Table Of Content

Package com.asdf.ssjava

Class Summary

AudioPlayer

Audio player class for the application.

HighScores

The high scores manager object.

SSJava

The equivalent of the main class.

Score

Definition of a score.

---

com.asdf.ssjava

Class AudioPlayer

```
java.lang.Object
    |
    +--com.asdf.ssjava.AudioPlayer
```

---

public class **AudioPlayer**
extends java.lang.Object

Audio player class for the application. Manages all application audio. Contains instances and controls.

Author:

Jeremy Brown

Fields

bulletImpactSound

`public static com.badlogic.gdx.audio.Sound` **`bulletImpactSound`**

The sound to be played when a bullet collides with an entity

---

creatorMusic

`public static com.badlogic.gdx.audio.Music` **`creatorMusic`**

The music to be played in the level creator

---

enemyDeathSound

`public static com.badlogic.gdx.audio.Sound` **`enemyDeathSound`**

The sound to be played when an enemy dies

---

gameMusic

`public static com.badlogic.gdx.audio.Music` **`gameMusic`**
The music to be played during gameplay

---

healthUpSound

`public static com.badlogic.gdx.audio.Sound` **`healthUpSound`**
The sound to be played when the ship collects the health up powerup

---

levelCompleteSound

`public static com.badlogic.gdx.audio.Sound` **`levelCompleteSound`**
The sound to be played at the end of a level

---

levelStartSound

`public static com.badlogic.gdx.audio.Sound` **`levelStartSound`**
The sound to be played at the start of a level

---

menuMusic

`public static com.badlogic.gdx.audio.Music` **`menuMusic`**
The music to be played in the game menus

---

pointsSound

`public static com.badlogic.gdx.audio.Sound` **`pointsSound`**
The sound to be played when the ship collects points

---

shipDeathSound

`public static com.badlogic.gdx.audio.Sound` **`shipDeathSound`**
The sound to be played when the ship dies

---

shipImpactSound

`public static com.badlogic.gdx.audio.Sound` **`shipImpactSound`**
The sound to be played when the ship collides with an entity

---

shotSound

`public static com.badlogic.gdx.audio.Sound` **`shotSound`**
The sound to be played when a bullet is fired by the player's ship

---

speedOfLightOffSound

`public static com.badlogic.gdx.audio.Sound` **`speedOfLightOffSound`**

The sound to be played when the effect of the speed of light powerup ends

---

speedOfLightOnSound

```
public static com.badlogic.gdx.audio.Sound speedOfLightOnSound
```

The sound to be played when the ship collects the speed of light powerup

Constructors

AudioPlayer

```
private  AudioPlayer()
```

Methods

bulletImpact

```
public static void bulletImpact()
```

Plays the explosion sound for the bullet impact.

---

dispose

```
public static void dispose()
```

Disposes all the sound and music instances.

---

enemyDeath

```
public static void enemyDeath()
```

Plays the enemy death sound.

---

healthUp

```
public static void healthUp()
```

Plays the health up activated sound.

---

levelComplete

```
public static void levelComplete()
```

Plays the level start sound.

---

levelStart

`public static void levelStart()`

Plays the level start sound.

---

pauseCreatorMusic

`public static void pauseCreatorMusic()`

Pauses the creator music

---

pauseGameMusic

`public static void pauseGameMusic()`

Pauses the game music.

---

pauseGameSounds

`public static void pauseGameSounds()`

Pauses all the game sounds.

---

pauseMenuMusic

`public static void pauseMenuMusic()`

Pauses the menu music.

---

playCreatorMusic

`public static void playCreatorMusic(boolean looping)`

Plays the creator music.

Parameters:

looping - true if the creator music should loop

---

playGameMusic

`public static void playGameMusic(boolean looping)`

Plays the game music.

Parameters:

looping - true if the game music should loop; false otherwise

---

playMenuMusic

`public static void playMenuMusic(boolean looping)`

Plays the menu music.

Parameters:

looping - true if the menu music should loop

---

pointsPickedUp

`public static void pointsPickedUp()`

Plays the points picked up sound.

---

resumeGameSounds

`public static void resumeGameSounds()`

Resumes all the game sounds.

---

shipDeath

`public static void shipDeath()`

Plays the ship death sound.

---

shipImpact

`public static void shipImpact()`

Plays the explosion sound for the ship impact.

---

shoot

`public static void shoot()`

Plays the shot sound for a fired bullet.

---

speedOfLightOff

`public static void speedOfLightOff()`

Plays the speed of light deactivated sound.

---

speedOfLightOn

`public static void` **`speedOfLightOn`**`()`

Plays the speed of light activated.

---

stopCreatorMusic

`public static void` **`stopCreatorMusic`**`()`

Stops the creator music

---

stopGameMusic

`public static void` **`stopGameMusic`**`()`

Stops the game music.

---

stopGameSounds

`public static void` **`stopGameSounds`**`()`

Stops all the game sounds.

---

stopMenuMusic

`public static void` **`stopMenuMusic`**`()`

Stops the menu music.

---

com.asdf.ssjava

Class HighScores

```
java.lang.Object
    |
    +--com.asdf.ssjava.HighScores
```

---

public class **HighScores**
extends java.lang.Object

The high scores manager object. Is exported & imported with the game.

Author:

Jeremy Brown

Fields

scores

```
private com.badlogic.gdx.utils.Array scores
```

The internal scores Array

Constructors

HighScores

```
public  HighScores()
```

Creates an empty object with an array of size 10

Methods

add

```
public boolean add(com.asdf.ssjava.Score s)
```

Adds a score to the list.

Returns:

true if the score was added

---

exportScores

```
public boolean exportScores()
```

Saves the high scores entries to disk. Important: must be called every time a score is added. Scores are not saved to disk automatically.

Returns:

true if the scores were saved

---

get

```
public com.asdf.ssjava.Score get(int index)
```

Gets the score at the specified index.

Parameters:

index - the index of the desired score

Returns:

the score at the designated index

---

getScoreArray

`public com.badlogic.gdx.utils.Array **getScoreArray**()`

Gets the scores list.

Returns:

the base score array

---

isHighScore

`public boolean **isHighScore**(com.asdf.ssjava.Score s)`

Verifies that the given score makes the high scores list.

Parameters:

s - the score to be checked

Returns:

true if the score will go into the high scores list; false otherwise.

---

remove

`public void **remove**(com.asdf.ssjava.Score s)`

Removes a score from the list.

---

size

`public int **size**()`

Gets the size of the high scores list.

Returns:

the size of the scores array

---

com.asdf.ssjava

Class SSJava

```
java.lang.Object
    |
    +--com.badlogic.gdx.Game
        |
        +--com.asdf.ssjava.SSJava
```

All Implemented Interfaces:

com.badlogic.gdx.ApplicationListener

---

public class **SSJava**
extends com.badlogic.gdx.Game

The equivalent of the main class. Contains game loop and event responding methods.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

DEBUG

`public static final boolean` **`DEBUG`**
Debugging switch

LOG

`public static final java.lang.String` **`LOG`**
Log tag

VERSION

`public static final java.lang.String` **`VERSION`**
Version number

---

assetManager

`public static com.badlogic.gdx.assets.AssetManager` **`assetManager`**
The asset manager instance

---

fileChooserPanel

`public javax.swing.JPanel` **`fileChooserPanel`**
File chooser JPanel instance

---

gameScreen

`public com.asdf.ssjava.screens.GameScreen` **`gameScreen`**
A reference to the game screen

---

height

`public int` **`height`**
The window height, in pixels

---

highScores

`public com.asdf.ssjava.HighScores` **`highScores`**
The high scores instance

highScoresPath

`static java.lang.String` **`highScoresPath`**
The path to the high scores file

highestCompletedLevelKey

`static java.lang.String` **`highestCompletedLevelKey`**
The preferences key for the highest completed level

prefs

`public static com.badlogic.gdx.Preferences` **`prefs`**
The preferences instance

screenshot

`public com.badlogic.gdx.graphics.g2d.TextureRegion` **`screenshot`**
The reference to the most recent screenshot

width

`public int` **`width`**
The window width, in pixels

Constructors

SSJava

`public` **`SSJava`**`()`

Methods

checkLevelCompletion

`public static boolean` **`checkLevelCompletion`**`(int level)`

Checks the preferences to see if the specified level has been completed.

Parameters:

level - the specified level

Returns:

true if the highest completed level is greater than the specified level; false otherwise

create

```
public void create()
```

---

dispose

```
public void dispose()
```

Overrides:

dispose in class com.badlogic.gdx.Game

---

loadHighScores

```
public static com.asdf.ssjava.HighScores loadHighScores(java.lang.String
filePath)
```

Loads the high scores.

Parameters:

filePath - the path where the high scores file is located

Returns:

the HighScores object containing all the high scores

---

pause

```
public void pause()
```

Overrides:

pause in class com.badlogic.gdx.Game

---

render

```
public void render()
```

Overrides:

render in class com.badlogic.gdx.Game

---

resize

```
public void resize(int width,
                   int height)
```

Overrides:

resize in class com.badlogic.gdx.Game

---

resume

```
public void resume()
```

Overrides:

resume in class com.badlogic.gdx.Game

---

writeCompletedLevel

```
public static void writeCompletedLevel(int level)
```

Saves the highest level completed by the player.

Parameters:

level - the level completed

---

com.asdf.ssjava

Class Score

```
java.lang.Object
    |
    +--com.asdf.ssjava.Score
```

All Implemented Interfaces:

java.lang.Comparable

---

public class **Score**
extends java.lang.Object
implements java.lang.Comparable

Definition of a score.

Author:

Jeremy Brown

Fields

name

```
java.lang.String name
```

The player's name

---

score

```
int score
```

The player's score

Constructors

Score

```
public  Score()
```

Constructor for de-serialization. All fields are set to default values. Serialized values can then be loaded over them.

Score

```
public  Score(java.lang.String name,
              int score)
```

Creates a new score

Parameters:

name - the name of the player who obtained this score
score - the score obtained by the player

Methods

compareTo

```
public int compareTo(com.asdf.ssjava.Score s)
```

---

getName

```
public java.lang.String getName()
```

Gets the name.

Returns:

the name

---

getScore

```
public int getScore()
```

Gets the score.

Returns:

the score

---

setName

```
public void setName(java.lang.String name)
```

Sets the name.

Parameters:

name - the name to set

setScore

`public void **setScore**(int score)`

Sets the score.

Parameters:

score - the score to set

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class java.lang.Object

Package com.asdf.ssjava.entities

Class Summary

AbstractEntity

Base class for an entity.

Asteroid

Asteroid implementation of an obstacle.

Bullet

Entity model for a bullet.

BulletType0

Concrete definition of a bullet to model a certain look and behaviour.

BulletType1

Concrete definition of a bullet to model a certain look and behaviour.

Enemy

Entity model for an enemy.

EnemyType1

Implementation of a first enemy type.

MagneticObject

MoveableEntity

Model of a moveable entity.

Obstacle

Entity model for an obstacle.

Planet

Planet type implementation of a game changer.

Points

Powerup implementation of points.

Powerup

PowerupHealthUp

Health up implementation of a powerup.

PowerupSpeedOfLight

Speed of Light implementation of a powerup.

Ship

Entity model of the ship.

SpaceRock

Space rock implementation of an obstacle.

Sun

The sun implementation of a game changer.

---

com.asdf.ssjava.entities

Class AbstractEntity

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
```

Direct Known Subclasses:
com.asdf.ssjava.entities.MoveableEntity

---

public abstract class **AbstractEntity**
extends java.lang.Object

Base class for an entity. Includes fields for position, dimensions, and hitbox.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

body

`protected transient com.badlogic.gdx.physics.box2d.Body` **`body`**
The Box2D body of this entity

---

box2DWorld

`protected transient com.badlogic.gdx.physics.box2d.World` **`box2DWorld`**
The Box2D world instance

---

gameWorld

`protected transient com.asdf.ssjava.world.GameWorld` **`gameWorld`**
The game world instance

---

health

`protected transient int` **`health`**
The entity's health level One point (integer) is equivalent to half a heart in gameplay

---

height

`protected float` **`height`**
The entity's height in pixels

initialized

```
protected transient boolean initialized
```

Whether or not the entity was initialized (given its intial velocity)

---

lastContactTime

```
public long lastContactTime
```

The timestamp at which the last contact w/ the ship was made

---

position

```
protected com.badlogic.gdx.math.Vector2 position
```

The entity's position (now the center of the entity)

---

rotation

```
protected float rotation
```

The entity's rotation, in degrees

---

visible

```
protected transient boolean visible
```

Whether or not the entity is currently visible

---

width

```
protected float width
```

The entity's width in pixels

Constructors

AbstractEntity

```
protected  AbstractEntity(com.badlogic.gdx.math.Vector2 position,
                          float width,
                          float height,
                          float rotation,
                          com.asdf.ssjava.world.GameWorld gameWorld,
                          com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates an entity with specified parameters

Parameters:

position - the entity's position
width - the entity's width
height - the entity's height
rotation - the entity's rotation in degrees
gameWorld - the GameWorld instance
box2DWorld - the World instance (Box2D for collisions)

Methods

createDef

`public void createDef()`

Creates the body definition for this entity

---

die

`public abstract void die()`

Called when the entity's health is 0 In most implementations, removes the entity and it's display instance from the world

---

getBody

`public com.badlogic.gdx.physics.box2d.Body getBody()`

Gets the Box2D body for this entity

Returns:

the body instance

---

getGameWorld

`public com.asdf.ssjava.world.GameWorld getGameWorld()`

Gets the GameWorld instance associated to this entity

Returns:

gameWorld the associated gameWorld instance

---

getHealth

`public synchronized int getHealth()`

Gets the entity's health

Returns:

the entity's health points

---

getHeight

`public float getHeight()`

Returns the entity's height

Returns:

the height of the entity

getHitScore

```
public abstract int getHitScore()
```

getKillScore

```
public abstract int getKillScore()
```

getPosition

```
public com.badlogic.gdx.math.Vector2 getPosition()
```

Returns the entity's position

Returns:

the position of the entity

getRotation

```
public float getRotation()
```

Gets the entity's rotation in degrees

Returns:

the rotation of the entity in degrees

getWidth

```
public float getWidth()
```

Returns the entity's width

Returns:

the width of the entity

healthChange

```
public synchronized void healthChange(int increment)
```

Modifies the health by the specified increment Increment can be positive or negative Does not allow health to go below zero

Parameters:

increment - the value to add/subtract from the entity's health

initWorlds

```
public void initWorlds(com.asdf.ssjava.world.GameWorld gameWorld,
                       com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Initialize the worlds Adds references to the two game worlds to this object Is called when loading a level to pass references of the non-serialized world objects to the entity

---

initialize

```
public abstract void initialize()
```

Initializes the entity to make it active in the game Should give the entity a default velocity and allow it to shoot

---

isDead

```
public boolean isDead()
```

Checks if the entity is dead

Returns:

true if the entity's health is zero; false otherwise

---

isInitialized

```
public boolean isInitialized()
```

Checks if the entity has been initialized

Returns:

true if the initialized flag is set

---

isVisible

```
public boolean isVisible()
```

Checks if the entity is currently visible

Returns:

true is the entity is visible; false otherwise

---

setHealth

```
public synchronized void setHealth(int health)
```

Sets the entity's health

Parameters:

health - the entity's health to be set

setHeight

`public void setHeight(float height)`

Sets the entity's height

Parameters:

height - the height to set

---

setInitialized

`public void setInitialized(boolean initialized)`

Sets the entity's initialized flag

Parameters:

initialized - the initialized flag to set

---

setPosition

`public void setPosition(com.badlogic.gdx.math.Vector2 position)`

Sets the entity's position

Parameters:

position - the position to set

---

setRotation

`public void setRotation(float rotation)`

Sets the entity's rotation in degrees

Parameters:

rotation -

---

setVisible

`public void setVisible(boolean visible)`

Sets the visibility of the entity

Parameters:

visible - sets whether or not the entity is visible

---

setWidth

`public void **setWidth**(float width)`

Sets the entity's width

Parameters:

width - the width to set

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class java.lang.Object

---

update

`public void **update**()`

Updates the entity Does nothing unless overridden Runs every time the game renders a frame

---

com.asdf.ssjava.entities

Class Asteroid

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Obstacle
                |
                +--com.asdf.ssjava.entities.Asteroid
```

---

public class **Asteroid**
extends com.asdf.ssjava.entities.Obstacle

Asteroid implementation of an obstacle.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

ASTEROID_WEIGHT

`public static final java.math.BigInteger **ASTEROID_WEIGHT**`

Weight mod of an asteroid, in kg

DEFAULT_HEALTH

```
public static final int DEFAULT_HEALTH
```

The entity's starting health

DEFAULT_HEIGHT

```
public static final float DEFAULT_HEIGHT
```

The asteroid's default height, in game coordinates

DEFAULT_ROTATION

```
public static final float DEFAULT_ROTATION
```

The asteroid's default rotation, in degrees

DEFAULT_VELOCITY

```
public static final com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY
```

The asteroid's default velocity

DEFAULT_WIDTH

```
public static final float DEFAULT_WIDTH
```

The asteroid's default width, in game coordinates

HIT_SCORE

```
public static final int HIT_SCORE
```

The score given for hitting this entity

KILL_SCORE

```
public static final int KILL_SCORE
```

The score given for killing this entity

Constructors

Asteroid

```
public Asteroid()
```

Constructor for serialization. Creates an asteroid with default parameters.

Asteroid

```
public  Asteroid(com.badlogic.gdx.math.Vector2 position,
                 float width,
                 float height,
                 float rotation)
```

Constructor for level creator. Creates an asteroid with the specified parameters. Does not initialize the world pointers.

Parameters:

position - the position of the asteroid
width - the width of the asteroid
height - the height of the asteroid
rotation - the rotation of the asteroid, in degrees

Asteroid

```
public  Asteroid(com.badlogic.gdx.math.Vector2 position,
                 float width,
                 float height,
                 float rotation,
                 com.asdf.ssjava.world.GameWorld gameWorld,
                 com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates an Asteroid with the specified parameters.

Parameters:

position - the position of the asteroid
width - the width of the asteroid
height - the height of the asteroid
rotation - the rotation of the asteroid
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.Obstacle

---

getHitScore

`public int **getHitScore**()`

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int **getKillScore**()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

`public void **initialize**()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class com.asdf.ssjava.entities.Obstacle

---

com.asdf.ssjava.entities

Class Bullet

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Bullet
```

Direct Known Subclasses:
com.asdf.ssjava.entities.BulletType0, com.asdf.ssjava.entities.BulletType1

---

public abstract class **Bullet**
extends com.asdf.ssjava.entities.MoveableEntity

Entity model for a bullet. Adds starting velocity which can be in any direction, but in most cases will only horizontal.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_HEALTH

```
public static final int DEFAULT_HEALTH
```

The health of a bullet (1, as it dies as soon as it collides)

DEFAULT_VELOCITY

```
com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY
```

The bullet's default velocity

---

damage

```
int damage
```

The damage incurred when the bullet hits the ship or an enemy

---

shooter

```
com.asdf.ssjava.entities.AbstractEntity shooter
```

The entity that shot this bullet

---

type

```
int type
```

The type of bullet

Constructors

Bullet

```
public  Bullet(com.badlogic.gdx.math.Vector2 position,
               float width,
               float height,
               float rotation,
               com.asdf.ssjava.world.GameWorld gameWorld,
               com.badlogic.gdx.physics.box2d.World box2DWorld,
               com.asdf.ssjava.entities.AbstractEntity shooter)
```

Creates a bullet with the specified parameters

Parameters:

position - the position of the bullet
width - the width of the bullet
height - the height of the bullet
rotation - the rotation of the bullet in degrees

Methods

die

`public void **die**()`

Overrides:

die in class com.asdf.ssjava.entities.AbstractEntity

---

getDamage

`public abstract int **getDamage**()`

Gets the damage value of the bullet.

Returns:

the damage value the bullet deals out to other entities

---

getShooter

`public abstract com.asdf.ssjava.entities.AbstractEntity **getShooter**()`

Gets the shooter of the bullet.

Returns:

the entity that fired the bullet

---

getType

`public abstract int **getType**()`

Gets the type of the bullet

Returns:

the type of the bullet

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class com.asdf.ssjava.entities.MoveableEntity

---

update

```
public void update()
```

Overrides:

update in class com.asdf.ssjava.entities.MoveableEntity

---

com.asdf.ssjava.entities

Class BulletType0

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Bullet
                |
                +--com.asdf.ssjava.entities.BulletType0
```

---

public class **BulletType0**
extends com.asdf.ssjava.entities.Bullet

Concrete definition of a bullet to model a certain look and behaviour. Behaviour is defined here and look in the WorldRenderer class. This bullet will be used by the ship.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_DAMAGE

```
public static final int DEFAULT_DAMAGE
```

The damage dealt to an entity that collides with this bullet

DEFAULT_VELOCITY

```
public static final com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY
```

The bullet's default velocity

HIT_SCORE

```
public static final int HIT_SCORE
```

The score given for hitting this entity

KILL_SCORE

`public static final int` **`KILL_SCORE`**
The score given for killing this entity

TYPE

`public final int` **`TYPE`**
The type of bullet

Constructors

BulletType0

```
public  BulletType0(com.badlogic.gdx.math.Vector2 position,
                    float width,
                    float height,
                    float rotation,
                    com.asdf.ssjava.world.GameWorld gameWorld,
                    com.badlogic.gdx.physics.box2d.World box2DWorld,
                    com.asdf.ssjava.entities.AbstractEntity shooter)
```

Creates a Bullet with the specified parameters

Parameters:

position - the position of the bullet
width - the width of the bullet
height - the height of the bullet
rotation - the rotation of the bullet in degrees
gameWorld - the GameWorld instance
box2DWorld - the World instance
shooter - the entity that shot the bullet

Methods

createDef

`public void` **`createDef`**`()`

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

getDamage

`public int` **`getDamage`**`()`

Overrides:

getDamage in class com.asdf.ssjava.entities.Bullet

---

getHitScore

`public int` **`getHitScore`**`()`

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int` **`getKillScore`**`()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

getShooter

`public com.asdf.ssjava.entities.AbstractEntity` **`getShooter`**`()`

Overrides:

getShooter in class com.asdf.ssjava.entities.Bullet

---

getType

`public int` **`getType`**`()`

Overrides:

getType in class com.asdf.ssjava.entities.Bullet

---

initialize

`public void` **`initialize`**`()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String` **`toString`**`()`

Overrides:

toString in class com.asdf.ssjava.entities.Bullet

---

update

```
public void update()
```

Overrides:

update in class com.asdf.ssjava.entities.Bullet

---

com.asdf.ssjava.entities

Class BulletType1

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Bullet
                |
                +--com.asdf.ssjava.entities.BulletType1
```

---

public class **BulletType1**
extends com.asdf.ssjava.entities.Bullet

Concrete definition of a bullet to model a certain look and behaviour. Behaviour is defined here and look in the WorldRenderer class. This bullet will be used by enemies (type 1).

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_DAMAGE

```
public static final int DEFAULT_DAMAGE
```

The damage dealt to an entity that collides with this bullet

DEFAULT_VELOCITY

```
public static final com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY
```

The bullet's default velocity

HIT_SCORE

```
public static final int HIT_SCORE
```

The score given for hitting this entity

KILL_SCORE

`public static final int` **`KILL_SCORE`**
The score given for killing this entity

TYPE

`public final int` **`TYPE`**
The type of bullet

Constructors

BulletType1

```
public  BulletType1(com.badlogic.gdx.math.Vector2 position,
                    float width,
                    float height,
                    float rotation,
                    com.asdf.ssjava.world.GameWorld gameWorld,
                    com.badlogic.gdx.physics.box2d.World box2DWorld,
                    com.asdf.ssjava.entities.AbstractEntity shooter)
```

Creates a Bullet with the specified parameters

Parameters:

position - the position of the bullet
width - the width of the bullet
height - the height of the bullet
rotation - the rotation of the bullet in degrees
gameWorld - the GameWorld instance
box2DWorld - the World instance
shooter - the entity that shot the bullet

Methods

createDef

`public void` **`createDef`**`()`

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

getDamage

`public int` **`getDamage`**`()`

Overrides:

getDamage in class com.asdf.ssjava.entities.Bullet

---

getHitScore

`public int` **`getHitScore`**`()`

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int` **`getKillScore`**`()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

getShooter

`public com.asdf.ssjava.entities.AbstractEntity` **`getShooter`**`()`

Overrides:

getShooter in class com.asdf.ssjava.entities.Bullet

---

getType

`public int` **`getType`**`()`

Overrides:

getType in class com.asdf.ssjava.entities.Bullet

---

initialize

`public void` **`initialize`**`()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String` **`toString`**`()`

Overrides:

toString in class com.asdf.ssjava.entities.Bullet

---

com.asdf.ssjava.entities

Class Enemy

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Enemy
```

Direct Known Subclasses:
com.asdf.ssjava.entities.EnemyType1

---

public abstract class **Enemy**
extends com.asdf.ssjava.entities.MoveableEntity

Entity model for an enemy.

Author:
Jeremy Brown

Author:
Simon Thompson

Constructors

Enemy

```
public  Enemy(com.badlogic.gdx.math.Vector2 position,
              float width,
              float height,
              float rotation,
              com.asdf.ssjava.world.GameWorld gameWorld,
              com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates an enemy with the specified parameters

Parameters:

position - the position of the enemy
width - the width of the enemy
height - the height of the enemy
rotation - the rotation of the enemy
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.AbstractEntity

---

fire

```
public abstract void fire()
```

Fires a bullet from the enemy ship Shot interval specified by shotCooldown

---

getType

```
public abstract int getType()
```

Get the type of the enemy

Returns:

type the type of the enemy

---

toString

```
public java.lang.String toString()
```

Overrides:

toString in class com.asdf.ssjava.entities.MoveableEntity

---

com.asdf.ssjava.entities

Class EnemyType1

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Enemy
                |
                +--com.asdf.ssjava.entities.EnemyType1
```

---

public class **EnemyType1**
extends com.asdf.ssjava.entities.Enemy

Implementation of a first enemy type.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_HEALTH

```
public static final int DEFAULT_HEALTH
```

The entity's starting health

DEFAULT_HEIGHT

`public static final float` **`DEFAULT_HEIGHT`**
The enemy's default height, in game coordinates

DEFAULT_ROTATION

`public static final float` **`DEFAULT_ROTATION`**
The enemy's default rotation, in degrees

DEFAULT_SHOT_COOLDOWN_MS

`public static final int` **`DEFAULT_SHOT_COOLDOWN_MS`**
The default cooldown, in milliseconds, of the enemy's fire

DEFAULT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2` **`DEFAULT_VELOCITY`**
Default velocity for the Type 1 Enemy

DEFAULT_WIDTH

`public static final float` **`DEFAULT_WIDTH`**
The enemy's default width, in game coordinates

ENEMY_TYPE_1_WEIGHT_MOD

`public static final long` **`ENEMY_TYPE_1_WEIGHT_MOD`**
The enemy's weight modifier

HIT_SCORE

`public static final int` **`HIT_SCORE`**
The score given for hitting this entity

KILL_SCORE

`public static final int` **`KILL_SCORE`**
The score given for killing this entity

---

bulletType

`public static final int` **`bulletType`**
The type of bullets the ship will fire

lastShotTime

```
private transient long lastShotTime
```

The time since the last shot was taken

---

shotCooldown

```
private int shotCooldown
```

The shot cool down time, in milliseconds, of the enemy's fire

---

type

```
public static final int type
```

The enemy's type

Constructors

EnemyType1

```
public EnemyType1()
```

Constructor for serialization. Creates an EnemyType1 with default parameters.

EnemyType1

```
public EnemyType1(com.badlogic.gdx.math.Vector2 position,
                  float width,
                  float height,
                  float rotation)
```

Constructor for level creator. Creates an EnemyType1 with default parameters.

Parameters:

position - the position of the enemy
width - the width of the enemy
height - the height of the enemy
rotation - the rotation of the enemy in degrees

EnemyType1

```
public  EnemyType1(com.badlogic.gdx.math.Vector2 position,
                   float width,
                   float height,
                   float rotation,
                   com.asdf.ssjava.world.GameWorld gameWorld,
                   com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates an Enemy with the specified parameters

Parameters:

position - the position of the enemy
width - the width of the enemy
height - the height of the enemy
rotation - the rotation of the enemy in degrees
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.Enemy

---

fire

```
public void fire()
```

Overrides:

fire in class com.asdf.ssjava.entities.Enemy

---

getHitScore

```
public int getHitScore()
```

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int` **`getKillScore`**`()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

getShotCooldown

`public int` **`getShotCooldown`**`()`

Gets the shot cooldown time

Returns:

the shot cooldown time for the enemy, in milliseconds

---

getType

`public int` **`getType`**`()`

Overrides:

getType in class com.asdf.ssjava.entities.Enemy

---

initialize

`public void` **`initialize`**`()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

setShotCooldown

`public void` **`setShotCooldown`**`(int shotCooldown)`

Sets the shot cooldown time

Parameters:

shotCooldown - the shot cooldown time to set, in milliseconds

---

toString

`public java.lang.String` **`toString`**`()`

Overrides:

toString in class com.asdf.ssjava.entities.Enemy

---

update

`public void update()`

Overrides:

update in class com.asdf.ssjava.entities.MoveableEntity

---

com.asdf.ssjava.entities

Class MagneticObject

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Obstacle
                |
                +--com.asdf.ssjava.entities.MagneticObject
```

---

public class **MagneticObject**
extends com.asdf.ssjava.entities.Obstacle

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_HEALTH

`public static final int DEFAULT_HEALTH`

The object's starting health

DEFAULT_HEIGHT

`public static final float DEFAULT_HEIGHT`

The object's default height, in game coordinates

DEFAULT_ROTATION

`public static final float DEFAULT_ROTATION`

The object's default rotation, in degrees

DEFAULT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY`

The object's default velocity (static)

DEFAULT_WIDTH

```
public static final float DEFAULT_WIDTH
```

The object's default width, in game coordinates

HIT_SCORE

```
public static final int HIT_SCORE
```

The score given for hitting this object

KILL_SCORE

```
public static final int KILL_SCORE
```

The score given for killing this object

MAG_OBJECT_WEIGHT_MOD

```
public static final double MAG_OBJECT_WEIGHT_MOD
```

Weight mod of a magnetic object, in kg

Constructors

MagneticObject

```
public MagneticObject()
```

Constructor for serialization. Creates a magnetic object with default parameters.

MagneticObject

```
public MagneticObject(com.badlogic.gdx.math.Vector2 position,
                      float width,
                      float height,
                      float rotation)
```

Constructor for level creator. Creates a magnetic object with the specified parameters. Does not initialize the world pointers.

Parameters:

position - the object's position
width - the object's width
height - the object's height
rotation - the object's rotation in degrees

MagneticObject

```
public  MagneticObject(com.badlogic.gdx.math.Vector2 position,
                       float width,
                       float height,
                       float rotation,
                       com.asdf.ssjava.world.GameWorld gameWorld,
                       com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a magnetic object with the specified parameters

Parameters:

position - the objects's position
width - the objects's width
height - the objects's height
rotation - the objects's rotation in degrees
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

getHitScore

```
public int getHitScore()
```

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

```
public int getKillScore()
```

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

```
public void initialize()
```

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

```
public java.lang.String toString()
```

Overrides:

toString in class com.asdf.ssjava.entities.Obstacle

---

com.asdf.ssjava.entities

Class MoveableEntity

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
```

Direct Known Subclasses:

com.asdf.ssjava.entities.Bullet, com.asdf.ssjava.entities.Enemy, com.asdf.ssjava.entities.Obstacle, com.asdf.ssjava.entities.Powerup, com.asdf.ssjava.entities.Ship

---

public abstract class **MoveableEntity**
extends com.asdf.ssjava.entities.AbstractEntity

Model of a moveable entity. Adds fields for velocity and acceleration vectors, as well as default acceleration and rotation. Implements update method for the entity's hitbox.

Author:

Jeremy Brown

Author:

Simon Thompson

Constructors

MoveableEntity

```
protected  MoveableEntity(com.badlogic.gdx.math.Vector2 position,
                          float width,
                          float height,
                          float rotation,
                          com.asdf.ssjava.world.GameWorld gameWorld,
                          com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates an entity with the specified parameters

Parameters:

position - the entity's position
width - the entity's width
height - the entity's height
rotation - the entity's rotation in degrees
gameWorld - the GameWorld instance
box2DWorld - the World instance (Box2D for collisions)

Methods

toString

`public java.lang.String **toString**()`

Overrides:

toString in class com.asdf.ssjava.entities.AbstractEntity

---

update

`public void **update**()`

Overrides:

update in class com.asdf.ssjava.entities.AbstractEntity

---

com.asdf.ssjava.entities

Class Obstacle

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Obstacle
```

Direct Known Subclasses:

com.asdf.ssjava.entities.Asteroid, com.asdf.ssjava.entities.MagneticObject, com.asdf.ssjava.entities.Planet, com.asdf.ssjava.entities.SpaceRock, com.asdf.ssjava.entities.Sun

---

public abstract class **Obstacle**
extends com.asdf.ssjava.entities.MoveableEntity

Entity model for an obstacle.

Author:

Jeremy Brown

Author:

Simon Thompson

Constructors

Obstacle

```
public  Obstacle(com.badlogic.gdx.math.Vector2 position,
                 float width,
                 float height,
                 float rotation,
                 com.asdf.ssjava.world.GameWorld gameWorld,
                 com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates an obstacle with the specified parameters

Parameters:

position -
width -
height -
rotation -
gameWorld -
box2DWorld -

Methods

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.AbstractEntity

---

toString

```
public java.lang.String toString()
```

Overrides:

toString in class com.asdf.ssjava.entities.MoveableEntity

---

com.asdf.ssjava.entities

Class Planet

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Obstacle
                |
                +--com.asdf.ssjava.entities.Planet
```

---

public class **Planet**
extends com.asdf.ssjava.entities.Obstacle

Planet type implementation of a game changer.

Author:
Simon Thompson

Author:
Jeremy Brown

Fields

DEFAULT_HEALTH

`public static final int` **`DEFAULT_HEALTH`**
The entity's starting health

DEFAULT_HEIGHT

`public static final float` **`DEFAULT_HEIGHT`**
The planet's default height, in game coordinates

DEFAULT_ROTATION

`public static final float` **`DEFAULT_ROTATION`**
The planet's default rotation, in degrees

DEFAULT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2` **`DEFAULT_VELOCITY`**
The planet's default velocity (static)

DEFAULT_WIDTH

`public static final float` **`DEFAULT_WIDTH`**
The planet's default width, in game coordinates

GRAVITATIONNAL_CONSTANT

`public static final double` **`GRAVITATIONNAL_CONSTANT`**
The gravitational constant

HIT_SCORE

`public static final int` **`HIT_SCORE`**
The score given for hitting this entity

KILL_SCORE

`public static final int` **`KILL_SCORE`**
The score given for killing this entity

PLANET_RADIUS

```
public static final long PLANET_RADIUS
```

The planet's radius

PLANET_WEIGHT

```
public static final java.math.BigInteger PLANET_WEIGHT
```

The planet's weight in kg

Constructors

Planet

```
public  Planet()
```

Constructor for serialization. Creates a planet with default parameters.

Planet

```
public  Planet(com.badlogic.gdx.math.Vector2 position,
               float width,
               float height,
               float rotation)
```

Constructor for level creator Creates a plane with default parameters.

Parameters:

position - the position of the planet
width - the width of the planet
height - the height of the planet
rotation - the rotation of the planet, in degrees

Planet

```
public  Planet(com.badlogic.gdx.math.Vector2 position,
               float width,
               float height,
               float rotation,
               com.asdf.ssjava.world.GameWorld gameWorld,
               com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a planet with the specified parameters

Parameters:

position - the position of the planet
width - the width of the planet
height - the height of the planet
rotation - the rotation of the planet, in degrees
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

`public void createDef()`

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

die

`public void die()`

Overrides:

die in class com.asdf.ssjava.entities.Obstacle

---

getHitScore

`public int getHitScore()`

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int getKillScore()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

`public void initialize()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String toString()`

Overrides:

toString in class com.asdf.ssjava.entities.Obstacle

com.asdf.ssjava.entities

Class Points

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Powerup
                |
                +--com.asdf.ssjava.entities.Points
```

---

public class **Points**
extends com.asdf.ssjava.entities.Powerup

Powerup implementation of points.

Author:
Simon Thompson

Author:
Jeremy Brown

Fields

DEFAULT_HEALTH

`public static final int` **`DEFAULT_HEALTH`**
The entity's starting health

DEFAULT_HEIGHT

`public static final float` **`DEFAULT_HEIGHT`**
The entity's default height, in game coordinates

DEFAULT_ROTATION

`public static final float` **`DEFAULT_ROTATION`**
The planet's default rotation, in degrees

DEFAULT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2` **`DEFAULT_VELOCITY`**
The entity's default velocity (static)

DEFAULT_WIDTH

`public static final float` **`DEFAULT_WIDTH`**
The entity's default width, in game coordinates

HIT_SCORE

```
public static final int HIT_SCORE
```

The score given for killing this entity

SCORE

```
public static final int SCORE
```

The score given for killing this entity

Constructors

Points

```
public  Points()
```

Constructor for serialization. Creates a points with default parameters.

Points

```
public  Points(com.badlogic.gdx.math.Vector2 position,
               float width,
               float height,
               float rotation)
```

Constructor for level creator. Creates a points with the specified parameters. Does not initialize the world pointers.

Parameters:

position - the position of the points
width - the width of the points
height - the height of the points
rotation - the rotation of the points, in degrees

Points

```
public  Points(com.badlogic.gdx.math.Vector2 position,
               float width,
               float height,
               float rotation,
               com.asdf.ssjava.world.GameWorld gameWorld,
               com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a points with the specified parameters.

Parameters:

position - the position of the points
width - the width of the points
height - the height of the points
rotation - the rotation of the points
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

`public void createDef()`

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

getHitScore

`public int getHitScore()`

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int getKillScore()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

`public void initialize()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String toString()`

Overrides:

toString in class com.asdf.ssjava.entities.Powerup

---

com.asdf.ssjava.entities

Class Powerup

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Powerup
```

Direct Known Subclasses:

com.asdf.ssjava.entities.Points, com.asdf.ssjava.entities.PowerupHealthUp, com.asdf.ssjava.entities.PowerupSpeedOfLight

---

public abstract class **Powerup**
extends com.asdf.ssjava.entities.MoveableEntity

Author:

Jeremy Brown

Author:

Simon Thompson

Constructors

Powerup

```
public  Powerup(com.badlogic.gdx.math.Vector2 position,
                float width,
                float height,
                float rotation,
                com.asdf.ssjava.world.GameWorld gameWorld,
                com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a powerup with the specified parameters

Parameters:

position - the position of the powerup
width - the width of the powerup
height - the height of the powerup
rotation - the rotation of the powerup, in degrees

Methods

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class com.asdf.ssjava.entities.MoveableEntity

---

com.asdf.ssjava.entities

Class PowerupHealthUp

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Powerup
                |
                +--com.asdf.ssjava.entities.PowerupHealthUp
```

---

public class **PowerupHealthUp**
extends com.asdf.ssjava.entities.Powerup

Health up implementation of a powerup.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_HEALTH

`public static final int **DEFAULT_HEALTH**`

The entity's starting health

DEFAULT_HEIGHT

`public static final float **DEFAULT_HEIGHT**`

The powerup's default height, in game coordinates

DEFAULT_ROTATION

`public static final float **DEFAULT_ROTATION**`

The powerup's default rotation, in degrees

DEFAULT_VELOCITY

```
public static final com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY
```
Default velocity for the Health up powerup

DEFAULT_WIDTH

```
public static final float DEFAULT_WIDTH
```
The powerup's default width, in game coordinates

HEALTH_GIVEN

```
public static final int HEALTH_GIVEN
```
The health points given to the ship when it picks up this power up

---

HIT_SCORE

```
public static final int HIT_SCORE
```
The score given for hitting this entity

KILL_SCORE

```
public static final int KILL_SCORE
```
The score given for killing this entity

Constructors

PowerupHealthUp

```
public PowerupHealthUp()
```

Constructor for serialization. Creates a health up powerup with default parameters.

PowerupHealthUp

```
public PowerupHealthUp(com.badlogic.gdx.math.Vector2 position,
                       float width,
                       float height,
                       float rotation)
```

Constructor for level creator. Creates a health up powerup with the specified parameters. Does not initialize the world pointers.

Parameters:

position - the position of the health up powerup
width - the width of the health up powerup
height - the height of the health up powerup
rotation - the rotation of the health up powerup, in degrees

PowerupHealthUp

```
public  PowerupHealthUp(com.badlogic.gdx.math.Vector2 position,
                        float width,
                        float height,
                        float rotation,
                        com.asdf.ssjava.world.GameWorld gameWorld,
                        com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a health up powerup with the specified parameters.

Parameters:

position - the position of the health up powerup
width - the width of the health up powerup
height - the height of the health up powerup
rotation - the rotation of the health up powerup
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.Powerup

---

getHitScore

```
public int getHitScore()
```

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

```
public int getKillScore()
```

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

```
public void initialize()
```

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

```
public java.lang.String toString()
```

Overrides:

toString in class com.asdf.ssjava.entities.Powerup

---

com.asdf.ssjava.entities

Class PowerupSpeedOfLight

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Powerup
                |
                +--com.asdf.ssjava.entities.PowerupSpeedOfLight
```

---

public class **PowerupSpeedOfLight**
extends com.asdf.ssjava.entities.Powerup

Speed of Light implementation of a powerup.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

COOLDOWN_SECONDS

```
public static final float COOLDOWN_SECONDS
```

The time the powerup's effect will last, in seconds

DEFAULT_HEALTH

```
public static final int DEFAULT_HEALTH
```

The entity's starting health

DEFAULT_HEIGHT

`public static final float` **`DEFAULT_HEIGHT`**
The powerup's default height, in game coordinates

DEFAULT_ROTATION

`public static final float` **`DEFAULT_ROTATION`**
The powerup's default rotation, in degrees

DEFAULT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2` **`DEFAULT_VELOCITY`**
Default velocity for the Speed of Light powerup

DEFAULT_WIDTH

`public static final float` **`DEFAULT_WIDTH`**
The powerup's default width, in game coordinates

HIT_SCORE

`public static final int` **`HIT_SCORE`**
The score given for hitting this entity

KILL_SCORE

`public static final int` **`KILL_SCORE`**
The score given for killing this entity

Constructors

PowerupSpeedOfLight

`public` **`PowerupSpeedOfLight`**`()`

Constructor for serialization

PowerupSpeedOfLight

```
public  PowerupSpeedOfLight(com.badlogic.gdx.math.Vector2 position,
                            float width,
                            float height,
                            float rotation)
```

Constructor for level creator. Creates a speed of light powerup with the specified parameters. Does not initialize the world pointers.

Parameters:

position - the position of the speed of light powerup
width - the width of the speed of light powerup
height - the height of the speed of light powerup
rotation - the rotation of the speed of light powerup, in degrees

---

PowerupSpeedOfLight

```
public  PowerupSpeedOfLight(com.badlogic.gdx.math.Vector2 position,
                            float width,
                            float height,
                            float rotation,
                            com.asdf.ssjava.world.GameWorld gameWorld,
                            com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a speed of light powerup with the specified parameters.

Parameters:

position - the position of the speed of light powerup
width - the width of the speed of light powerup
height - the height of the speed of light powerup
rotation - the rotation of the speed of light powerup
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.Powerup

---

getHitScore

`public int **getHitScore**()`

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int **getKillScore**()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

`public void **initialize**()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class com.asdf.ssjava.entities.Powerup

---

com.asdf.ssjava.entities

Class Ship

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Ship
```

---

public class **Ship**
extends com.asdf.ssjava.entities.MoveableEntity

Entity model of the ship.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_ACCELERATION

`public static com.badlogic.gdx.math.Vector2 DEFAULT_ACCELERATION`

The ship's default acceleration The ship does not initially have a horizontal (x) acceleration, as it moves at a constant speed, which varies only from hitting obstacles and enemies. The y acceleration controls how fast the player is able to move the ship up and down.

DEFAULT_HEALTH

`public static final int DEFAULT_HEALTH`

The ship's default starting health

DEFAULT_HEIGHT

`public static final float DEFAULT_HEIGHT`

The ship's default height, in game coordinates

DEFAULT_ROTATION

`public static final float DEFAULT_ROTATION`

The ship's default rotation, in game degrees

DEFAULT_SHOT_COOLDOWN_MS

`public static final int DEFAULT_SHOT_COOLDOWN_MS`

The ship's default shot cooldown, in milliseconds

DEFAULT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY`

The ship's default velocity The ship will slowly return to the x velocity after it has hit another entity. The y velocity also limits the ship's vertical motion, which is controlled by the player. This is not automatically set by the constructor!

DEFAULT_WIDTH

`public static final float DEFAULT_WIDTH`

The ship's default width, in game coordinates

HIT_SCORE

`public static final int HIT_SCORE`

The score given for hitting this entity

KILL_SCORE

`public static final int` **`KILL_SCORE`**
The score given for killing this entity

SHIP_WEIGHT

`public static final java.math.BigInteger` **`SHIP_WEIGHT`**
Weight of the ship, in kg

SPEED_OF_LIGHT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2` **`SPEED_OF_LIGHT_VELOCITY`**
The velocity of the ship when the speed of light powerup is activated

---

bulletType

`public final int` **`bulletType`**
The type of bullets the ship will fire

---

lastShotTime

`private long` **`lastShotTime`**
The timestamp at which the last shot was fired from this ship, in milliseconds

---

maxDownSpeedReached

`public boolean` **`maxDownSpeedReached`**
Indicates whether or not the ship is currently at its maximum downwards speed

---

maxUpSpeedReached

`public boolean` **`maxUpSpeedReached`**
Indicates whether or not the ship is currently at its maximum upwards speed

---

renderer

`com.asdf.ssjava.world.WorldRenderer` **`renderer`**
The WorldRenderer instance

---

shotCooldown

`private int` **`shotCooldown`**
The time allowed between shots from this ship, in milliseconds

---

speedOfLightEnabled

```
private boolean speedOfLightEnabled
```

The ship cannot lose health from collisions as long as this is true

Constructors

Ship

```
public  Ship(com.badlogic.gdx.math.Vector2 position,
             float width,
             float height,
             float rotation,
             com.asdf.ssjava.world.GameWorld gameWorld,
             com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a ship with a position, dimensions and rotation. Also creates body & fixture definitions.

Parameters:

position - the position of the ship
width - the width of the ship
height - the height of the ship
rotation - the rotation of the ship
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

die

```
public void die()
```

Is called when the ship's health reaches zero (or lower)

Overrides:

die in class com.asdf.ssjava.entities.AbstractEntity

---

disableSpeedOfLight

```
public void disableSpeedOfLight()
```

Disables the speed of light mode of the ship.

---

enableSpeedOfLight

`public void` **`enableSpeedOfLight`**`()`

Enables the speed of light mode of the ship.

---

fire

`public void` **`fire`**`()`

Fires a bullet from the ship. Bullet leaves in the horizontal (right-side) direction.

---

getHitScore

`public int` **`getHitScore`**`()`

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

`public int` **`getKillScore`**`()`

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

getShotCooldown

`public int` **`getShotCooldown`**`()`

Gets the shot cooldown time

Returns:

the shot cooldown time for the ship, in milliseconds

---

healthChange

`public synchronized void` **`healthChange`**`(int increment)`

Overrides:

healthChange in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

`public void **initialize**()`

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

isSpeedOfLightEnabled

`public boolean **isSpeedOfLightEnabled**()`

Gets the speed of light flag

Returns:

true if light speed mode flag is set

---

setHealth

`public synchronized void **setHealth**(int health)`

Overrides:

setHealth in class com.asdf.ssjava.entities.AbstractEntity

---

setShotCooldown

`public void **setShotCooldown**(int shotCooldown)`

Sets the shot cooldown time

Parameters:

shotCooldown - the shot cooldown time to set, in milliseconds

---

setSpeedOfLightEnabled

`public void **setSpeedOfLightEnabled**(boolean enabled)`

Sets the speed of light flag

Parameters:

enabled - true if the speed of light flag is to be enabled

---

toString

`public java.lang.String **toString**()`

Overrides:

toString in class com.asdf.ssjava.entities.MoveableEntity

---

update

`public void update()`

Overrides:

update in class com.asdf.ssjava.entities.MoveableEntity

---

com.asdf.ssjava.entities

Class SpaceRock

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Obstacle
                |
                +--com.asdf.ssjava.entities.SpaceRock
```

---

public class **SpaceRock**
extends com.asdf.ssjava.entities.Obstacle

Space rock implementation of an obstacle.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_HEALTH

`public static final int DEFAULT_HEALTH`

The entity's starting health

DEFAULT_HEIGHT

`public static final float DEFAULT_HEIGHT`

The space rock's default height, in game coordinates

DEFAULT_ROTATION

`public static final float DEFAULT_ROTATION`

The space rock's default rotation, in degrees

DEFAULT_VELOCITY

`public static final com.badlogic.gdx.math.Vector2` **`DEFAULT_VELOCITY`**
The space rock's default velocity (yes, it is static)

DEFAULT_WIDTH

`public static final float` **`DEFAULT_WIDTH`**
The space rock's default width, in game coordinates

HIT_SCORE

`public static final int` **`HIT_SCORE`**
The score given for hitting this entity

KILL_SCORE

`public static final int` **`KILL_SCORE`**
The score given for killing this entity

SPACEROCK_WEIGHT

`public static final java.math.BigInteger` **`SPACEROCK_WEIGHT`**
Weight mod of a space rock, in kg

Constructors

SpaceRock

`public` **`SpaceRock`**`()`

Constructor for serialization. Creates a space rock with default parameters.

SpaceRock

```
public SpaceRock(com.badlogic.gdx.math.Vector2 position,
                 float width,
                 float height,
                 float rotation)
```

Constructor for level creator. Creates a space rock with the specified parameters. Does not initialize the world pointers.

Parameters:

position - the position of the space rock
width - the width of the space rock
height - the height of the space rock
rotation - the rotation of the space rock, in degrees

SpaceRock

```
public  SpaceRock(com.badlogic.gdx.math.Vector2 position,
                  float width,
                  float height,
                  float rotation,
                  com.asdf.ssjava.world.GameWorld gameWorld,
                  com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a space rock with the specified parameters.

Parameters:

position - the position of the space rock
width - the width of the space rock
height - the height of the space rock
rotation - the rotation of the space rock
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

die

```
public void die()
```

Overrides:

die in class com.asdf.ssjava.entities.Obstacle

---

getHitScore

```
public int getHitScore()
```

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

```
public int getKillScore()
```

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

```
public void initialize()
```

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

```
public java.lang.String toString()
```

Overrides:

toString in class com.asdf.ssjava.entities.Obstacle

---

com.asdf.ssjava.entities

Class Sun

```
java.lang.Object
    |
    +--com.asdf.ssjava.entities.AbstractEntity
        |
        +--com.asdf.ssjava.entities.MoveableEntity
            |
            +--com.asdf.ssjava.entities.Obstacle
                |
                +--com.asdf.ssjava.entities.Sun
```

---

public class **Sun**
extends com.asdf.ssjava.entities.Obstacle

The sun implementation of a game changer.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_HEALTH

```
public static final int DEFAULT_HEALTH
```

The sun's default health

DEFAULT_HEIGHT

```
public static final float DEFAULT_HEIGHT
```

The sun's default height, in game coordinates

DEFAULT_ROTATION

```
public static final float DEFAULT_ROTATION
```

The sun's default rotation, in degrees

DEFAULT_VELOCITY

```
public static final com.badlogic.gdx.math.Vector2 DEFAULT_VELOCITY
```

The sun's default velocity (static)

DEFAULT_WIDTH

```
public static final float DEFAULT_WIDTH
```

The sun's default width, in game coordinates

KILL_SCORE

```
public static final int KILL_SCORE
```

Kill score for killing the sun

SUN_WEIGHT

```
public static final java.math.BigInteger SUN_WEIGHT
```

The sun's weight, in kg

Constructors

Sun

```
public Sun()
```

Constructor for serialization. Creates a sun with default parameters.

Sun

```
public Sun(com.badlogic.gdx.math.Vector2 position,
           float width,
           float height,
           float rotation)
```

Constructor for level creator. Creates a sun with the specified parameters. Does not initialize the world pointers.

Parameters:

position - the position of the sun
width - the width of the sun
height - the height of the sun
rotation - the rotation of the sun, in degrees

Sun

```
public  Sun(com.badlogic.gdx.math.Vector2 position,
            float width,
            float height,
            float rotation,
            com.asdf.ssjava.world.GameWorld gameWorld,
            com.badlogic.gdx.physics.box2d.World box2DWorld)
```

Creates a sun with the specified parameters.

Parameters:

position - the position of the sun
width - the width of the sun
height - the height of the sun
rotation - the rotation of the sun
gameWorld - the GameWorld instance
box2DWorld - the World instance

Methods

createDef

```
public void createDef()
```

Overrides:

createDef in class com.asdf.ssjava.entities.AbstractEntity

---

getHitScore

```
public int getHitScore()
```

Overrides:

getHitScore in class com.asdf.ssjava.entities.AbstractEntity

---

getKillScore

```
public int getKillScore()
```

Overrides:

getKillScore in class com.asdf.ssjava.entities.AbstractEntity

---

initialize

```
public void initialize()
```

Overrides:

initialize in class com.asdf.ssjava.entities.AbstractEntity

---

toString

`public java.lang.String` **`toString`**`()`

Overrides:

toString in class com.asdf.ssjava.entities.Obstacle

Package com.asdf.ssjava.screens

Class Summary

CreditsMenu

Credits menu containing labels for the different people who worked on the project and resources used in the game.

GameScreen

The screen shown during gameplay.

HighScoresMenu

High scores menu to display the 10 best level scores stored on the current system.

LevelCompletedMenu

Shows the player's score at the end of the level and gives the options to replay, select a level, and exit.

LevelCreatorOptionsMenu

The options menu for the level creator.

LevelCreatorScreen

The level creation screen.

LevelCreatorScreen.LevelCreatorInput

Input manager for level creator.

LevelRetryMenu

Allows the player to retry the failed level, select another level, or exit to the main menu.

LevelSaveMenu

Allows the user to save a level under a specified name.

LevelSelectMenu

Allows the player to choose a level to play.

MainMenu

Main menu of the game.

OptionsMenu

An options menu allowing the player to modify the sound and music volumes.

PauseMenu

Pause menu allowing the player to temporarily stop playing the current level.

SplashScreen

The first screen shown in the application.

com.asdf.ssjava.screens

Class CreditsMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.CreditsMenu
```

All Implemented Interfaces:
com.badlogic.gdx.Screen

---

public class **CreditsMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

Credits menu containing labels for the different people who worked on the project and resources used in the game.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

artistLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`artistLabel`**
The labels

---

artistTitleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`artistTitleLabel`**
The labels

---

audioLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`audioLabel`**
The labels

---

audioTitleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`audioTitleLabel`**
The labels

---

backButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`backButton`**
The buttons

---

bgImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`bgImage`**
The background image

---

coderLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`coderLabel`**
The labels

---

coderTitleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`coderTitleLabel`**
The labels

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

miscLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`miscLabel`**
The labels

---

miscTitleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`miscTitleLabel`**
The labels

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
The labels

---

whiteFont

```
com.badlogic.gdx.graphics.g2d.BitmapFont whiteFont
```

The text font

Constructors

CreditsMenu

```
public  CreditsMenu(com.asdf.ssjava.SSJava game,
                    com.badlogic.gdx.Screen referrer)
```

Creates a credits menu with the specified parameters.

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class GameScreen

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.GameScreen
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **GameScreen**
extends java.lang.Object
implements com.badlogic.gdx.Screen

The screen shown during gameplay. Contains instances of the game, world and renderer. Calls all render and update methods.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

game

`com.asdf.ssjava.SSJava` **`game`**

The game instance

---

gameWorld

`com.asdf.ssjava.world.GameWorld` **`gameWorld`**

The world instance

---

renderer

`com.asdf.ssjava.world.WorldRenderer` **`renderer`**

The renderer instance

Constructors

GameScreen

```
public  GameScreen(com.asdf.ssjava.SSJava game,
                   com.badlogic.gdx.files.FileHandle levelFile)
```

Constructor of the Game Screen which takes

Parameters:

game - The game instance of type SSJava
levelFile - the FileHandle for the level

GameScreen

```
public  GameScreen(com.asdf.ssjava.SSJava game,
                   com.badlogic.gdx.files.FileHandle levelFile,
                   com.asdf.ssjava.screens.LevelCreatorScreen creator)
```

Constructor for testing levels.

Parameters:

game - the SSJava instance
levelFile - the FileHandle for the level
creator - the level creator screen instance for reference to return to

Methods

dispose

```
public void dispose()
```

---

getGameWorld

```
public com.asdf.ssjava.world.GameWorld getGameWorld()
```

Gets the GameWorld instance.

Returns:

the gameWorld

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class HighScoresMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.HighScoresMenu
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **HighScoresMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

High scores menu to display the 10 best level scores stored on the current system.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

backButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`backButton`**
Buttons

---

bgImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`bgImage`**
The background image

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

namesLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`namesLabel`**
The score labels

---

positionsLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`positionsLabel`**
The score labels

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

scoresLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`scoresLabel`**
The score labels

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
The title label

---

whiteFont

```
com.badlogic.gdx.graphics.g2d.BitmapFont whiteFont
```

The font

Constructors

HighScoresMenu

```
public  HighScoresMenu(com.asdf.ssjava.SSJava game,
                       com.badlogic.gdx.Screen referrer)
```

Creates a high scores menu with the specified parameters.

Parameters:

game - the game instance of type SSJava
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class LevelCompletedMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.LevelCompletedMenu
```

All Implemented Interfaces:
com.badlogic.gdx.Screen

---

public class **LevelCompletedMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

Shows the player's score at the end of the level and gives the options to replay, select a level, and exit. Displayed once a level has been completed.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

bgImage

```
com.badlogic.gdx.scenes.scene2d.ui.Image bgImage
```
Background images

---

exitButton

```
com.asdf.ssjava.screens.screenelements.MenuButton exitButton
```
Buttons

---

game

```
com.asdf.ssjava.SSJava game
```
The game instance

---

gameWorld

`com.asdf.ssjava.world.GameWorld` **`gameWorld`**
The gameWorld instance

---

nameField

`com.badlogic.gdx.scenes.scene2d.ui.TextField` **`nameField`**
The name entry field

---

nameLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`nameLabel`**
Display labels

---

newHighScore

`boolean` **`newHighScore`**
Whether or not the new score is a high score

---

opacityImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`opacityImage`**
Background images

---

playerScore

`com.asdf.ssjava.Score` **`playerScore`**
The player's score

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

retryButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`retryButton`**
Buttons

---

scoreLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`scoreLabel`**
Display labels

---

selectLevelButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`selectLevelButton`**
Buttons

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
Display labels

---

whiteFont

`com.badlogic.gdx.graphics.g2d.BitmapFont` **`whiteFont`**
Menu text font

Constructors

LevelCompletedMenu

```
public  LevelCompletedMenu(com.asdf.ssjava.SSJava game,
                           com.badlogic.gdx.Screen referrer)
```

Creates a new level completed menu with the specified parameters.

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

saveScore

```
public void saveScore(com.asdf.ssjava.Score s)
```

Saves the user's score to the high scores object, which also gets saved to disk.

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class LevelCreatorOptionsMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.LevelCreatorOptionsMenu
```

All Implemented Interfaces:
com.badlogic.gdx.Screen

---

public class **LevelCreatorOptionsMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

The options menu for the level creator. Gives the options of returnning to the creator, testing & saving the created level, and exiting to the main menu.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

backButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`backButton`**
Buttons

---

bgImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`bgImage`**
Background images

---

exitButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`exitButton`**
Buttons

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

loadButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`loadButton`**
Buttons

---

opacityImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`opacityImage`**
Background images

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

saveButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`saveButton`**
Buttons

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

testButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`testButton`**
Buttons

---

thisMenu

`com.badlogic.gdx.Screen` **`thisMenu`**
A concrete reference to this screen instance

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
Menu display title

---

whiteFont

`com.badlogic.gdx.graphics.g2d.BitmapFont` **`whiteFont`**
The title font

Constructors

LevelCreatorOptionsMenu

```
public  LevelCreatorOptionsMenu(com.asdf.ssjava.SSJava game,
                                com.badlogic.gdx.Screen referrer)
```

Creates a creator options menu with the specified parameters.

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

getReferrer

```
public com.badlogic.gdx.Screen getReferrer()
```

Gets the referring screen.

Returns:

the referring screen

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class LevelCreatorScreen

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.LevelCreatorScreen
```

All Implemented Interfaces:
com.badlogic.gdx.Screen

---

public class **LevelCreatorScreen**
extends java.lang.Object
implements com.badlogic.gdx.Screen

The level creation screen. Allows the placement of all game entities in the world. Camera moves to place entities throughout the world.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

batch

`com.badlogic.gdx.graphics.g2d.SpriteBatch` **`batch`**
Sprite batch for creator-specific screen elements

---

clickDown

`boolean` **`clickDown`**
True the cursor was clicked on an entity and is still down

---

entityTypes

`com.badlogic.gdx.utils.Array` **`entityTypes`**
A list of entities to cycle through

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

gameWorld

`com.asdf.ssjava.world.GameWorld` **`gameWorld`**
The world instance

---

levelModified

```
private boolean levelModified
```

True if the level has been modified since it was last saved

---

renderer

```
com.asdf.ssjava.world.WorldRenderer renderer
```

The renderer instance

---

selectedEntity

```
public com.asdf.ssjava.entities.AbstractEntity selectedEntity
```

The currently selected entity

Constructors

LevelCreatorScreen

```
public  LevelCreatorScreen(com.asdf.ssjava.SSJava game,
                           com.badlogic.gdx.files.FileHandle levelFile)
```

Creates a level creator screen with the specified parameters.

Parameters:

game - the SSJava instance
levelFile - the FileHandle for the level

Methods

addEntity

```
protected void addEntity(com.asdf.ssjava.entities.AbstractEntity e)
```

Adds a new instance of the passed entity type to the level.

Parameters:

e - the entity type to be added to the level

---

dispose

```
public void dispose()
```

---

getGameWorld

`public com.asdf.ssjava.world.GameWorld **getGameWorld**()`

Gets the GameWorld instance.

Returns:

the gameWorld

---

getNextEntityType

`public com.asdf.ssjava.entities.AbstractEntity **getNextEntityType**()`

Gets the following entity in the array of entity types.

Returns:

the entity following entityToAdd in the entityTypes array

---

getPrevEntityType

`public com.asdf.ssjava.entities.AbstractEntity **getPrevEntityType**()`

Gets the previous entity in the array of entity types.

Returns:

the entity previous to entityToAdd in the entityTypes array

---

hide

`public void **hide**()`

---

isLevelModified

`public boolean **isLevelModified**()`

Checks if the level modified flag is set.

Returns:

true if the level modified flag is set; false otherwise

---

pause

`public void **pause**()`

---

removeEntity

```
public void removeEntity(com.asdf.ssjava.entities.AbstractEntity selectedEntity)
```

Remove the selected entity from the level.

Parameters:

selectedEntity - the entity to be removed

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

setLevelModified

```
public void setLevelModified(boolean levelModified)
```

Sets the level modified flag.

Parameters:

levelModified - the level modified flag to set

---

setSelectedEntity

```
public void setSelectedEntity(com.asdf.ssjava.entities.AbstractEntity e)
```

Sets the currently selected entity.

---

show

```
public void show()
```

---

showCreatorOptions

`public void` **`showCreatorOptions`**`()`

Shows the options screen which allows testing, saving, and loading of a level, and exiting the creator mode.

---

updateLevelEnd

`public void` **`updateLevelEnd`**`(com.asdf.ssjava.entities.AbstractEntity a)`

Sets the level end point. Called adds a new entity past the previous level end point.

Parameters:

a - the entity added, pushing the end point past it

---

com.asdf.ssjava.screens

Class LevelCreatorScreen.LevelCreatorInput

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.LevelCreatorScreen.LevelCreatorInput
```

All Implemented Interfaces:

com.badlogic.gdx.InputProcessor

---

class **LevelCreatorScreen.LevelCreatorInput**
extends java.lang.Object
implements com.badlogic.gdx.InputProcessor

Input manager for level creator. Defines behaviour for the different keys in the level creator. Camera movement requires continuous polling and is implemented in the GameWorld class.

com.asdf.ssjava.world.GameWorld#update()

Author:

Jeremy Brown

Constructors

LevelCreatorScreen.LevelCreatorInput

`LevelCreatorScreen.LevelCreatorInput``()`

Methods

isClickOnEntity

```
public boolean isClickOnEntity(com.asdf.ssjava.entities.AbstractEntity e,
                               float screenX,
                               float screenY)
```

Checks if the mouse click is on a given entity

Parameters:

e - the entity to be checked
screenX - the x coordinate of the mouse pointer
screenY - the y coordinate of the mouse pointer

Returns:

true if the mouse click is on the specified entity; false otherwise

---

keyDown

```
public boolean keyDown(int keycode)
```

---

keyTyped

```
public boolean keyTyped(char character)
```

---

keyUp

```
public boolean keyUp(int keycode)
```

---

mouseMoved

```
public boolean mouseMoved(int screenX,
                          int screenY)
```

---

scrolled

```
public boolean scrolled(int amount)
```

---

touchDown

```
public boolean touchDown(int screenX,
                         int screenY,
                         int pointer,
                         int button)
```

touchDragged

```
public boolean touchDragged(int screenX,
                            int screenY,
                            int pointer)
```

---

touchUp

```
public boolean touchUp(int screenX,
                       int screenY,
                       int pointer,
                       int button)
```

---

com.asdf.ssjava.screens

Class LevelRetryMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.LevelRetryMenu
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **LevelRetryMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

Allows the player to retry the failed level, select another level, or exit to the main menu. Shows the score obtained so far. Is displayed when the player dies partway through the level.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

bgImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`bgImage`**

Background images

---

exitButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`exitButton`**

The buttons

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

gameWorld

`com.asdf.ssjava.world.GameWorld` **`gameWorld`**
The game world instance

---

opacityImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`opacityImage`**
Background images

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

retryButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`retryButton`**
The buttons

---

scoreLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`scoreLabel`**
Display labels

---

selectLevelButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`selectLevelButton`**
The buttons

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

thisMenu

`com.badlogic.gdx.Screen` **`thisMenu`**
A reference to this menu object to pass to the anonymous listener classes

---

titleLabel

```
com.badlogic.gdx.scenes.scene2d.ui.Label titleLabel
```

Display labels

---

whiteFont

```
com.badlogic.gdx.graphics.g2d.BitmapFont whiteFont
```

The text font

Constructors

LevelRetryMenu

```
public  LevelRetryMenu(com.asdf.ssjava.SSJava game,
                       com.badlogic.gdx.Screen referrer)
```

Creates a level retry menu with the specified parameters.

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class LevelSaveMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.LevelSaveMenu
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **LevelSaveMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

Allows the user to save a level under a specified name.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

backButton

```
com.asdf.ssjava.screens.screenelements.MenuButton backButton
```

The Buttons

---

bgImage

```
com.badlogic.gdx.scenes.scene2d.ui.Image bgImage
```

Background images

---

fileNameField

`com.badlogic.gdx.scenes.scene2d.ui.TextField` **`fileNameField`**
The file name text field

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

opacityImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`opacityImage`**
Background images

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

saveButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`saveButton`**
The Buttons

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

textLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`textLabel`**
The labels

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
The labels

---

whiteFont

`com.badlogic.gdx.graphics.g2d.BitmapFont` **`whiteFont`**

Constructors

LevelSaveMenu

```
public  LevelSaveMenu(com.asdf.ssjava.SSJava game,
                      com.badlogic.gdx.Screen referrer)
```

Creates a level save menu with the specified parameters.

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

saveLevel

```
public void saveLevel(java.lang.String name)
```

Save the level with the specified name. Appends the .json extension to the file if it is not already specified.

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class LevelSelectMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.LevelSelectMenu
```

All Implemented Interfaces:
com.badlogic.gdx.Screen

---

public class **LevelSelectMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

Allows the player to choose a level to play. Displays buttons for the tutorial + the 5 in-game levels. Allows for choosing a level from an external file and returning to the main menu.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

backButton

```
com.asdf.ssjava.screens.screenelements.MenuButton backButton
```

The buttons

---

bgImage

```
com.badlogic.gdx.scenes.scene2d.ui.Image bgImage
```

The background image

---

game

```
com.asdf.ssjava.SSJava game
```

The game instance

level1Button

`com.asdf.ssjava.screens.screenelements.LevelSelectButton` **`level1Button`**
The level seleciton buttons

---

level2Button

`com.asdf.ssjava.screens.screenelements.LevelSelectButton` **`level2Button`**
The level seleciton buttons

---

level3Button

`com.asdf.ssjava.screens.screenelements.LevelSelectButton` **`level3Button`**
The level seleciton buttons

---

level4Button

`com.asdf.ssjava.screens.screenelements.LevelSelectButton` **`level4Button`**
The level seleciton buttons

---

level5Button

`com.asdf.ssjava.screens.screenelements.LevelSelectButton` **`level5Button`**
The level seleciton buttons

---

loadButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`loadButton`**
The buttons

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
The title label

---

tutorialButton

```
com.asdf.ssjava.screens.screenelements.LevelSelectButton tutorialButton
```

The level seleciton buttons

---

whiteFont

```
com.badlogic.gdx.graphics.g2d.BitmapFont whiteFont
```

The title font

Constructors

LevelSelectMenu

```
public LevelSelectMenu(com.asdf.ssjava.SSJava game,
                       com.badlogic.gdx.Screen referrer)
```

Creates a level selection menu with the specified parameters.

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class MainMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.MainMenu
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **MainMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

Main menu of the game. Allows the player to play the game, change the options, start the level creator, view the high scores, view the credits, and exit to the desktop.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

bgImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`bgImage`**

The background image

---

creatorButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`creatorButton`**

The buttons

creditsButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`creditsButton`**
The buttons

---

creditsLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`creditsLabel`**
The text labels

---

exitButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`exitButton`**
The buttons

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

highScoresButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`highScoresButton`**
The buttons

---

optionsButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`optionsButton`**
The buttons

---

playButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`playButton`**
The buttons

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

thisMainMenu

`com.badlogic.gdx.Screen` **`thisMainMenu`**
Concrete reference to this menu

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
The text labels

---

whiteFont

`com.badlogic.gdx.graphics.g2d.BitmapFont` **`whiteFont`**
The text font

Constructors

MainMenu

`public` **`MainMenu`**`(com.asdf.ssjava.SSJava game)`

Creates a main menu with the specified parameters.

Parameters:

game - The game instance of type SSJava

Methods

dispose

`public void` **`dispose`**`()`

---

hide

`public void` **`hide`**`()`

---

pause

`public void` **`pause`**`()`

---

render

`public void` **`render`**`(float delta)`

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class OptionsMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.OptionsMenu
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **OptionsMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

An options menu allowing the player to modify the sound and music volumes.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

backButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`backButton`**

The back button

---

bgImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`bgImage`**

Background images

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

musicField

`com.badlogic.gdx.scenes.scene2d.ui.TextField` **`musicField`**
The text fields

---

musicLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`musicLabel`**
The text labels

---

opacityImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`opacityImage`**
Background images

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

soundField

`com.badlogic.gdx.scenes.scene2d.ui.TextField` **`soundField`**
The text fields

---

soundLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`soundLabel`**
The text labels

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

titleLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`titleLabel`**
The title label

---

volumeLabel

```
com.badlogic.gdx.scenes.scene2d.ui.Label volumeLabel
```

The text labels

---

whiteFont

```
com.badlogic.gdx.graphics.g2d.BitmapFont whiteFont
```

The text font

Constructors

OptionsMenu

```
public  OptionsMenu(com.asdf.ssjava.SSJava game,
                    com.badlogic.gdx.Screen referrer)
```

Creates an options menu with the specified parameters.

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

saveVolume

```
public int saveVolume(int newVolume,
                      java.lang.String key)
```

Saves the new volume to the preferences & activates it immediately.

Parameters:

newVolume - the new volume to be saved
key - the key under which to save the new volume

Returns:

the new volume

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class PauseMenu

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.PauseMenu
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **PauseMenu**
extends java.lang.Object
implements com.badlogic.gdx.Screen

Pause menu allowing the player to temporarily stop playing the current level. Allows to continue playing the level, go to the options menu, and exit to the main menu.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

backButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`backButton`**
The buttons

---

bgImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`bgImage`**
Background images

---

exitButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`exitButton`**
The buttons

---

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

opacityImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`opacityImage`**
Background images

---

optionsButton

`com.asdf.ssjava.screens.screenelements.MenuButton` **`optionsButton`**
The buttons

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The screen which to switch to when the back button is clicked

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage instance

---

thisMenu

`com.badlogic.gdx.Screen` **`thisMenu`**
A reference to this menu object to pass to the anonymous listener classes

titleLabel

```
com.badlogic.gdx.scenes.scene2d.ui.Label titleLabel
```

The title label

---

whiteFont

```
com.badlogic.gdx.graphics.g2d.BitmapFont whiteFont
```

The text font

Constructors

PauseMenu

```
public  PauseMenu(com.asdf.ssjava.SSJava game,
                  com.badlogic.gdx.Screen referrer)
```

Creates a pause menu with the specified parameters

Parameters:

game - the SSJava instance
referrer - the referring screen

Methods

dispose

```
public void dispose()
```

---

hide

```
public void hide()
```

---

pause

```
public void pause()
```

---

render

```
public void render(float delta)
```

---

resize

```
public void resize(int width,
                   int height)
```

---

resume

```
public void resume()
```

---

show

```
public void show()
```

---

com.asdf.ssjava.screens

Class SplashScreen

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.SplashScreen
```

All Implemented Interfaces:

com.badlogic.gdx.Screen

---

public class **SplashScreen**
extends java.lang.Object
implements com.badlogic.gdx.Screen

The first screen shown in the application. Displays a big text with the title of the game. Stays until all the game assets have finished loading.

Author:

Jeremy Brown

Fields

assetManager

```
com.badlogic.gdx.assets.AssetManager assetManager
```

---

batch

```
com.badlogic.gdx.graphics.g2d.SpriteBatch batch
```

---

game

```
com.asdf.ssjava.SSJava game
```

splashSprite

`com.badlogic.gdx.graphics.g2d.Sprite` **`splashSprite`**

splashTexture

`com.badlogic.gdx.graphics.Texture` **`splashTexture`**

tweenManager

`aurelienribon.tweenengine.TweenManager` **`tweenManager`**

Constructors

SplashScreen

`public` **`SplashScreen`**`(com.asdf.ssjava.SSJava game)`

Creates a splash screen.

Parameters:

game - the SSJava instance

Methods

dispose

`public void` **`dispose`**`()`

hide

`public void` **`hide`**`()`

loadAssets

`private void` **`loadAssets`**`()`

Loads all the game assets into memory asynchronously.

pause

`public void` **`pause`**`()`

render

```
public void render(float delta)
```

resize

```
public void resize(int width,
                   int height)
```

resume

```
public void resume()
```

show

```
public void show()
```

tweenCompleted

```
private void tweenCompleted()
```

Sets the screen to the main menu when the fade out animation of the splash screen has completed.

Package com.asdf.ssjava.screens.screenelements

Class Summary

BackButton

Specification of the back button.

LevelSelectButton

Level selection button base class.

MenuButton

Definition for a menu button.

Toast

Utility class to display a toast on the screen.

ToastMessage

Simple data structure for a toast message.

---

com.asdf.ssjava.screens.screenelements

Class BackButton

```
java.lang.Object
    |
    +--com.badlogic.gdx.scenes.scene2d.Actor
        |
        +--com.badlogic.gdx.scenes.scene2d.Group
            |
            +--com.badlogic.gdx.scenes.scene2d.ui.WidgetGroup
                |
                +--com.badlogic.gdx.scenes.scene2d.ui.Table
                    |
                    +--com.badlogic.gdx.scenes.scene2d.ui.Button
                        |
                        +--com.badlogic.gdx.scenes.scene2d.ui.TextButton
                            |
                            +--com.asdf.ssjava.screens.screenelements.MenuButton
                                |
                                +--com.asdf.ssjava.screens.screenelements.BackButton
```

All Implemented Interfaces:

com.badlogic.gdx.scenes.scene2d.utils.Cullable, com.badlogic.gdx.scenes.scene2d.utils.Disableable, com.badlogic.gdx.scenes.scene2d.utils.Layout

---

public class **BackButton**
extends com.asdf.ssjava.screens.screenelements.MenuButton

Specification of the back button.

Author:

Jeremy Brown

Fields

game

`com.asdf.ssjava.SSJava` **`game`**
The game instance

---

referrer

`com.badlogic.gdx.Screen` **`referrer`**
The referring screen

Constructors

BackButton

```
public  BackButton(float width,
                   float height,
                   com.asdf.ssjava.SSJava game,
                   com.badlogic.gdx.Screen referrer)
```

Creates a back button with the specified parameters.

Parameters:

width - the width of the button
height - the height of the button
game - the SSJava instance
referrer - the referring screen

Methods

addBackButtonListener

```
public void addBackButtonListener()
```

Adds an action listener to this button. Listener has the default behaviour of setting the screen to the referring screen

---

getReferrer

```
public com.badlogic.gdx.Screen getReferrer()
```

Gets the referring screen.

Returns:

the referrer

---

setReferrer

```
public void setReferrer(com.badlogic.gdx.Screen referrer)
```

Sets the referring screen.

Parameters:

referrer - the referrer to set

com.asdf.ssjava.screens.screenelements

Class LevelSelectButton

```
java.lang.Object
    |
    +--com.badlogic.gdx.scenes.scene2d.Actor
        |
        +--com.badlogic.gdx.scenes.scene2d.Group
            |
            +--com.badlogic.gdx.scenes.scene2d.ui.WidgetGroup
                |
                +--com.badlogic.gdx.scenes.scene2d.ui.Table
                    |
                    +--com.badlogic.gdx.scenes.scene2d.ui.Button
                        |
                        +--com.badlogic.gdx.scenes.scene2d.ui.TextButton
                            |
                            +--com.asdf.ssjava.screens.screenelements.LevelSelectBut
```

All Implemented Interfaces:

com.badlogic.gdx.scenes.scene2d.utils.Cullable, com.badlogic.gdx.scenes.scene2d.utils.Disableable, com.badlogic.gdx.scenes.scene2d.utils.Layout

---

public class **LevelSelectButton**
extends com.badlogic.gdx.scenes.scene2d.ui.TextButton

Level selection button base class.

Author:

Jeremy Brown

Fields

enabled

`boolean` **`enabled`**

Whether or not the button is enabled

---

game

`com.asdf.ssjava.SSJava` **`game`**

The game instance

---

levelFile

`com.badlogic.gdx.files.FileHandle` **`levelFile`**

The path of the level to be loaded when this button is clicked

---

text

`java.lang.String` **`text`**

The display text of this button

Constructors

LevelSelectButton

```
public
LevelSelectButton(com.asdf.ssjava.screens.screenelements.LevelSelectButton
button,
                  com.asdf.ssjava.SSJava game)
```

Creates a level select button based on another button.

Parameters:

button - the button which to copy
game - the SSJava instance

LevelSelectButton

```
public  LevelSelectButton(java.lang.String text,
                          float width,
                          float height,
                          com.asdf.ssjava.SSJava game,
                          com.badlogic.gdx.files.FileHandle levelFile)
```

Creates a level select button with the specified parameters.

Parameters:

text - the text for the button
width - the width of the button
height - the height of the button
game - the SSJava instance
levelFile - the FileHandle of the corresponding level

Methods

addLevelButtonListener

```
public void addLevelButtonListener()
```

Adds an action listener to this button. Listener has the default behaviour of setting the screen to a game screen that loads the specified level

---

setEnabled

```
public void setEnabled(boolean enabled)
```

Sets the button's text according to the levels unlocked. Button is white if enabled, gray if disabled.

---

com.asdf.ssjava.screens.screenelements

Class MenuButton

```
java.lang.Object
    |
    +--com.badlogic.gdx.scenes.scene2d.Actor
        |
        +--com.badlogic.gdx.scenes.scene2d.Group
            |
            +--com.badlogic.gdx.scenes.scene2d.ui.WidgetGroup
                |
                +--com.badlogic.gdx.scenes.scene2d.ui.Table
                    |
                    +--com.badlogic.gdx.scenes.scene2d.ui.Button
                        |
                        +--com.badlogic.gdx.scenes.scene2d.ui.TextButton
                            |
                            +--com.asdf.ssjava.screens.screenelements.MenuButton
```

All Implemented Interfaces:

com.badlogic.gdx.scenes.scene2d.utils.Cullable, com.badlogic.gdx.scenes.scene2d.utils.Disableable, com.badlogic.gdx.scenes.scene2d.utils.Layout

Direct Known Subclasses:

com.asdf.ssjava.screens.screenelements.BackButton

---

public class **MenuButton**
extends com.badlogic.gdx.scenes.scene2d.ui.TextButton

Definition for a menu button. Skin is the default b&w skin for all the menu buttons.

Author:

Jeremy Brown

Constructors

MenuButton

```
public  MenuButton(java.lang.String text,
                   float width,
                   float height,
                   com.asdf.ssjava.SSJava game)
```

Creates a menu button with the pre-defined skin.

Parameters:

text - the text of the button
width - the width of the button
height - the height of the button

---

com.asdf.ssjava.screens.screenelements

Class Toast

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.screenelements.Toast
```

---

public class **Toast**
extends java.lang.Object

Utility class to display a toast on the screen.

Author:

Jeremy Brown

Constructors

Toast

```
private  Toast()
```

Methods

create

```
public static com.badlogic.gdx.scenes.scene2d.Actor
create(com.asdf.ssjava.screens.screenelements.ToastMessage tm)
```

Toasts a message from a ToastMessage object

Parameters:

tm - the ToastMessage specified to be shown on the screen

Returns:

the new window actor to be added to a stage

---

create

```
public static com.badlogic.gdx.scenes.scene2d.Actor create(java.lang.String text,
                                                           float time)
```

Toasts a message from the specified text and duration.

Parameters:

text - the message text
time - the duration to display the toast

Returns:

the new window actor to be added to a stage

---

com.asdf.ssjava.screens.screenelements

Class ToastMessage

```
java.lang.Object
    |
    +--com.asdf.ssjava.screens.screenelements.ToastMessage
```

---

public class **ToastMessage**
extends java.lang.Object

Simple data structure for a toast message. Used in level intros to guide the player. More potential uses, simple to implement.

Author:

Jeremy Brown

Fields

duration

`public float` **`duration`**

The duration of the message, in seconds

---

message

`public java.lang.String` **`message`**

The message to be shown

---

progress

`public float` **`progress`**

The level progress at which to show the message

Constructors

ToastMessage

`public` **`ToastMessage`**`()`

Constructor for de-serialization. All is null but gets assigned at de-serialization.

ToastMessage

```
public ToastMessage(java.lang.String message,
                    float duration,
                    float progress)
```

Creates a toast message with the specified parameters.

Parameters:

message - the message string
progress - the point in the level at which to display the message
duration - the time for which the message should stay on screen

Package com.asdf.ssjava.tweenaccessors

Class Summary

SpriteTween

Tweet accessor for the splash screen sprite

---

com.asdf.ssjava.tweenaccessors

Class SpriteTween

```
java.lang.Object
    |
    +--com.asdf.ssjava.tweenaccessors.SpriteTween
```

All Implemented Interfaces:

aurelienribon.tweenengine.TweenAccessor

---

public class **SpriteTween**
extends java.lang.Object
implements aurelienribon.tweenengine.TweenAccessor

Tweet accessor for the splash screen sprite

Author:

Jeremy Brown

Fields

ALPHA

```
public static final int ALPHA
```

The alpha value for the tween

Constructors

SpriteTween

```
public  SpriteTween()
```

Methods

getValues

```
public int getValues(com.badlogic.gdx.graphics.g2d.Sprite target,
                     int tweenType,
                     float[] returnValues)
```

---

setValues

```
public void setValues(com.badlogic.gdx.graphics.g2d.Sprite target,
                      int tweenType,
                      float[] newValues)
```

Package com.asdf.ssjava.world

Class Summary

GameCollisionListener

Implements what to do in the event of collisions between fixtures in the game.

GameInputManager

Implements InputProcessor to catch all the player's interactions with the game.

GameWorld

Base world class containing all entities present in the world at the current time.

Level

The level object containing all data for a specific level.

ScoreKeeper

Manages the score for the current level.

WorldRenderer

Manages the rendering and drawing dependencies of all entities present in the associated GameWorld instance.

---

com.asdf.ssjava.world

Class GameCollisionListener

```
java.lang.Object
    |
    +--com.asdf.ssjava.world.GameCollisionListener
```

All Implemented Interfaces:

com.badlogic.gdx.physics.box2d.ContactListener

---

public class **GameCollisionListener**
extends java.lang.Object
implements com.badlogic.gdx.physics.box2d.ContactListener

Implements what to do in the event of collisions between fixtures in the game.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

gameWorld

`com.asdf.ssjava.world.GameWorld` **`gameWorld`**

The world instance

Constructors

GameCollisionListener

```
public  GameCollisionListener(com.asdf.ssjava.world.GameWorld gameWorld)
```

Creates a new instance.

Parameters:

gameWorld - the GameWorld instance

Methods

beginContact

```
public void beginContact(com.badlogic.gdx.physics.box2d.Contact contact)
```

---

bulletImpact

```
public void bulletImpact(com.asdf.ssjava.entities.Bullet b,
                         com.asdf.ssjava.entities.AbstractEntity e)
```

Removes the bullet and deals damage to the entity. Called when a bullet collides with an entity.

Parameters:

b - the bullet that collided
e - the entity that collided

---

endContact

```
public void endContact(com.badlogic.gdx.physics.box2d.Contact contact)
```

---

gravityActivate

```
public void gravityActivate(com.asdf.ssjava.entities.Planet b,
                            com.asdf.ssjava.entities.Ship s)
```

Called when the ship enters a planet's gravitational pull Sets the gravityActivated boolean variable in GameWorld to "true" Sends the instance of Planet over to GameWorld

Parameters:

b - the planet instance
s - the ship

---

gravityDeactivate

```
public void gravityDeactivate()
```

Un-sets the gravity activated flag. Called when the ship exits the planet's attraction zone.

---

healthUpActivate

```
public void healthUpActivate(com.asdf.ssjava.entities.PowerupHealthUp p,
                             com.asdf.ssjava.entities.Ship s)
```

Called when the "health up" powerup is picked up. Removes the powerup and gives health to the ship.

Parameters:

p - the powerup collected
s - the ship

---

magnetActivate

```
public void magnetActivate(com.asdf.ssjava.entities.MagneticObject b,
                           com.asdf.ssjava.entities.Ship s)
```

Called when the ship enters a magnetic object's magnetic field Sets the magnetActivated boolean variable in GameWorld to "true" Sends the instance of MagneticObject over to GameWorld

Parameters:

b - the magnetic object instance
s - the ship

---

magnetDeactivate

```
public void magnetDeactivate()
```

Un-sets the magnet activated flag. Called when the ship exits the magnet's repulsion zone.

---

pointsCollected

```
public void pointsCollected(com.asdf.ssjava.entities.Points p,
                            com.asdf.ssjava.entities.Ship s)
```

Called when the "points" object is picked up. Removes the object and adds score to the ScoreKeeper.

Parameters:

p - the object picked up
s - the ship

---

postSolve

```
public void postSolve(com.badlogic.gdx.physics.box2d.Contact contact,
                      com.badlogic.gdx.physics.box2d.ContactImpulse impulse)
```

---

preSolve

```
public void preSolve(com.badlogic.gdx.physics.box2d.Contact contact,
                     com.badlogic.gdx.physics.box2d.Manifold oldManifold)
```

speedOfLightActivate

```
public void speedOfLightActivate(com.asdf.ssjava.entities.PowerupSpeedOfLight p,
                                 com.asdf.ssjava.entities.Ship s)
```

Called when the "speed of light" powerup is collected. Removes the powerup and sends the ship into light speed mode. Sets timer to disable speed of light mode after a given interval.

Parameters:

p - the powerup picked up
s - the ship

---

sunActivate

```
public void sunActivate(com.asdf.ssjava.entities.Sun b,
                        com.asdf.ssjava.entities.Ship s)
```

Activates the sun's heat effect on the ship. Called when the ship enters the sun's zone of heat. Sets the sunActivated boolean variable in GameWorld to "true". Sends the instance of Sun over to GameWorld.

Parameters:

b - the sun instance
s - the ship

---

sunDeactivate

```
public void sunDeactivate()
```

Un-sets the sun activated flag. Called when the ship exits the sun's zone of heat.

---

com.asdf.ssjava.world

Class GameInputManager

```
java.lang.Object
    |
    +--com.asdf.ssjava.world.GameInputManager
```

All Implemented Interfaces:
com.badlogic.gdx.InputProcessor

---

public class **GameInputManager**
extends java.lang.Object
implements com.badlogic.gdx.InputProcessor

Implements InputProcessor to catch all the player's interactions with the game. Currently maps ship's up and down movements to the arrow keys, as well as W and S and mouse clicks/screen touches in the top and bottom of the left half of the screen. Firing a bullet is done with the spacebar or clicking/touching anywhere in the right half of the screen.

Author:
Jeremy Brown

Author:

Simon Thompson

Fields

game

```
com.asdf.ssjava.SSJava game
```

The game instance

---

gameWorld

```
com.asdf.ssjava.world.GameWorld gameWorld
```

The world's instance

---

ship

```
com.asdf.ssjava.entities.Ship ship
```

The ship's instance

Constructors

GameInputManager

```
public  GameInputManager(com.asdf.ssjava.SSJava game,
                         com.asdf.ssjava.world.GameWorld gameWorld)
```

Creates an instance.

Parameters:

game -
gameWorld -

Methods

keyDown

```
public boolean keyDown(int keycode)
```

---

keyTyped

```
public boolean keyTyped(char character)
```

---

keyUp

```
public boolean keyUp(int keycode)
```

mouseMoved

```
public boolean mouseMoved(int screenX,
                          int screenY)
```

---

scrolled

```
public boolean scrolled(int amount)
```

---

touchDown

```
public boolean touchDown(int screenX,
                         int screenY,
                         int pointer,
                         int button)
```

---

touchDragged

```
public boolean touchDragged(int screenX,
                            int screenY,
                            int pointer)
```

---

touchUp

```
public boolean touchUp(int screenX,
                       int screenY,
                       int pointer,
                       int button)
```

---

com.asdf.ssjava.world

Class GameWorld

```
java.lang.Object
    |
    +--com.asdf.ssjava.world.GameWorld
```

---

public class **GameWorld**
extends java.lang.Object

Base world class containing all entities present in the world at the current time. Manages updating the entities, the score, loading, beginning and ending the level.

Author:

Jeremy Brown

Author:
Simon Thompson

Fields

CREATOR_TYPE

`public static final int` **`CREATOR_TYPE`**
World creator type constant definition

GAME_TYPE

`public static final int` **`GAME_TYPE`**
World game type constant definition

---

box2DWorld

`public com.badlogic.gdx.physics.box2d.World` **`box2DWorld`**
The Box2D World

---

bullets

`public com.badlogic.gdx.utils.Array` **`bullets`**
Array containing all the bullets present in the level

---

creator

`com.asdf.ssjava.screens.LevelCreatorScreen` **`creator`**
The level creator instance

---

game

`com.asdf.ssjava.SSJava` **`game`**
The SSJava instance

---

gravityActivated

`public boolean` **`gravityActivated`**
If the ship is within range of a planet

---

level

`private com.asdf.ssjava.world.Level` **`level`**
Array containing all the obstacles in the current level

---

levelFile

`com.badlogic.gdx.files.FileHandle` **`levelFile`**
The file containing the current level

---

magnet

`public com.asdf.ssjava.entities.MagneticObject` **`magnet`**
The MagneticObject instance

---

magnetActivated

`public boolean` **`magnetActivated`**
If the ship is within range of a magnetic object

---

manager

`com.badlogic.gdx.InputProcessor` **`manager`**
The InputManager instance

---

planet

`public com.asdf.ssjava.entities.Planet` **`planet`**
The Planet instance

---

playEnded

`private boolean` **`playEnded`**
If the current level has stopped being played (either ship died or end of level reached)

---

progress

`private int` **`progress`**
The ship's current progress in the level

---

renderer

`com.asdf.ssjava.world.WorldRenderer` **`renderer`**
The WorldRenderer instance

---

resetShipXVelocity

`com.badlogic.gdx.utils.Timer.Task` **`resetShipXVelocity`**
Task to restore default speed (after Speed Of Light powerup)

---

scoreKeeper

`com.asdf.ssjava.world.ScoreKeeper` **`scoreKeeper`**

The ScoreKeeper instance

---

ship

```
com.asdf.ssjava.entities.Ship ship
```

The ship instance

---

shipHeatIndicator

```
public double shipHeatIndicator
```

The ship's heat indicator

---

sun

```
public com.asdf.ssjava.entities.Sun sun
```

The Sun instance

---

sunActivated

```
public boolean sunActivated
```

If the ship is within range of the sun

---

time

```
private double time
```

The time in seconds

---

worldType

```
int worldType
```

The world type

Constructors

GameWorld

```
public  GameWorld(com.asdf.ssjava.SSJava game,
                  int worldType,
                  com.badlogic.gdx.files.FileHandle levelFile)
```

Creates a world instance from the specified parameters.

Parameters:

game - the SSJava instance
worldType - the type of this world instance (game or creator)
levelFile - the FileHandle for the file of this level

GameWorld

```
public  GameWorld(com.asdf.ssjava.SSJava game,
                  int worldType,
                  com.badlogic.gdx.files.FileHandle levelFile,
                  com.asdf.ssjava.screens.LevelCreatorScreen creator)
```

Creates a world instance from the specified parameters. Constructor for testing a level from the level creator. Helper constructor that sets the creator and uses the main constructor. Allows for returning to the creator when the play is ended.

Parameters:

game - the SSJava instance
worldType - the type of this world instance (game or creator)
levelFile - the FileHandle for the file of this level
creator - the level creator screen instance (if worldType is CREATOR_TYPE)

Methods

dispose

```
public void dispose()
```

Dispose method for the World.

---

exportLevel

```
public void exportLevel(com.badlogic.gdx.files.FileHandle levelFile)
```

Exports the current level to a file in JSON format.

Parameters:

levelFile - the FileHandle path at which to save the exported level

---

getBullets

```
public com.badlogic.gdx.utils.Array getBullets()
```

Gets the bullets array

Returns:

the bullets array

---

getCreator

```
public com.asdf.ssjava.screens.LevelCreatorScreen getCreator()
```

Gets the associated creator instance.

Returns:

the LevelCreator instance

---

getEnemies

`public com.badlogic.gdx.utils.Array` **`getEnemies`**`()`

Gets the enemies array

Returns:

the enemies array

---

getGameChangers

`public com.badlogic.gdx.utils.Array` **`getGameChangers`**`()`

Gets the game changers array

Returns:

the gameChangers array

---

getLevel

`public com.asdf.ssjava.world.Level` **`getLevel`**`()`

Gets the level object instance.

Returns:

level the game world's level instance

---

getLevelFile

`public com.badlogic.gdx.files.FileHandle` **`getLevelFile`**`()`

Gets the level file FileHandle.

Returns:

the level file FileHandle

---

getMagneticObject

`public com.asdf.ssjava.entities.MagneticObject` **`getMagneticObject`**`()`

Gets the magnetic object currently affecting the ship.

Returns:

the magnetic object affecting the ship

---

getManager

`public com.badlogic.gdx.InputProcessor **getManager**()`

Gets the associated input manager.

Returns:

the associated GameInputManager instance

---

getObstacles

`public com.badlogic.gdx.utils.Array **getObstacles**()`

Gets the obstacles array

Returns:

the obstacles array

---

getPlanet

`public com.asdf.ssjava.entities.Planet **getPlanet**()`

Gets the planet currently affecting the ship.

Returns:

the currently activated planet

---

getPowerups

`public com.badlogic.gdx.utils.Array **getPowerups**()`

Gets the powerups array

Returns:

the powerups array

---

getProgress

`public int **getProgress**()`

Gets the ship's progress in the level.

Returns:

the ship's progress

---

getRenderer

`public com.asdf.ssjava.world.WorldRenderer` **`getRenderer`**`()`

Gets the associated renderer.

Returns:

renderer the associated WorldRenderer instance

---

getScoreKeeper

`public com.asdf.ssjava.world.ScoreKeeper` **`getScoreKeeper`**`()`

Gets the score keeper instance.

Returns:

the ScoreKeeper instance

---

getShip

`public com.asdf.ssjava.entities.Ship` **`getShip`**`()`

Gets the ship instance

Returns:

the ship instance

---

getSun

`public com.asdf.ssjava.entities.Sun` **`getSun`**`()`

Gets the Sun currently affecting the ship.

Returns:

the currently activated Sun

---

getWorldType

`public int` **`getWorldType`**`()`

Gets the world type.

Returns:

the world's type (game or creator)

---

gravityActivate

`public void` **`gravityActivate`**`()`

Causes the ship to be attracted towards the planet.

isLevelComplete

`public boolean **isLevelComplete**()`

Checks whether or not the level is complete.

Returns:

true if the ship has completed the level; false otherwise

---

isPlayEnded

`public boolean **isPlayEnded**()`

Gets the play ended flag.

Returns:

true if the play is ended; false otherwise

---

levelCompleted

`public void **levelCompleted**()`

Changes ship behaviour when the level is completed. Ship slows down until its speed reaches zero and starts spinning. After 2 seconds the next screen is shown.

---

loadLevel

`private void **loadLevel**(com.badlogic.gdx.files.FileHandle levelFile)`

Loads a level from a JSON file into the level instance.

Parameters:

levelFile - the FileHandle of the JSON level file to be loaded

---

magnetActivate

`public void **magnetActivate**()`

Causes the ship to be attracted/repelled from the magnetic object.

---

pauseGame

`public void **pauseGame**()`

Calls the pause screen and stops rendering the game.

---

setLevelFile

`public void **setLevelFile**(com.badlogic.gdx.files.FileHandle levelFile)`

Sets the level file.

Parameters:

levelFile - the levelFile to set

---

setMagneticObject

`public void **setMagneticObject**(com.asdf.ssjava.entities.MagneticObject m)`

Sets the magnetic object currently affecting the ship.

Parameters:

m - the magnetic object to affect the ship

---

setManager

`public void **setManager**(com.badlogic.gdx.InputProcessor manager)`

Sets the InputManager instance

Parameters:

manager - the manager to set

---

setPlanet

`public void **setPlanet**(com.asdf.ssjava.entities.Planet p)`

Sets the planet currently affecting the ship.

Parameters:

p - the planet to affect the ship

---

setPlayEnded

`private void **setPlayEnded**(boolean playEnded)`

Sets the play ended flag.

Parameters:

playEnded - true if the play has ended

---

setProgress

```
private void setProgress(int progress)
```

Sets the ship's progress in the level.

Parameters:

progress - the progress to set

---

setRenderer

```
public void setRenderer(com.asdf.ssjava.world.WorldRenderer renderer)
```

Sets the associated renderer.

Parameters:

renderer - the world renderer to be set

---

setSun

```
public void setSun(com.asdf.ssjava.entities.Sun s)
```

Sets the Sun currently affecting the ship.

Parameters:

s - the sun to affect the ship

---

shipDied

```
public void shipDied()
```

Called when the ship has died

---

sunActivate

```
public void sunActivate()
```

Causes the ship to be burned by the sun.

---

update

```
public void update()
```

Update method run in every iteration of the main loop to update entity position, rotation and behaviour. Also verifies which entities/bodies are dead and removes them.

---

com.asdf.ssjava.world

Class Level

```
java.lang.Object
    |
    +--com.asdf.ssjava.world.Level
```

---

public class **Level**
extends java.lang.Object

The level object containing all data for a specific level.

Author:
Jeremy Brown

Fields

backgroundPath

`private java.lang.String` **`backgroundPath`**
The path to the level's background

---

enemies

`public com.badlogic.gdx.utils.Array` **`enemies`**
ArrayList containing all the enemies in the current level

---

gameChangers

`public com.badlogic.gdx.utils.Array` **`gameChangers`**
ArrayList containing all the powerups in the current level

---

levelCode

`int` **`levelCode`**
The level code for distinguishing different game levels

---

levelEnd

`private float` **`levelEnd`**
The x-coordinate of the level's ending point

---

messages

`public com.badlogic.gdx.utils.Array` **`messages`**
The list of messages for this level

---

nextLevelPath

`public java.lang.String` **`nextLevelPath`**
The path of the level, if this level is an "intro" level

---

obstacles

`public com.badlogic.gdx.utils.Array` **`obstacles`**
Array containing all the obstacles in the current level

---

powerups

`public com.badlogic.gdx.utils.Array` **`powerups`**
ArrayList containing all the powerups in the current level

Constructors

Level

`public` **`Level`**`()`

Default Constructor. Initializes all arrays of entities and messages. Sets default values for the level end point, the background and the level code.

Methods

getBackgroundPath

`public java.lang.String` **`getBackgroundPath`**`()`

Gets the background path.

Returns:

the backgroundPath

---

getLevelCode

`public int` **`getLevelCode`**`()`

Gets the level code.

Returns:

the levelCode

---

getLevelEnd

`public float` **`getLevelEnd`**`()`

Gets the level end point.

Returns:

the level end point, in world coordinates

---

setBackgroundPath

`public void` **`setBackgroundPath`**`(java.lang.String backgroundPath)`

Sets the background path.

Parameters:

backgroundPath - the backgroundPath to set

---

setLevelCode

`public void` **`setLevelCode`**`(int levelCode)`

Sets the level code.

Parameters:

levelCode - the levelCode to set

---

setLevelEnd

`public void` **`setLevelEnd`**`(float levelEnd)`

Sets the level end point.

Parameters:

levelEnd - the levelEnd to set, in world coordinates

---

com.asdf.ssjava.world

Class ScoreKeeper

```
java.lang.Object
    |
    +--com.asdf.ssjava.world.ScoreKeeper
```

---

public class **ScoreKeeper**
extends java.lang.Object

Manages the score for the current level.

Author:

Jeremy Brown

Author:

Simon Thompson

Fields

DEFAULT_SCORE

`public static final int` **`DEFAULT_SCORE`**
The default score for the start of a level

---

score

`private int` **`score`**
The player's current score

Constructors

ScoreKeeper

`public` **`ScoreKeeper`**`()`

Methods

add

`public synchronized void` **`add`**`(int addedScore)`

Adds the specified score to the current score.

Parameters:

addedScore - the points to add to the current score

---

getScore

`public int` **`getScore`**`()`

Gets the current score.

Returns:

the score

---

com.asdf.ssjava.world

Class WorldRenderer

```
java.lang.Object
    |
    +--com.asdf.ssjava.world.WorldRenderer
```

---

public class **WorldRenderer**

extends java.lang.Object

Manages the rendering and drawing dependencies of all entities present in the associated GameWorld instance. Draws the HUD comprising of the score, the player's life points, the ship heat indicator, and the level progress.

Author:
Jeremy Brown

Author:
Simon Thompson

Fields

asteroidImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`asteroidImage`**
Images for the level creator HUD stage

---

asteroidTexture

`com.badlogic.gdx.graphics.Texture` **`asteroidTexture`**
Textures for all game elements

---

batch

`com.badlogic.gdx.graphics.g2d.SpriteBatch` **`batch`**
The sprite batches responsible for drawing all elements in the world

---

bgSprite

`com.badlogic.gdx.graphics.g2d.Sprite` **`bgSprite`**
Background sprite

---

bgTexture

`com.badlogic.gdx.graphics.Texture` **`bgTexture`**
Background texture

---

bulletType0Texture

`com.badlogic.gdx.graphics.Texture` **`bulletType0Texture`**
Textures for all game elements

---

bulletType1Texture

`com.badlogic.gdx.graphics.Texture` **`bulletType1Texture`**
Textures for all game elements

---

bulletType2Texture

`com.badlogic.gdx.graphics.Texture` **`bulletType2Texture`**
Textures for all game elements

---

bulletType3Texture

`com.badlogic.gdx.graphics.Texture` **`bulletType3Texture`**
Textures for all game elements

---

cam

`com.badlogic.gdx.graphics.OrthographicCamera` **`cam`**
The camera

---

debugLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`debugLabel`**
The debug text elements

---

debugRenderer

`com.badlogic.gdx.physics.box2d.Box2DDebugRenderer` **`debugRenderer`**
Shape renderer for debugging

---

emptyHeartImage1

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`emptyHeartImage1`**
Images for the game HUD stage

---

emptyHeartImage2

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`emptyHeartImage2`**
Images for the game HUD stage

---

emptyHeartImage3

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`emptyHeartImage3`**
Images for the game HUD stage

---

emptyHeartImage4

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`emptyHeartImage4`**
Images for the game HUD stage

---

emptyHeartTexture

`com.badlogic.gdx.graphics.Texture` **`emptyHeartTexture`**
Texture for the game HUD stage

---

enemyType1Image

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`enemyType1Image`**
Images for the level creator HUD stage

---

enemyType1Texture

`com.badlogic.gdx.graphics.Texture` **`enemyType1Texture`**
Textures for all game elements

---

enemyType2Texture

`com.badlogic.gdx.graphics.Texture` **`enemyType2Texture`**
Textures for all game elements

---

enemyType3Texture

`com.badlogic.gdx.graphics.Texture` **`enemyType3Texture`**
Textures for all game elements

---

entityToAdd

`private com.asdf.ssjava.entities.AbstractEntity` **`entityToAdd`**
The entity type to add to the level

---

fullHeartImage1

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`fullHeartImage1`**
Images for the game HUD stage

---

fullHeartImage2

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`fullHeartImage2`**
Images for the game HUD stage

---

fullHeartImage3

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`fullHeartImage3`**
Images for the game HUD stage

---

fullHeartImage4

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`fullHeartImage4`**
Images for the game HUD stage

---

fullHeartTexture

`com.badlogic.gdx.graphics.Texture` **`fullHeartTexture`**
Texture for the game HUD stage

---

game

`com.asdf.ssjava.SSJava` **`game`**
The SSJava instance

---

gameWorld

`com.asdf.ssjava.world.GameWorld` **`gameWorld`**
The GameWorld instance

---

halfHeartImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`halfHeartImage`**
Images for the game HUD stage

---

halfHeartTexture

`com.badlogic.gdx.graphics.Texture` **`halfHeartTexture`**
Texture for the game HUD stage

---

healthUpTexture

`com.badlogic.gdx.graphics.Texture` **`healthUpTexture`**
Textures for all game elements

---

height

`float` **`height`**
The display width & height

---

magneticObjectImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`magneticObjectImage`**
Images for the level creator HUD stage

---

magneticObjectTexture

`com.badlogic.gdx.graphics.Texture` **`magneticObjectTexture`**
Textures for all game elements

---

planetImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`planetImage`**
Images for the level creator HUD stage

---

planetTexture

`com.badlogic.gdx.graphics.Texture` **`planetTexture`**
Textures for all game elements

---

pointsImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`pointsImage`**
Images for the level creator HUD stage

---

pointsTexture

`com.badlogic.gdx.graphics.Texture` **`pointsTexture`**
Textures for all game elements

---

powerupHealthUpImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`powerupHealthUpImage`**
Images for the level creator HUD stage

---

powerupHealthUpTexture

`com.badlogic.gdx.graphics.Texture` **`powerupHealthUpTexture`**
Textures for all game elements

---

powerupSpeedOfLightImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`powerupSpeedOfLightImage`**
Images for the level creator HUD stage

---

powerupSpeedOfLightTexture

`com.badlogic.gdx.graphics.Texture` **`powerupSpeedOfLightTexture`**
Textures for all game elements

---

progressLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`progressLabel`**
Labels for the game HUD

---

scoreLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`scoreLabel`**
Labels for the game HUD

---

selectedEntity

`private com.asdf.ssjava.entities.AbstractEntity` **`selectedEntity`**
The currently selected entity

---

ship

`com.asdf.ssjava.entities.Ship` **`ship`**
The ship's instance

---

shipHeatIndicatorLabel

`com.badlogic.gdx.scenes.scene2d.ui.Label` **`shipHeatIndicatorLabel`**
The ship's heat indicator

---

shipTexture

`com.badlogic.gdx.graphics.Texture` **`shipTexture`**
Textures for all game elements

---

spaceRockImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`spaceRockImage`**
Images for the level creator HUD stage

---

spaceRockTexture

`com.badlogic.gdx.graphics.Texture` **`spaceRockTexture`**
Textures for all game elements

---

speedOfLightTexture

`com.badlogic.gdx.graphics.Texture` **`speedOfLightTexture`**
Textures for all game elements

---

sr

`com.badlogic.gdx.graphics.glutils.ShapeRenderer` **`sr`**
Shape renderer for selected entity debugging

---

stage

`com.badlogic.gdx.scenes.scene2d.Stage` **`stage`**
The stage for drawing the score, life hearts and level progress

---

sunImage

`com.badlogic.gdx.scenes.scene2d.ui.Image` **`sunImage`**
Images for the level creator HUD stage

---

sunTexture

`com.badlogic.gdx.graphics.Texture` **`sunTexture`**
Textures for all game elements

---

width

`float` **`width`**
The display width & height

Constructors

WorldRenderer

```
public  WorldRenderer(com.asdf.ssjava.world.GameWorld gameWorld)
```

Creates a renderer for the specified GameWorld instance.

Parameters:

gameWorld - the GameWorle instance to be renderered

Methods

createHUD

```
public void createHUD(int width,
                      int height)
```

Creates the HUD display.

Parameters:

width - the width of the screen
height - the height of the screen

---

dispose

`public void **dispose**()`

Disposes the sprite batch and the debugging renderers.

---

getCamera

`public com.badlogic.gdx.graphics.Camera **getCamera**()`

Gets the renderer's camera.

Returns:

the renderer's camera

---

getEntityToAdd

`public com.asdf.ssjava.entities.AbstractEntity **getEntityToAdd**()`

Gets the entity to add in level creator mode.

Returns:

the entity to add

---

getStage

`public com.badlogic.gdx.scenes.scene2d.Stage **getStage**()`

Gets the renderer's main stage.

Returns:

the stage

---

getTexture

`public com.badlogic.gdx.graphics.Texture`
`**getTexture**(com.asdf.ssjava.entities.AbstractEntity e)`

Gets the texture for the specified entity.

Parameters:

e - the entity for which the texture is required

Returns:

the corresponding texture

---

render

`public void **render**()`

Render loop.

setEntityToAdd

`public void **setEntityToAdd**(com.asdf.ssjava.entities.AbstractEntity entityToAdd)`

Sets the entity to add in level creator mode.

Parameters:

entityToAdd - the new entity to add

---

setSelectedEntity

`public void **setSelectedEntity**(com.asdf.ssjava.entities.AbstractEntity selectedEntity)`

Sets the selected entity in level creator mode.

Parameters:

selectedEntity - the new selected entity

---

updateHUD

`public void **updateHUD**()`

Updates the HUD display. Score, life and progress for the gameplay mode. Entity list, position, and selected entity box for level creator mode.