

Mensch ärgere dich nicht

Schnittstellenbeschreibung



Package

de.spareripsproduction.madn.client

"Mensch aergere dich nicht" Package

Contains all the stuff that is necessary, for a computer version of "Mensch aergere dich nicht"

de.spareripsproduction.madn.client Class Game

```
java.lang.Object
├── de.spareripsproduction.tinyengine.Game
│   └── de.spareripsproduction.madn.client.Game
```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

```
public class Game
extends Game
implements RenderInterface, UpdateInterface
```

This Class is responsible for showing the correct Scene (Game, Settings, Credits, etc) Created by marian on 12/03/14.

Fields

ENV_DEV

```
public static final java.lang.String ENV_DEV
```

DEV environment shows FPS
Constant value: **DEV**

ENV_PROD

```
public static final java.lang.String ENV_PROD
```

PROD does not show FPS
Constant value: **PROD**

ENV_TEST

```
public static final java.lang.String ENV_TEST
```

for testing purposes
Constant value: **TEST**

Constructors

Game

```
public Game()
```

Default Constructor

Methods

getInstance

```
public static Game getInstance()
```

(continued on next page)

(continued from last page)

Singleton of the Game

Returns:

Singleton

getInstance

```
public static Game getInstance(String environment)
```

Singleton of the Game with environment variable

Returns:

Singleton

render

```
public void render(java.awt.Graphics2D context)
```

update

```
public void update()
```

loadScene

```
public boolean loadScene(String sceneName)
```

Loads a specific scene, and unloads the old one

Parameters:

sceneName - Name of the Scene

Returns:

true if Scene was found false if not

getEnvironment

```
public String getEnvironment()
```

Returns:

current environment

de.spareripsproduction.madn.client Class Main

java.lang.Object

└--de.spareripsproduction.madn.client.Main

public class **Main**
extends Object

Created by marian on 08/02/14.

Constructors

Main

public **Main**()

Methods

main

public static void **main**(String[] args)

Starting point of the game

Parameters:

args - Application Start Args

Package

**de.spareripsproduction.madn.client.grap
hics**

Lorem ipsum

de.spareripsproduction.madn.client.graphics Class Board

```
java.lang.Object
├── de.spareripsproduction.tinyengine.entity.Entity
│   └── de.spareripsproduction.madn.client.graphics.Board
```

All Implemented Interfaces:

[RenderAndUpdateable](#), [UpdateInterface](#), [RenderInterface](#)

```
public class Board
extends Entity
implements RenderInterface, UpdateInterface, RenderAndUpdateable
```

Created by marian on 12/03/14.

Constructors

Board

```
public Board(int x,
             int y)
```

Parameters:

x - x-coordinate

y - y-coordinate

Methods

getInstance

```
public static Board getInstance()
```

Returns:

Singleton instance of the game

update

```
public void update()
```

render

```
public void render(java.awt.Graphics2D g)
```

(continued from last page)

getGameFigures

```
public ArrayList getGameFigures()
```

Returns:

a list of all GameFigures

getPlayers

```
public ArrayList getPlayers()
```

Returns:

a list of all players

getActivePlayer

```
public Player getActivePlayer()
```

getDice

```
public Dice getDice()
```

Returns:

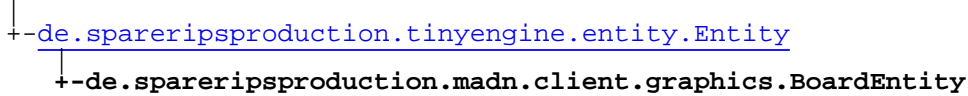
the dice for this board

isGameOver

```
public boolean isGameOver()
```


de.spareripsproduction.madn.client.graphics Class BoardEntity

java.lang.Object



All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

Direct Known Subclasses:

[GameFigure](#), [Field](#)

public class **BoardEntity**
extends [Entity](#)

Since:

2014-04-13

Fields

FIELD_SIZE

public static final int **FIELD_SIZE**

Field Size in px
Constant value: **48**

id

protected int **id**

boardPosition

protected java.awt.Point **boardPosition**

hover

protected boolean **hover**

clicked

protected boolean **clicked**

Constructors

(continued from last page)

BoardEntity

```
public BoardEntity(String spriteRef,  
                    int id)
```

Parameters:

`spriteRef` - path to a sprite image
`id` - id-position of the board

Methods

getX

```
public float getX()
```

getY

```
public float getY()
```

update

```
public void update()
```

getBoardPosition

```
public java.awt.Point getBoardPosition()
```

Returns:

the position of this entity, in a matrix of fields

setId

```
protected void setId(int id)
```

Every accessible Field on the board has a unique id

Parameters:

`id` - id-position

getId

```
public int getId()
```

Every accessible Field on the board has a unique id

Returns:

id-position of the this entity

updateBoardPosition

protected java.awt.Point **updateBoardPosition**()

isHover

protected boolean **isHover**()

Returns:

true if the entity is hovered by the mouse

isClicked

public boolean **isClicked**()

Returns:

true if this entity is clicked

insideView

protected boolean **insideView**(java.awt.Point p)

de.spareripsproduction.madn.client.graphics

Interface RenderAndUpdateable

All Known Implementing Classes:

[Scene](#), [Dice](#), [Player](#), [GameFigure](#), [Field](#), [Board](#)

public interface **RenderAndUpdateable**
extends

Created by marian on 12/03/14.

Methods

update

```
public void update()
```

This Method will called during the Update Cycle, **before** the rendering starts

render

```
public void render(java.awt.Graphics2D g)
```

This function gets called every circle and draws the object

Parameters:

context - on which we draw

Package

**de.spareripsproduction.madn.client.grap
hics.field**

Lorem ipsum

de.spareripsproduction.madn.client.graphics.field

Class Field

```
java.lang.Object
├-- de.spareripsproduction.tinyengine.entity.Entity
│   ├── de.spareripsproduction.madn.client.graphics.BoardEntity
│   │   └-- de.spareripsproduction.madn.client.graphics.field.Field
```

All Implemented Interfaces:

[RenderAndUpdateable](#), [UpdateInterface](#), [RenderInterface](#)

Direct Known Subclasses:

[NormalField](#), [SpawnField](#)

public abstract class **Field**

extends [BoardEntity](#)

implements [RenderInterface](#), [UpdateInterface](#), [RenderAndUpdateable](#)

Created by marian on 12/03/14.

Fields

SPRITE_NORMAL

```
public static final java.lang.String SPRITE_NORMAL
```

Reference to the sprite image of the **normal** Field
Constant value: **sprites/normalField.png**

SPRITE_RED

```
public static final java.lang.String SPRITE_RED
```

Reference to the sprite image of the **red** Field
Constant value: **sprites/SpawnFieldRed.png**

SPRITE_YELLOW

```
public static final java.lang.String SPRITE_YELLOW
```

Reference to the sprite image of the **yellow** Field
Constant value: **sprites/SpawnFieldYellow.png**

SPRITE_BLUE

```
public static final java.lang.String SPRITE_BLUE
```

Reference to the sprite image of the **blze** Field
Constant value: **sprites/SpawnFieldBlue.png**

(continued from last page)

SPRITE_GREEN

```
public static final java.lang.String SPRITE_GREEN
```

Reference to the sprite image of the **green** `Field`

Constant value: **`sprites/SpawnFieldGreen.png`**

Constructors

Field

```
public Field(String spriteRef,  
             int id)
```

de.spareripsproduction.madn.client.graphics.field Class HomeEntryField

```
java.lang.Object
├-- de.spareripsproduction.tinyengine.entity.Entity
│   ├── de.spareripsproduction.madn.client.graphics.BoardEntity
│   │   ├── de.spareripsproduction.madn.client.graphics.field.Field
│   │   │   ├── de.spareripsproduction.madn.client.graphics.field.SpawnField
│   │   │   └-- de.spareripsproduction.madn.client.graphics.field.HomeEntryField
```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#), [RenderAndUpdateable](#)

```
public class HomeEntryField
extends SpawnField
```

Created by marian on 12/03/14.

Constructors

HomeEntryField

```
public HomeEntryField(String spriteRef,
                      int id)
```


de.spareripsproduction.madn.client.graphics.field Class NormalField

```
java.lang.Object
├── de.spareripsproduction.tinyengine.entity.Entity
│   ├── de.spareripsproduction.madn.client.graphics.BoardEntity
│   │   ├── de.spareripsproduction.madn.client.graphics.field.Field
│   │   │   └── de.spareripsproduction.madn.client.graphics.field.NormalField
```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#), [RenderAndUpdateable](#)

```
public class NormalField
extends Field
```

Created by marian on 12/03/14.

Constructors

NormalField

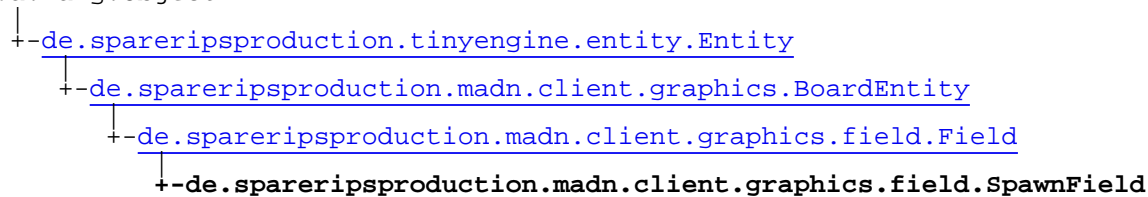
```
public NormalField(int id)
```

NormalField

```
public NormalField(String spriteRef,
                    int id)
```

de.spareripsproduction.madn.client.graphics.field Class SpawnField

java.lang.Object



All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#), [RenderAndUpdateable](#)

Direct Known Subclasses:

[HomeEntryField](#)

```
public class SpawnField
extends Field
```

Created by marian on 12/03/14.

Constructors

SpawnField

```
public SpawnField(String spriteRef,
                  int id)
```

Package

de.spareripsproduction.madn.client.graphics.figure

This Package contains the `GameFigure` related code

de.spareripsproduction.madn.client.graphics.figure Class BlueFigure

```
java.lang.Object
├── de.spareripsproduction.tinyengine.entity.Entity
│   ├── de.spareripsproduction.madn.client.graphics.BoardEntity
│   │   ├── de.spareripsproduction.madn.client.graphics.figure.GameFigure
│   │   │   └── de.spareripsproduction.madn.client.graphics.figure.BlueFigure
```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#), [RenderAndUpdateable](#)

```
public class BlueFigure
extends GameFigure
```

Since:

2014-04-13

Fields

startId

```
protected int startId
```

homeStartId

```
protected int homeStartId
```

Constructors

BlueFigure

```
public BlueFigure(int index)
```

Methods

updateBoardPosition

```
protected java.awt.Point updateBoardPosition()
```

getStartId

```
protected int getStartId()
```

(continued from last page)

getHomeStartId

```
public int getHomeStartId()
```

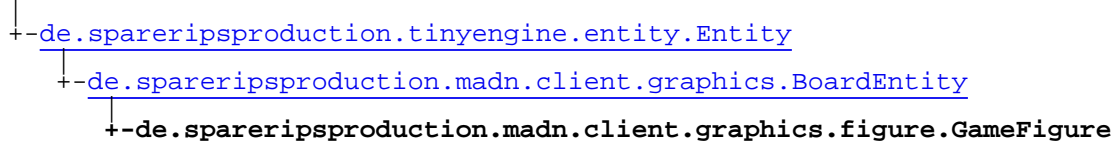
getPlayerType

```
protected int getPlayerType()
```

de.spareripsproduction.madn.client.graphics.figure

Class GameFigure

java.lang.Object



All Implemented Interfaces:

[RenderAndUpdateable](#), [UpdateInterface](#), [RenderInterface](#)

Direct Known Subclasses:

[BlueFigure](#), [GreenFigure](#), [RedFigure](#), [YellowFigure](#)

public abstract class **GameFigure**

extends [BoardEntity](#)

implements [RenderInterface](#), [UpdateInterface](#), [RenderAndUpdateable](#)

Created by marian on 12/03/14.

Fields

SPRITE_RED

public static final java.lang.String **SPRITE_RED**

Reference to the sprite image of the red GameFigure
Constant value: **sprites/figureRed**

SPRITE_YELLOW

public static final java.lang.String **SPRITE_YELLOW**

Reference to the sprite image of the yellow GameFigure
Constant value: **sprites/figureYellow**

SPRITE_BLUE

public static final java.lang.String **SPRITE_BLUE**

Reference to the sprite image of the red GameFigure
Constant value: **sprites/figureBlue**

SPRITE_GREEN

public static final java.lang.String **SPRITE_GREEN**

Reference to the sprite image of the green GameFigure
Constant value: **sprites/figureGreen**

(continued from last page)

IN_HOUSE_ID

```
public static final int IN_HOUSE_ID
```

id-position for GameFigures in house
Constant value: -1

index

```
protected int index
```

hoverSprite

```
protected de.spareripsproduction.tinyengine.graphics.Sprite hoverSprite
```

owner

```
protected de.spareripsproduction.madn.client.logic.Player owner
```

Constructors

GameFigure

```
public GameFigure(String spriteRef,  
                  int id,  
                  int index)
```

Parameters:

spriteRef - sprite of the figure
id - position on the field
index - figure index 0 to 3

Methods

getY

```
public float getY()
```

kick

```
public void kick()
```

Gamefigure gets place into the beginning house

move

```
protected boolean move(int delta,  
                       boolean dryRun)
```

(continued from last page)

move

```
public boolean move(int delta)
```

Parameters:

delta - number of fields to move (1-6)

Returns:

true if move is successful, false if not

isFieldOccupied

```
public boolean isFieldOccupied(int id)
```

Checks if Field is occupied by GameFigure from the same Owner

Parameters:

id - id-position

Returns:

true if the field is occupied, false if not

isStartIdOccupied

```
public boolean isStartIdOccupied()
```

canMove

```
public boolean canMove(int delta)
```

Parameters:

delta - number of fields to move (1-6)

Returns:

true if GameFigure can make the move of delta steps, false if not

getStartId

```
protected abstract int getStartId()
```

getHomeStartId

```
public abstract int getHomeStartId()
```

(continued from last page)

getPlayerType

```
protected abstract int getPlayerType()
```

render

```
public void render(java.awt.Graphics2D g)
```

setId

```
protected void setId(int id)
```

Every accessible Field on the board has a unique id

getOwner

```
protected Player getOwner()
```

isActive

```
protected boolean isActive()
```

de.spareripsproduction.madn.client.graphics.figure

Class GreenFigure

```
java.lang.Object
├── de.spareripsproduction.tinyengine.entity.Entity
│   ├── de.spareripsproduction.madn.client.graphics.BoardEntity
│   │   ├── de.spareripsproduction.madn.client.graphics.figure.GameFigure
│   │   │   └── de.spareripsproduction.madn.client.graphics.figure.GreenFigure
```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#), [RenderAndUpdateable](#)

```
public class GreenFigure
extends GameFigure
```

Since:

2014-04-13

Fields

startId

```
protected int startId
```

homeStartId

```
protected int homeStartId
```

Constructors

GreenFigure

```
public GreenFigure(int index)
```

Methods

updateBoardPosition

```
protected java.awt.Point updateBoardPosition()
```

getStartId

```
protected int getStartId()
```

(continued from last page)

getHomeStartId

```
public int getHomeStartId()
```

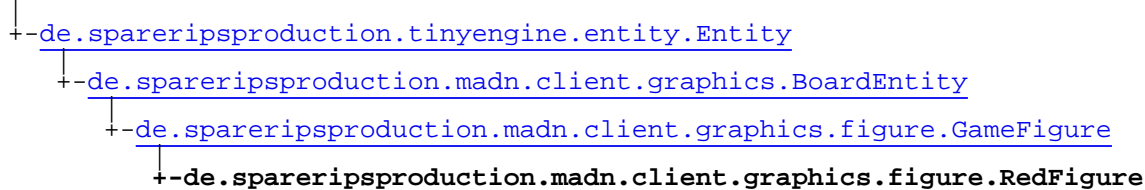
getPlayerType

```
protected int getPlayerType()
```

de.spareripsproduction.madn.client.graphics.figure

Class RedFigure

java.lang.Object



All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#), [RenderAndUpdateable](#)

public class **RedFigure**
extends [GameFigure](#)

Since:

2014-04-13

Fields

startId

protected int **startId**

homeStartId

protected int **homeStartId**

Constructors

RedFigure

public **RedFigure**(int index)

Methods

updateBoardPosition

protected java.awt.Point **updateBoardPosition**()

getStartId

protected int **getStartId**()

(continued from last page)

getHomeStartId

```
public int getHomeStartId()
```

getPlayerType

```
protected int getPlayerType()
```

de.spareripsproduction.madn.client.graphics.figure Class YellowFigure

```
java.lang.Object
├── de.spareripsproduction.tinyengine.entity.Entity
│   ├── de.spareripsproduction.madn.client.graphics.BoardEntity
│   │   ├── de.spareripsproduction.madn.client.graphics.figure.GameFigure
│   │   │   └── de.spareripsproduction.madn.client.graphics.figure.YellowFigure
```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#), [RenderAndUpdateable](#)

```
public class YellowFigure
extends GameFigure
```

Since:

2014-04-13

Fields

startId

```
protected int startId
```

homeStartId

```
protected int homeStartId
```

Constructors

YellowFigure

```
public YellowFigure(int index)
```

Methods

updateBoardPosition

```
protected java.awt.Point updateBoardPosition()
```

getStartId

```
protected int getStartId()
```

(continued from last page)

getHomeStartId

```
public int getHomeStartId()
```

getPlayerType

```
protected int getPlayerType()
```

Package

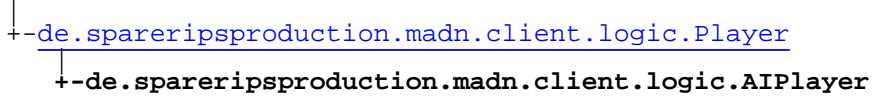
de.spareripsproduction.madn.client.logic

This package contains some logic heavy entities

de.spareripsproduction.madn.client.logic

Class AIPlayer

java.lang.Object



All Implemented Interfaces:

[RenderAndUpdateable](#)

```
public class AIPlayer
extends Player
```

Since:

2014-04-19

Constructors

AIPlayer

```
public AIPlayer(int type)
```

Methods

makeMove

```
public void makeMove()
```

called when player is active, process player input to make next move

de.spareripsproduction.madn.client.logic Class Dice

java.lang.Object

└─de.spareripsproduction.madn.client.logic.Dice

All Implemented Interfaces:

[RenderAndUpdateable](#)

```
public class Dice
  extends Object
  implements RenderAndUpdateable
```

The Dice Created by marian on 12/03/14.

Fields

hover

protected boolean **hover**

clicked

protected boolean **clicked**

locked

protected boolean **locked**

Constructors

Dice

```
public Dice(int x,
            int y)
```

Default Constructor

Parameters:

x - x-coordinate

y - y-coordinate

Methods

getNextNumber

```
public int getNextNumber()
```

(continued from last page)

Rolls the dice again

Returns:
the next randomly generated number

reset

```
public void reset()
```

Resets the dice, so no number is displayed

update

```
public void update()
```

setX

```
public void setX(int x)
```

setY

```
public void setY(int y)
```

render

```
public void render(java.awt.Graphics2D g)
```

renderOne

```
protected void renderOne(java.awt.Graphics2D g)
```

renderTwo

```
protected void renderTwo(java.awt.Graphics2D g)
```

renderThree

```
protected void renderThree(java.awt.Graphics2D g)
```

renderFour

```
protected void renderFour(java.awt.Graphics2D g)
```

renderFive

```
protected void renderFive(java.awt.Graphics2D g)
```

renderSix

```
protected void renderSix(java.awt.Graphics2D g)
```

renderCircle

```
protected void renderCircle(java.awt.Graphics2D g,  
    int x,  
    int y,  
    int radius)
```

getLastNumber

```
public int getLastNumber()
```

getX

```
public int getX()
```

getY

```
public int getY()
```

getWidth

```
public int getWidth()
```

getHeight

```
public int getHeight()
```

isHover

```
protected boolean isHover()
```

isClicked

```
protected boolean isClicked()
```

insideView

```
protected boolean insideView(java.awt.Point p)
```

unlock

```
public void unlock()
```

frees the dice, then the dice can be rolled again

isLocked

```
public boolean isLocked()
```

checks if the dice can be rolled

Returns:

true if dice can be rolled false if not

lock

```
public void lock()
```

de.spareripsproduction.madn.client.logic

Class Player

java.lang.Object

└--de.spareripsproduction.madn.client.logic.Player

All Implemented Interfaces:

[RenderAndUpdateable](#)

Direct Known Subclasses:

[AIPlayer](#)

```
public class Player
  extends Object
  implements RenderAndUpdateable
```

Player Class Created by marian on 12/03/14.

Fields

RED_PLAYER

```
public static final int RED_PLAYER
```

Constant value: 0

BLUE_PLAYER

```
public static final int BLUE_PLAYER
```

Constant value: 1

GREEN_PLAYER

```
public static final int GREEN_PLAYER
```

Constant value: 2

YELLOW_PLAYER

```
public static final int YELLOW_PLAYER
```

Constant value: 3

type

```
protected int type
```

active

protected boolean **active**

rollCount

protected int **rollCount**

name

protected java.lang.String **name**

last

protected static long **last**

nameLabel

protected de.spareripsproduction.tinyengine.gui.TELabel **nameLabel**

position

protected int **position**

winner

protected static int **winner**

Constructors

Player

public **Player**(int type)

Parameters:

type - PlayerType: Player.RED_PLAYER, Player.BLUE_PLAYER, Player.GREEN_PLAYER, Player.YELLOW_PLAYER

Methods

(continued from last page)

makeMove

```
public void makeMove()
```

called when player is active, process player input to make next move

update

```
public void update()
```

canRollDiceThreeTimes

```
protected boolean canRollDiceThreeTimes()
```

canRollDiceAgain

```
protected boolean canRollDiceAgain()
```

canMove

```
protected boolean canMove()
```

getDice

```
protected Dice getDice()
```

activate

```
public void activate()
```

Activates player, so he can roll the dice and move

nextPlayer

```
public void nextPlayer()
```

Activates next player

render

```
public void render(java.awt.Graphics2D g)
```

getGameFigures

```
public ArrayList getGameFigures()
```

(continued from last page)

Returns:a list of gameFigures from this player

isActive

```
public boolean isActive()
```

Returns:true if active, false if not

getType

```
public int getType()
```

Returns:Player.RED_PLAYER or Player.BLUE_PLAYER or Player.GREEN_PLAYER or Player.YELLOW_PLAYER

isFinished

```
public boolean isFinished()
```

getPosition

```
public int getPosition()
```

scoreLabelStr

```
public String scoreLabelStr()
```

de.spareripsproduction.madn.client.logic

Class Settings

java.lang.Object

└--de.spareripsproduction.madn.client.logic.Settings

public class **Settings**
extends Object

Stores Settings Created by atissen on 26.03.14.

Fields

Player1Name

public static java.lang.String **Player1Name**

Stores Name for Player One

Player2Name

public static java.lang.String **Player2Name**

Stores Name for Player One

Player3Name

public static java.lang.String **Player3Name**

Stores Name for Player One

Player4Name

public static java.lang.String **Player4Name**

Stores Name for Player One

playerCount

public static int **playerCount**

Stores How many People are playing

Constructors

Settings

public **Settings**()

Package

de.spareripsproduction.madn.client.scene

This package holds the scenes.

A scene is responsible for displaying a certain part of the game. For instance the Settings or the Intro.

de.spareripsproduction.madn.client.scene

Class CreditsScene

java.lang.Object

└- [de.spareripsproduction.madn.client.scene.Scene](#)

└- **de.spareripsproduction.madn.client.scene.CreditsScene**

All Implemented Interfaces:

[RenderAndUpdateable](#)

public class **CreditsScene**

extends [Scene](#)

Since:

2014-04-15

Fields

background

protected de.spareripsproduction.tinyengine.entity.Entity **background**

fatRaccoon

protected javax.swing.ImageIcon **fatRaccoon**

showRaccoon

protected boolean **showRaccoon**

backButton

protected de.spareripsproduction.tinyengine.gui.TEButton **backButton**

title

protected de.spareripsproduction.tinyengine.gui.TELabel **title**

creditsCollection

protected de.spareripsproduction.tinyengine.gui.TECollectionVertical **creditsCollection**

Constructors

CreditsScene

```
public CreditsScene()
```

Methods

load

```
public void load()
```

Loads this scene, is called when the scene is to be showed

unload

```
public void unload()
```

Unloads this scene, is called when the scene disappears

update

```
public void update()
```

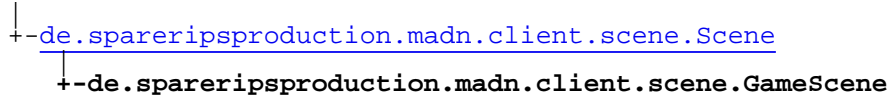
render

```
public void render(java.awt.Graphics2D g)
```

de.spareripsproduction.madn.client.scene

Class GameScene

java.lang.Object



All Implemented Interfaces:

[RenderAndUpdateable](#)

```
public class GameScene  
extends Scene
```

Created by marian on 12/03/14.

Constructors

GameScene

```
public GameScene()
```

Methods

load

```
public void load()
```

Loads this scene, is called when the scene is to be showed

unload

```
public void unload()
```

Unloads this scene, is called when the scene disappears

update

```
public void update()
```

render

```
public void render(java.awt.Graphics2D g)
```

getBoard

```
public Board getBoard()
```

(continued from last page)

Returns:

the board on this scene

de.spareripsproduction.madn.client.scene Class IntroScene

java.lang.Object



All Implemented Interfaces:

[RenderAndUpdateable](#)

```
public class IntroScene  
extends Scene
```

Since:

2014-03-19

Constructors

IntroScene

```
public IntroScene()
```

Methods

load

```
public void load()
```

Loads this scene, is called when the scene is to be showed

unload

```
public void unload()
```

Unloads this scene, is called when the scene disappears

update

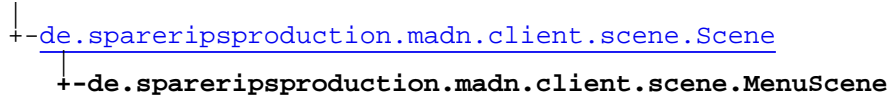
```
public void update()
```

render

```
public void render(java.awt.Graphics2D g)
```


de.spareripsproduction.madn.client.scene Class MenuScene

java.lang.Object



All Implemented Interfaces:
[RenderAndUpdateable](#)

```
public class MenuScene
extends Scene
```

Menu Scene Created by marian on 12/03/14.

Constructors

MenuScene

```
public MenuScene()
```

Methods

load

```
public void load()
```

Loads this scene, is called when the scene is to be showed

unload

```
public void unload()
```

Unloads this scene, is called when the scene disappears

update

```
public void update()
```

render

```
public void render(java.awt.Graphics2D g)
```

de.spareripsproduction.madn.client.scene

Class Scene

java.lang.Object

└─de.spareripsproduction.madn.client.scene.Scene

All Implemented Interfaces:

[RenderAndUpdateable](#)

Direct Known Subclasses:

[CreditsScene](#), [GameScene](#), [IntroScene](#), [MenuScene](#), [ScoreScene](#), [SettingsScene](#)

public abstract class **Scene**

extends Object

implements [RenderAndUpdateable](#)

The base scene Created by marian on 12/03/14.

Fields

SCENE_GAME

public static final java.lang.String **SCENE_GAME**

Game scene, actual game play happens here

Constant value: **game**

SCENE_MENU

public static final java.lang.String **SCENE_MENU**

Menu scene, the Menu

Constant value: **menu**

SCENE_SCORE

public static final java.lang.String **SCENE_SCORE**

Score scene, after the game ended

Constant value: **score**

SCENE_INTRO

public static final java.lang.String **SCENE_INTRO**

Intro scene, first scene

Constant value: **intro**

SCENE_SETTINGS

public static final java.lang.String **SCENE_SETTINGS**

Settings scene, configure the player count and player names

Constant value: **settings**

SCENE_CREDITS

```
public static final java.lang.String SCENE_CREDITS
```

Constant value: **credits**

Constructors

Scene

```
public Scene()
```

Methods

load

```
public abstract void load()
```

Loads this scene, is called when the scene is to be showed

unload

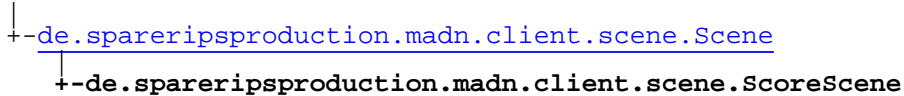
```
public abstract void unload()
```

Unloads this scene, is called when the scene disappears

de.spareripsproduction.madn.client.scene

Class ScoreScene

java.lang.Object



All Implemented Interfaces:

[RenderAndUpdateable](#)

```
public class ScoreScene
extends Scene
```

Created by marian on 12/03/14.

Fields

background

```
protected de.spareripsproduction.tinyengine.entity.Entity background
```

title

```
protected de.spareripsproduction.tinyengine.gui.TELabel title
```

Constructors

ScoreScene

```
public ScoreScene()
```

Methods

load

```
public void load()
```

Loads this scene, is called when the scene is to be showed

unload

```
public void unload()
```

Unloads this scene, is called when the scene disappears

(continued from last page)

update

```
public void update()
```

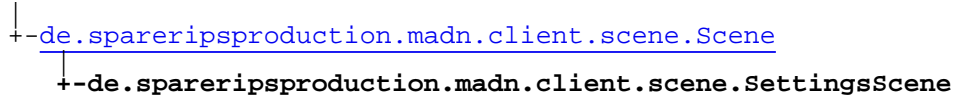
render

```
public void render(java.awt.Graphics2D g)
```

de.spareripsproduction.madn.client.scene

Class SettingsScene

java.lang.Object



All Implemented Interfaces:

[RenderAndUpdateable](#)

```
public class SettingsScene
extends Scene
```

Created by atissen on 26.03.14.

Constructors

SettingsScene

```
public SettingsScene()
```

Methods

load

```
public void load()
```

Loads this scene, is called when the scene is to be showed

unload

```
public void unload()
```

Unloads this scene, is called when the scene disappears

update

```
public void update()
```

render

```
public void render(java.awt.Graphics2D g)
```

Package

de.spareripsproduction.tinyengine

Provides the basic functionality to display graphics and ui, typically used by a 2D Game.

[TinyEngine on Github](#)

Initial purpose for this library was the Game "Mensch Ã¶rgere dich nicht"

de.spareripsproduction.tinyengine Class Core

java.lang.Object

└─de.spareripsproduction.tinyengine.Core

Deprecated.

public class **Core**
extends Object

Core class should no longer be used, will be removed

Since:

2014-03-08

Constructors

Core

public **Core**()

Deprecated.

Methods

log

public static void **log**(String s)

Deprecated.

log a message

Parameters:

s - Message to log

exit

public static void **exit**(int code)

Deprecated.

Exits the program

Parameters:

code - ExitCode

de.spareripsproduction.tinyengine

Class FontManager

java.lang.Object

└--de.spareripsproduction.tinyengine.FontManager

public class **FontManager**
extends Object

The FontManager Class is responsible loading fonts

Since:

2014-03-11

Fields

FONT_PRESS_START_2P

public static final java.lang.String **FONT_PRESS_START_2P**

Reference to the *Press Start 2P* True Type Font File, used by TinyEngine
Constant value: **fonts/PressStart2P-Regular.ttf**

FONT_COMIC_NEUE

public static final java.lang.String **FONT_COMIC_NEUE**

Reference to the *Comic Neue* True Type Font File
Constant value: **fonts/ComicNeue-Regular.ttf**

FONT_COMIC_NEUE_BOLD

public static final java.lang.String **FONT_COMIC_NEUE_BOLD**

Reference to the *Comic Neue* True Type Font File
Constant value: **fonts/ComicNeue-Bold-Oblique.ttf**

Constructors

FontManager

public **FontManager**()

Methods

getFont

public static java.awt.Font **getFont**(String resourceRef,
int fontSize)

Parameters:

(continued from last page)

`resourceRef` - Path to the ttf-Font file

`fontSize` - Size of the Font in px

Returns:

the Font

de.spareripsproduction.tinyengine

Class Game

java.lang.Object

└─de.spareripsproduction.tinyengine.Game

Direct Known Subclasses:

[Game](#)

```
public class Game
extends Object
```

Main Class for the Game, handles logic updates und rendering

Since:

2014-03-10

Constructors

Game

```
public Game(String title,
            int width,
            int height)
```

Default constructor

Parameters:

title - of the Window

width - of the Window in px

height - of the Window in px

Game

```
public Game(String title,
            int width,
            int height,
            UpdateInterface updater,
            RenderInterface renderer)
```

Constructor with update und render objects

Parameters:

title - of the Window

width - of the Window in px

height - of the Window in px

updater - Updates the Logic

renderer - Renders the window

Methods

run

```
public void run()
```

the game loop

setRenderer

```
public void setRenderer(RenderInterface renderer)
```

Sets the render

Parameters:

`renderer` - is responsible for rendering

setUpdater

```
public void setUpdater(UpdateInterface updater)
```

Sets the updater

Parameters:

`updater` - is responsible for updating

getWindow

```
public GameWindow getWindow()
```

the game window

Returns:

the Window

de.spareripsproduction.tinyengine Class GameWindow

```
java.lang.Object
├-- java.awt.Component
│   └-- java.awt.Canvas
│       └-- de.spareripsproduction.tinyengine.GameWindow
```

All Implemented Interfaces:

Serializable, java.awt.MenuContainer, java.awt.image.ImageObserver, javax.accessibility.Accessible

```
public class GameWindow
extends java.awt.Canvas
```

Responsible for the Game Window

Since:

2014-03-08

Constructors

GameWindow

```
public GameWindow(String title,
                  int width,
                  int height)
```

Constructor for the Game Window, should not be used, use getInstance() instead

Parameters:

title - of the Window
width - of the Window in px
height - of the Window in px

Methods

getInstance

```
public static GameWindow getInstance()
```

Singleton method with no params, should be used for getting the singleton

Returns:

Window singleton

getInstance

```
public static GameWindow getInstance(String title,
                                     int width,
                                     int height)
```

Single method should be used for creating a new Window

Parameters:

(continued from last page)

title - of the Window
width - of the Window in px
height - of the Window in px

Returns:

Window singleton

getGraphicsContext

```
public java.awt.Graphics2D getGraphicsContext()
```

Returns the context on which we draw

Returns:

the Graphics Context

repaint

```
public void repaint()
```

Repaints the window

setTitle

```
public void setTitle(String title)
```

Sets the Title of the window

Parameters:

title - new Title

getFrame

```
public javax.swing.JFrame getFrame()
```

Returns:

the frame of the Window

de.spareripsproduction.tinyengine

Class Timer

java.lang.Object

└─de.spareripsproduction.tinyengine.Timer

public class **Timer**
extends Object

The Standard Timer Class

Since:

2014-03-10

Constructors

Timer

public **Timer**()

Methods

getTime

public static long **getTime**()

Get current Time in milliseconds

Returns:

current Time in milliseconds

sleep

public static void **sleep**(long ms)

Pauses execution

Parameters:

ms - milliseconds to pause

Package

de.spareripsproduction.tinyengine.entity

Contains the Entity related things

de.spareripsproduction.tinyengine.entity

Class Entity

java.lang.Object

↳ `de.spareripsproduction.tinyengine.entity.Entity`

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

Direct Known Subclasses:

[Board](#), [BoardEntity](#)

```
public class Entity
extends Object
implements RenderInterface, UpdateInterface
```

Entity Object

Since:

2014-03-08

Fields

x

protected float **x**

y

protected float **y**

width

protected float **width**

height

protected float **height**

sprite

protected de.spareripsproduction.tinyengine.graphics.Sprite **sprite**

Constructors

(continued from last page)

Entity

```
public Entity(String spriteRef)
```

Entity

```
public Entity(String spriteRef,  
              float x,  
              float y)
```

Entity

```
public Entity(String spriteRef,  
              float x,  
              float y,  
              float width,  
              float height)
```

Methods

getIntX

```
public int getIntX()
```

getIntY

```
public int getIntY()
```

getIntHeight

```
public int getIntHeight()
```

getIntWidth

```
public int getIntWidth()
```

getX

```
public float getX()
```

(continued from last page)

getY

```
public float getY()
```

getHeight

```
public float getHeight()
```

getWidth

```
public float getWidth()
```

setLocation

```
public void setLocation(float x,  
                        float y)
```

setSize

```
public void setSize(float width,  
                  float height)
```

Sets the Size

Parameters:

width - in pixel

height - in pixel

move

```
public void move(float dX,  
                float dY)
```

Moves

Parameters:

dX - delta length

dY - delta height

getHitBox

```
protected java.awt.Rectangle getHitBox()
```

collidesWith

```
public boolean collidesWith(Entity e)
```

Collision detection

(continued from last page)

Parameters:

e - other entity

Returns:

true if entities collide, false if not

render

```
public void render(java.awt.Graphics2D context)
```

update

```
public void update()
```

Package

de.spareripsproduction.tinyengine.graphics

This package contains graphic elements

de.spareripsproduction.tinyengine.graphics

Interface RenderInterface

All Known Implementing Classes:

[Fps](#), [TEView](#), [Entity](#), [Game](#)

public interface **RenderInterface**
extends

Enables objects to render things on the screen

Since:

2014-03-10

Methods

render

```
public void render(java.awt.Graphics2D context)
```

This function gets called every circle and draws the object

Parameters:

context - on which we draw

de.spareripsproduction.tinyengine.graphics

Class Sprite

java.lang.Object

└─**de.spareripsproduction.tinyengine.graphics.Sprite**

public class **Sprite**
extends Object

Responsible for drawing simple images

Since:

2014-03-09

Constructors

Sprite

public **Sprite**(java.net.URL url)

Create a new sprite based on an image

Parameters:

url - The url that is the path th sprite

Methods

getWidth

public int **getWidth**()

Get the width of the drawn sprite

Returns:

The width in pixels of this sprite

getHeight

public int **getHeight**()

Get the height of the drawn sprite

Returns:

The height in pixels of this sprite

render

public void **render**(java.awt.Graphics g,
int x,
int y)

Draw the sprite onto the graphics context provided

Parameters:

g - The graphics context on which to draw the sprite

(continued from last page)

`x` - The x location at which to draw the sprite

`y` - The y location at which to draw the sprite

de.spareripsproduction.tinyengine.graphics

Class SpriteStore

java.lang.Object

└─de.spareripsproduction.tinyengine.graphics.SpriteStore

public class **SpriteStore**
extends Object

Holds all sprites in a HashMap, so they don't need to be created more than once

Since:

2014-03-09

Methods

getInstance

public static [SpriteStore](#) **getInstance**()

Get the singleton SpriteStore

Returns:

singleton

get

public [Sprite](#) **get**(String ref)

get a Sprite from the store

Parameters:

ref - Reference to the spriteImage

Returns:

Sprite

Package

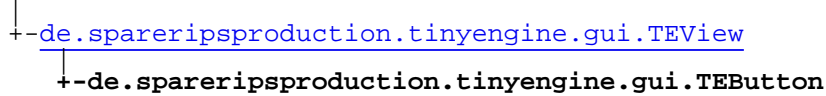
de.spareripsproduction.tinyengine.gui

this contains basic User Interface Elements, like Buttons, Labels and Textfields.

de.spareripsproduction.tinyengine.gui

Class TEButton

java.lang.Object



All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

Direct Known Subclasses:

[TESwitch](#)

public class **TEButton**

extends [TEView](#)

A Button

Since:

2014-03-21

Fields

label

protected de.spareripsproduction.tinyengine.gui.TELabel **label**

Constructors

TEButton

```
public TEButton(String label,  
                int x,  
                int y,  
                int width,  
                int height,  
                int borderSize,  
                java.awt.Font labelFont)
```

Construct with default colors

Parameters:

label - Label
x - x-coordinate
y - y-coordinate
width - in pixel
height - in pixel
borderSize - border thickness in pixel
labelFont - Font for the label

(continued from last page)

TEButton

```
public TEButton(String label,  
                int x,  
                int y,  
                int width,  
                int height,  
                int borderSize,  
                java.awt.Font labelFont,  
                java.awt.Color color,  
                java.awt.Color clickedColor,  
                java.awt.Color highlightColor)
```

Default Constructor

Parameters:

label - Label
x - x-coordinate
y - y-coordinate
width - in pixel
height - in pixel
borderSize - border thickness in pixel
labelFont - Font for the label
color - Color of the button
clickedColor - border color when clicked
highlightColor - color when button is hovered

Methods

getBorderSize

```
public int getBorderSize()
```

setBorderSize

```
public void setBorderSize(int borderSize)
```

render

```
public void render(java.awt.Graphics2D context)
```

update

```
public void update()
```

isClicked

```
public boolean isClicked()
```

Checks if button is clicked

Returns:

(continued from last page)

true if clicked

isHighlighted

```
public boolean isHighlighted()
```

Checks if button is highlighted

Returns:

true if highlighted

getColor

```
public java.awt.Color getColor()
```

Returns:

Color of this button

getClickedColor

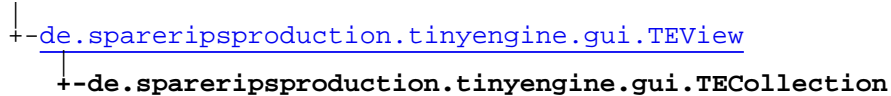
```
public java.awt.Color getClickedColor()
```

Returns:

color of this button in clicked state

de.spareripsproduction.tinyengine.gui Class TECollection

java.lang.Object



All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

Direct Known Subclasses:

[TECollectionVertical](#)

```
public class TECollection
extends TEView
```

Collection of Views

Since:

2014-03-22

Fields

collection

protected java.util.ArrayList **collection**

margin

protected int **margin**

Constructors

TECollection

```
public TECollection(int x,
                    int y,
                    int margin)
```

Default constructor

Parameters:

x - x-coordinate

y - y-coordinate

margin - margin between the views

Methods

(continued from last page)

addView

```
public void addView(TEView view)
```

Adds a TEView to the collection

Parameters:

view - View to be added

removeView

```
public void removeView(TEView view)
```

Removes a view form the collection

Parameters:

view - View to be removed

clearCollection

```
public void clearCollection()
```

Empty the collection

get

```
public TEView get(int index)
```

Get a view by index

Parameters:

index - of the view in the collection

Returns:

the View at the index

getCollection

```
public ArrayList getCollection()
```

render

```
public void render(java.awt.Graphics2D context)
```

update

```
public void update()
```

de.spareripsproduction.tinyengine.gui

Class TECollectionVertical

```

java.lang.Object
├── de.spareripsproduction.tinyengine.gui.TEView
│   ├── de.spareripsproduction.tinyengine.gui.TECollection
│       └── de.spareripsproduction.tinyengine.gui.TECollectionVertical

```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

```

public class TECollectionVertical
extends TECollection

```

Collection of views vertical stacked

Since:

2014-03-22

Constructors

TECollectionVertical

```

public TECollectionVertical(int x,
                           int y,
                           int margin)

```

Default constructor

Parameters:

x - x-coordinate

y - y-coordinate

margin - vertical margin between the views

Methods

addView

```

public void addView(TEView view)

```

Adds a TEView to the collection

removeView

```

public void removeView(TEView view)

```

Removes a view form the collection

setX

```

public void setX(int x)

```

Sets the x-coordinate

setY

```
public void setY(int y)
```

Sets the y-coordinate

verticalAlignCenter

```
public void verticalAlignCenter(int x1,  
                                int x2)
```

Aligns the View vertical between two coordinates

horizontalAlignCenter

```
public void horizontalAlignCenter(int y1,  
                                   int y2)
```

Aligns the View horizontal between two coordinates

verticalAlignRight

```
public void verticalAlignRight(int x)
```

Aligns the View horizontal right on

getHeight

```
public int getHeight()
```

Current height

getWidth

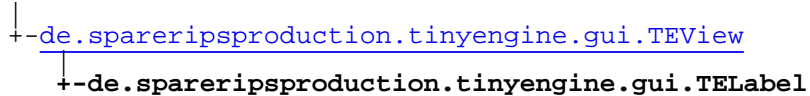
```
public int getWidth()
```

Current width

de.spareripsproduction.tinyengine.gui

Class TELabel

java.lang.Object



All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

```
public class TELabel  
extends TEView
```

Label

Since:

2014-03-21

Fields

textColor

protected java.awt.Color **textColor**

Constructors

TELabel

```
public TELabel(String text,  
               int x,  
               int y,  
               java.awt.Font font)
```

Parameters:

text
x
y
font

Methods

getHeight

```
public int getHeight()
```

Current height

getWidth

```
public int getWidth()
```

(continued from last page)

Current width

getText

```
public String getText()
```

setText

```
public void setText(String text)
```

getFont

```
public java.awt.Font getFont()
```

getY

```
public int getY()
```

current y-coordinate

setFont

```
public void setFont(java.awt.Font font)
```

render

```
public void render(java.awt.Graphics2D context)
```

de.spareripsproduction.tinyengine.gui

Class TESwitch

```
java.lang.Object
├── de.spareripsproduction.tinyengine.gui.TEView
│   ├── de.spareripsproduction.tinyengine.gui.TEButton
│       └── de.spareripsproduction.tinyengine.gui.TESwitch
```

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

```
public class TESwitch
extends TEButton
```

Since:

2014-03-23

Constructors

TESwitch

```
public TESwitch(int x,
                int y,
                int width,
                int height,
                int borderSize,
                java.awt.Font font)
```

constructor with default state off

Parameters:

x - in pixel
y - in pixel
width - in pixel
height - in pixel
borderSize - in pixel
font - Font

TESwitch

```
public TESwitch(int x,
                int y,
                int width,
                int height,
                int borderSize,
                java.awt.Font font,
                boolean state)
```

Default Constructor

Parameters:

x - in pixel
y - in pixel
width - in pixel
height - in pixel

(continued from last page)

borderSize - in pixel
font - Font
state - off = false, on = true

Methods

render

```
public void render(java.awt.Graphics2D context)
```

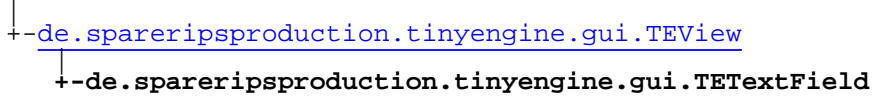
update

```
public void update()
```

de.spareripsproduction.tinyengine.gui

Class TETextField

java.lang.Object



All Implemented Interfaces:

java.awt.event.KeyListener, [UpdateInterface](#), [RenderInterface](#)

public class **TETextField**

extends [TEView](#)

implements [RenderInterface](#), [UpdateInterface](#), java.awt.event.KeyListener

Textfield

Since:

2014-03-25

Fields

label

protected de.spareripsproduction.tinyengine.gui.TELabel **label**

active

public boolean **active**

hover

protected boolean **hover**

size

protected int **size**

borderSize

protected int **borderSize**

(continued from last page)

borderColor

```
protected java.awt.Color borderColor
```

activeBorderColor

```
protected java.awt.Color activeBorderColor
```

hoverBorderColor

```
protected java.awt.Color hoverBorderColor
```

backgroundColor

```
protected java.awt.Color backgroundColor
```

textColor

```
protected java.awt.Color textColor
```

Constructors

TETextField

```
public TETextField(String text,  
                  int x,  
                  int y,  
                  int width,  
                  int height,  
                  java.awt.Font font,  
                  int size,  
                  int borderSize,  
                  java.awt.Color borderColor,  
                  java.awt.Color backgroundColor,  
                  java.awt.Color textColor)
```

Default Constructor

Parameters:

```
text  
x  
y  
width  
height  
font  
size  
borderSize  
borderColor  
backgroundColor  
textColor
```

TETextField

```
public TETextField(String text,  
                  int x,  
                  int y,  
                  int width,  
                  int height,  
                  java.awt.Font font,  
                  int size,  
                  int borderSize)
```

constructor with default colors

Parameters:

- text
- x
- y
- width
- height
- font
- size
- borderSize

Methods

update

```
public void update()
```

render

```
public void render(java.awt.Graphics2D context)
```

isActive

```
public boolean isActive()
```

currently selected and accepts input

Returns:

true if active

getBorderSize

```
public int getBorderSize()
```

returns borderthickness

Returns:

in pixel

getBackgroundColor

```
public java.awt.Color getBackgroundColor()
```


(continued from last page)

getActiveBorderColor

```
public java.awt.Color getActiveBorderColor()
```

isHover

```
public boolean isHover()
```

currently hovered by the mouse

Returns:

true if hovered

getSize

```
public int getSize()
```

keyTyped

```
public void keyTyped(java.awt.event.KeyEvent keyEvent)
```

keyPressed

```
public void keyPressed(java.awt.event.KeyEvent keyEvent)
```

keyReleased

```
public void keyReleased(java.awt.event.KeyEvent keyEvent)
```

getText

```
public String getText()
```

Text in the textfield

Returns:

textfield text

de.spareripsproduction.tinyengine.gui

Class TEView

java.lang.Object

↳ **de.spareripsproduction.tinyengine.gui.TEView**

All Implemented Interfaces:

[UpdateInterface](#), [RenderInterface](#)

Direct Known Subclasses:

[TEButton](#), [TECollection](#), [TELabel](#), [TETextField](#)

public class **TEView**

extends Object

implements [RenderInterface](#), [UpdateInterface](#)

Base Class for all views

Since:

2014-03-21

Constructors

TEView

```
public TEView(int x,  
              int y,  
              int width,  
              int height)
```

Default Constructor

Parameters:

x - X-Coordinate
y - Y-Coordinate
width - in pixel
height - in pixel

TEView

```
public TEView(int x,  
              int y)
```

Constructor for views with no height an width

Parameters:

x - X-Coordinate
y - Y-Coordinate

Methods

getX

```
public int getX()
```

(continued from last page)

Current x-coordinate

Returns:

x-coordinate

getY

```
public int getY()
```

current y-coordinate

Returns:

y-coordinate

getWidth

```
public int getWidth()
```

Current width

Returns:

width in pixel

getHeight

```
public int getHeight()
```

Current height

Returns:

height in pixel

setX

```
public void setX(int x)
```

Sets the x-coordinate

Parameters:

x - x-coordinate

setY

```
public void setY(int y)
```

Sets the y-coordinate

Parameters:

y - y-coordinate

setWidth

```
public void setWidth(int width)
```

Sets the width

Parameters:

width - in pixel

setHeight

```
public void setHeight(int height)
```

Sets the height

Parameters:

height - in px

render

```
public void render(java.awt.Graphics2D context)
```

update

```
public void update()
```

verticalAlignCenter

```
public void verticalAlignCenter(int x1,  
                                int x2)
```

Aligns the View vertical between two coordinates

Parameters:

x1 - top coordinate

x2 - bottom coordinate

verticalAlignRight

```
public void verticalAlignRight(int x)
```

Aligns the View horizontal right on

Parameters:

x - right coordinate

horizontalAlignCenter

```
public void horizontalAlignCenter(int y1,  
                                   int y2)
```

Aligns the View horizontal between two coordinates

Parameters:

y1 - left coordinate

y2 - right coordinate

insideView

```
protected boolean insideView(java.awt.Point p)
```

Checks if a point is inside this view

(continued from last page)

Parameters:

p - Point

Returns:

obvious

Package

de.spareripsproduction.tinyengine.input

This Package contains functionality, to respond to ser input (Keyboard, Mouse Interaction)

de.spareripsproduction.tinyengine.input Class Keyboard

java.lang.Object

└─de.spareripsproduction.tinyengine.input.Keyboard

All Implemented Interfaces:

java.awt.event.AWTEventListener

public class **Keyboard**
extends Object
implements java.awt.event.AWTEventListener

Keeps track of the Keyboard Input

Provides central access to *Keyboard Input*, without additional binding to the *Keyboard Events*

Since:

2014-03-10

Methods

getInstance

protected static [Keyboard](#) **getInstance**()

isPressed

public static boolean **isPressed**(int key)

Checks if a certain key is pressed

Parameters:

key - KeyEvent Code

Returns:

true if pressed, false if not

isKeyPressed

protected boolean **isKeyPressed**(int key)

keyPressed

protected void **keyPressed**(java.awt.event.KeyEvent e)

Notification of a key press

Parameters:

e - The event details

keyReleased

```
protected void keyReleased(java.awt.event.KeyEvent e)
```

Notification of a key release

Parameters:

e - The event details

eventDispatched

```
public void eventDispatched(java.awt.AWTEvent e)
```

Notification of an AWT event

Parameters:

e - The event details

de.spareripsproduction.tinyengine.input Class Mouse

java.lang.Object

└─de.spareripsproduction.tinyengine.input.Mouse

All Implemented Interfaces:

java.awt.event.AWTEventListener

```
public class Mouse
  extends Object
  implements java.awt.event.AWTEventListener
```

Since:

2014-03-22

Constructors

Mouse

```
public Mouse()
```

Methods

getInstance

```
public static Mouse getInstance()
```

position

```
public static java.awt.Point position(int button)
```

isClicked

```
public static boolean isClicked(int button)
```

location

```
public static java.awt.Point location()
```

buttonReleased

```
public void buttonReleased(java.awt.event.MouseEvent e)
```

(continued from last page)

Notification of a key release

Parameters:

e - The event details

moved

```
public void moved(java.awt.event.MouseEvent e)
```

eventDispatched

```
public void eventDispatched(java.awt.AWTEvent e)
```

Notification that an event has occurred in the AWT event system

Parameters:

e - The event details

Package

de.spareripsproduction.tinyengine.logic

Tis package contains the logic of TinyEngine

de.spareripsproduction.tinyengine.logic Class Fps

java.lang.Object

└─de.spareripsproduction.tinyengine.logic.Fps

All Implemented Interfaces:

[RenderInterface](#)

public class **Fps**
extends Object
implements [RenderInterface](#)

Objects from this Class, can measure the current **FPS** of the Game

Since:

2014-03-11

Constructors

Fps

public **Fps**()

Constructs the FPS-Meter

Methods

render

public void **render**(java.awt.Graphics2D context)

getFps

public int **getFps**()

Calculates the current *Frame per Seconds* of the Game

Returns:

Frames Per Second

de.spareripsproduction.tinyengine.logic Interface UpdateInterface

All Known Implementing Classes:

[TEView](#), [Entity](#), [Game](#)

public interface **UpdateInterface**
extends

The UpdateInterface enables Objects, to get called during the update-Cycle of the Game

Since:

2014-03-10

Methods

update

```
public void update()
```

This Method will called during the Update Cycle, **before** the rendering starts

Index

A

activate 40
active 39, 86
activeBorderColor 87
addView 78, 80
AIPlayer 33

B

backButton 44
background 44, 52
backgroundColor 87
BLUE_PLAYER 38
BlueFigure 20
Board 7
BoardEntity 9
boardPosition 9
borderColor 86
borderSize 86
buttonReleased 97

C

canMove 24, 40
canRollDiceAgain 40
canRollDiceThreeTimes 40
clearCollection 79
clicked 9, 34
collection 78
collidesWith 67
Core 56
creditsCollection 44
CreditsScene 44

D

Dice 34

E

Entity 65, 66
ENV_DEV 3

ENV_PROD 3
ENV_TEST 3
eventDispatched 96, 98
exit 56

F

fatRaccoon 44
Field 15
FIELD_SIZE 9
FONT_COMIC_NEUE 57
FONT_COMIC_NEUE_BOLD 57
FONT_PRESS_START_2P 57
FontManager 57
Fps 100

G

Game 3, 59
GameFigure 23
GameScene 46
GameWindow 61
get 73, 79
getActiveBorderColor 89
getActivePlayer 8
getBackgroundColor 88
getBoard 46
getBoardPosition 10
getBorderSize 76, 88
getClickedColor 77
getCollection 79
getColor 77
getDice 8, 40
getEnvironment 4
getFont 57, 83
getFps 100
getFrame 62
getGameFigures 7, 40
getGraphicsContext 62
getHeight 36, 67, 71, 81, 82, 91
getHitBox 67
getHomeStartId 21, 24, 27, 29, 31
getId 10
getInstance 3, 4, 7, 61, 73, 95, 97
getIntHeight 66

getIntWidth 66
getIntX 66
getIntY 66
getLastNumber 36
getNextNumber 34
getOwner 25
getPlayers 8
getPlayerType 21, 24, 27, 29, 31
getPosition 41
getSize 89
getStartId 20, 24, 26, 28, 30
getText 83, 89
getTime 63
getType 41
getWidth 36, 67, 71, 81, 82, 91
getWindow 60
getX 10, 36, 66, 90
getY 10, 23, 36, 66, 83, 91
GREEN_PLAYER 38
GreenFigure 26

H

height 65
HomeEntryField 16
homeStartId 20, 26, 28, 30
horizontalAlignCenter 81, 92
hover 9, 34, 86
hoverBorderColor 87
hoverSprite 23

I

id 9
IN_HOUSE_ID 22
index 23
insideView 11, 37, 92
IntroScene 48
isActive 25, 41, 88
isClicked 11, 37, 76, 97
isFieldOccupied 24
isFinished 41
isGameOver 8
isHighlighted 77
isHover 11, 36, 89

isKeyPressed 95
isLocked 37
isPressed 95
isStartIdOccupied 24

K

keyPressed 89, 95
keyReleased 89, 96
keyTyped 89
kick 23

L

label 75, 86
last 39
load 45, 46, 48, 49, 51, 52, 54
loadScene 4
location 97
lock 37
locked 34
log 56

M

Main 5
main 5
makeMove 33, 39
margin 78
MenuScene 49
Mouse 97
move 23, 24, 67
moved 98

N

name 39
nameLabel 39
nextPlayer 40
NormalField 17

O

owner 23

P

Player 39
Player1Name 42
Player2Name 42
Player3Name 42
Player4Name 42
playerCount 42
position 39, 97

R

RED_PLAYER 38
RedFigure 28
removeView 79, 80
render 4, 7, 12, 25, 35, 40, 45, 46, 48, 49, 53, 54, 68, 70, 71, 76,
79, 83, 85, 88, 92, 100
renderCircle 36
renderFive 36
renderFour 35
renderOne 35
renderSix 36
renderThree 35
renderTwo 35
repaint 62
reset 35
rollCount 39
run 59

S

Scene 51
SCENE_CREDITS 51
SCENE_GAME 50
SCENE_INTRO 50
SCENE_MENU 50
SCENE_SCORE 50
SCENE_SETTINGS 50
scoreLabelStr 41
ScoreScene 52
setBorderSize 76
setFont 83
setHeight 92
setId 10, 25
setLocation 67

setRenderer 60
setSize 67
setText 83
Settings 42
SettingsScene 54
setTitle 62
setUpdater 60
setWidth 91
setX 35, 80, 91
setY 35, 81, 91
showRaccoon 44
size 86
sleep 63
SpawnField 18
Sprite 71
sprite 65
SPRITE_BLUE 14, 22
SPRITE_GREEN 14, 22
SPRITE_NORMAL 14
SPRITE_RED 14, 22
SPRITE_YELLOW 14, 22
startId 20, 26, 28, 30

T

TEButton 75
TECollection 78
TECollectionVertical 80
TELabel 82
TESwitch 84
TETextField 87, 88
TEView 90
textColor 82, 87
Timer 63
title 44, 52
type 38

U

unload 45, 46, 48, 49, 51, 52, 54
unlock 37
update 4, 7, 10, 12, 35, 40, 45, 46, 48, 49, 52, 54, 68, 76, 79, 85,
88, 92, 101
updateBoardPosition 11, 20, 26, 28, 30

V

verticalAlignCenter 81, 92

verticalAlignRight 81, 92

W

width 65

winner 39

X

x 65

Y

y 65

YELLOW_PLAYER 38

YellowFigure 30