( )
```

Sets the current sprite of PacMan based on his previous sprite and the direction he is facing.

Returns

the current sprite of PacMan

Reimplemented from Entity.

3.20.3.4 isDead()

```
boolean isDead ( )
```

Returns

whether PacMan is dead or not

3.20.3.5 isEdible()

```
boolean isEdible ( )
```

Returns

Whether the entity is edible.

Reimplemented from Entity.

3.20.3.6 keyPressed()

```
void keyPressed (
            KeyEvent e )
```

Sets the current direction of PacMan based on the key pressed.

Parameters

e	the key pressed

3.20.3.7 menuMove()

```
void menuMove (
            int screenWidth )
```

Used for the menu screen.

Moves PacMan to the right.

Parameters

screenWidth	the width of the menu window

3.20.3.8 setCanEatGhosts()

```
void setCanEatGhosts (
            boolean canEatGhosts )
```

Sets whether PacMan can eat ghosts or not.

Parameters

canEatGhosts	whether PacMan can eat ghosts or not

3.20.3.9 setInitialDirection()

```
void setInitialDirection (
            String direction )
```

Sets the initial direction of PacMan (only used for the menu screen).

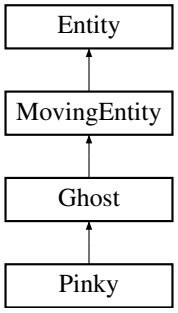
Parameters

direction	the initial direction of PacMan

3.21 Pinky Class Reference

This class represents Pinky.

Inheritance diagram for Pinky:



Public Member Functions

- Pinky (int x, int y)
  Constructor for Pinky.
- BufferedImage getSprite ()
  Gets the sprite for Pinky.
- void **update** ()
  Moves the entity.

Public Member Functions inherited from Ghost

- boolean isEdible ()
  Checks if the ghost is edible.
- void eatenBy (PacMan pacMan)
  Tries to eat the ghost, fails if the ghost is not frightened.
- void update (Point newLocation, String newDirection, boolean canEatGhosts)
  Updates the ghost's location, direction, and frightened state.
- boolean isDead ()
  Checks if the ghost is dead.

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Additional Inherited Members

Protected Member Functions inherited from Ghost

- Ghost (int x, int y)

  Constructor for the ghost.

Protected Member Functions inherited from MovingEntity

- MovingEntity (int x, int y)

  Constructor for the moving entity.
- void **getNextAnimation** ()

  Sets the sprite to the next one in the animation.

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes inherited from Ghost

- boolean **isFrightened**

  True if the ghost is frightened, false otherwise.
- boolean **isDead**

  True if the ghost is dead, false otherwise.
- Point **pacManLocation**

  The current location of PacMan.
- String **pacManDirection**

  The current direction of PacMan.
- int **deathTimer**

  The number of frames the ghost is dead for.

Protected Attributes inherited from MovingEntity

- int **speedX**

  Stores the speed on the x axis of the entity.
- int **speedY**

  Stores the speed on the y axis of the entity.
- String **currentDirection** = NEUTRAL

  Stores the current direction of the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

Static Protected Attributes inherited from MovingEntity

- static final String **NEUTRAL** = "neutral"

  Sprite constant.
- static final String **LEFT_1** = "left_1"

  Sprite constant.
- static final String **LEFT_2** = "left_2"

  Sprite constant.
- static final String **RIGHT_1** = "right_1"

  Sprite constant.
- static final String **RIGHT_2** = "right_2"

  Sprite constant.
- static final String **UP_1** = "up_1"

  Sprite constant.
- static final String **UP_2** = "up_2"

  Sprite constant.
- static final String **DOWN_1** = "down_1"

  Sprite constant.
- static final String **DOWN_2** = "down_2"

  Sprite constant.

3.21.1 Detailed Description

This class represents Pinky.

3.21.2 Constructor & Destructor Documentation

3.21.2.1 Pinky()

```
Pinky (
            int x,
            int y )
```

Constructor for Pinky.

Parameters

x	x-coordinate
y	y-coordinate

3.21.3 Member Function Documentation

3.21.3.1 getSprite()

```
BufferedImage getSprite ( )
```

Gets the sprite for Pinky.

Returns

current sprite

Reimplemented from Ghost.

3.22 ResourceHandler Class Reference

This class handles the usage of resources (sprites, audio, etc.)

Static Public Member Functions

- static void **init** ()
  Initializes the resources.
- static HashMap< String, BufferedImage > getSpriteMap (String entityType)
- static Font getPacFont ()
- static Image getIcon (String iconName)
- static HashSet< String > getEntityTypes ()
- static void saveLevel (String fileName, LevelData levelData)
  Save level to a file.
- static LevelData loadLevel (String fileName)
  Load level from a file.
- static void levelSelectDialog (JComponent component)
  Show a dialog that allows the user to select a level.
- static ArrayList< ArrayList< Entity > > getLevelEntities ()
- static HashMap< String, Point > getInitialLocations ()
- static int getFoodCount ()

3.22.1 Detailed Description

This class handles the usage of resources (sprites, audio, etc.)

3.22.2 Member Function Documentation

3.22.2.1 getEntityTypes()

```
static HashSet< String > getEntityTypes ( )  [static]
```

Returns

The set of entity types

3.22.2.2 getFoodCount()

```
static int getFoodCount ( )  [static]
```

Returns

the number of food entities in the current level

3.22.2.3 getIcon()

```
static Image getIcon (
            String iconName )  [static]
```

Parameters

iconName	The name of the icon

Returns

The icon with the given name

3.22.2.4 getInitialLocations()

```
static HashMap< String, Point > getInitialLocations ( )  [static]
```

Returns

the initial locations of the ghosts and pacman in the current level

3.22.2.5 getLevelEntities()

```
static ArrayList< ArrayList< Entity > > getLevelEntities ( )  [static]
```

Returns

the current level

3.22.2.6 getPacFont()

```
static Font getPacFont ( )  [static]
```

Returns

the custom font used by the application

3.22.2.7 getSpriteMap()

```
static HashMap< String, BufferedImage > getSpriteMap (
            String entityType )  [static]
```

Parameters

entityType	The type of entity

Returns

The sprite map for the given entity type

3.22.2.8 levelSelectDialog()

```
static void levelSelectDialog (
            JComponent component )  [static]
```

Show a dialog that allows the user to select a level.

Parameters

component	the component that the dialog is displayed on

3.22.2.9 loadLevel()

```
static LevelData loadLevel (
            String fileName )  [static]
```

Load level from a file.

Parameters

fileName	the name of the file

Returns

the level that was loaded

3.22.2.10 saveLevel()

```
static void saveLevel (
            String fileName,
            LevelData levelData )  [static]
```

Save level to a file.

Parameters

fileName	the name of the file
levelData	the level to be saved

3.23 ShortestPathFinder Class Reference

This class is used to find the shortest path between two points.

Static Public Member Functions

- static Point findNextCellForShortestPath (Point start, Point end)

  Finds the shortest path between two points using the BFS algorithm.

3.23.1 Detailed Description

This class is used to find the shortest path between two points.

3.23.2 Member Function Documentation

3.23.2.1 findNextCellForShortestPath()

```
static Point findNextCellForShortestPath (
            Point start,
            Point end )  [static]
```

Finds the shortest path between two points using the BFS algorithm.

Parameters

start	the `Point` where the path starts
end	the `Point` where the path ends

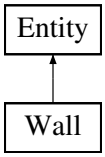
Returns

the next `Point` on the shortest path

3.24 Wall Class Reference

This class represents a wall entity.

Inheritance diagram for Wall:



Public Member Functions

- Wall (int x, int y)

  Constructor for the wall entity.
- BufferedImage getSprite ()

  Gets the sprite of the wall entity.
- boolean isTraversableByPacMan ()
- boolean isTraversableByGhosts ()

Public Member Functions inherited from Entity

- void draw (Graphics g)

  Draws the entity on the screen.
- int getX ()
- int getY ()
- int getWidth ()
- int getHeight ()
- void setSprite (String spriteName)

  Sets the sprite of the entity.
- void **setNotTraversableByPacMan** ()

  Sets the entity to be not traversable by PacMan.
- void **setNotTraversableByGhosts** ()

  Sets the entity to be not traversable by ghosts.
- boolean isEdible ()
- void eatenBy (PacMan pacMan)

  Called when the entity is eaten by PacMan.
- void **update** ()

  Moves the entity.
- void addObserver (EntityObserver observer)

  Adds an observer to the entity.

Additional Inherited Members

Protected Member Functions inherited from Entity

- Entity (int x, int y)

  Constructor for the entity.

Protected Attributes inherited from Entity

- int **x**

  Stores the x coordinate of the entity.
- int **y**

  Stores the y coordinate of the entity.
- String **spriteName**

  Stores the name of the sprite of the entity.
- boolean **traversableByPacMan** = true

  Stores whether the entity is traversable by PacMan.
- boolean **traversableByGhosts** = true

  Stores whether the entity is traversable by ghosts.
- transient EntityObserver **observer**

  Stores the observer of the entity.

3.24.1 Detailed Description

This class represents a wall entity.

3.24.2 Constructor & Destructor Documentation

3.24.2.1 Wall()

```
Wall (
            int x,
            int y )
```

Constructor for the wall entity.

Parameters

x	x-coordinate
y	y-coordinate

3.24.3 Member Function Documentation

3.24.3.1 getSprite()

```
BufferedImage getSprite ( )
```

Gets the sprite of the wall entity.

Returns

The sprite of the wall entity

Reimplemented from Entity.

3.24.3.2 isTraversableByGhosts()

```
boolean isTraversableByGhosts ( )
```

Returns

True if "empty" or PacMan non-traversable wall, false otherwise

Reimplemented from Entity.

3.24.3.3 isTraversableByPacMan()

```
boolean isTraversableByPacMan ( )
```

Returns

True if "empty" wall, false otherwise

Reimplemented from Entity.