

# All Classes and Interfaces

[All Classes and Interfaces](#)

[Interfaces](#)

[Classes](#)

[Enum Classes](#)

[Record Classes](#)

## Class

### Description

#### **AnimationWrapper**

Animation wrapper class for our projects animations that get used.

#### **Application**

The entry point of the application.

#### **DailyRewardController**

Controller for the daily reward popup screen.

#### **DailyRewardModel**

Model for handling daily reward logic and state.

#### **DailyRewardType**

Enum representing the possible daily reward outcomes.

#### **DefinitionController**

Controller for definition popup

#### **DefinitionPopup**

Represents Definition window called when a word is played

#### **DictionaryDAO**

The SQLite Dictionary.

#### **EmptyTileSlotController**

Empty Tile controller for EmptyTile view.

#### **EmptyTileSlotModel**

An empty tile slot model.

#### **FXMLLoader**

FXML Loader wrapper class for the project to load the game screen FXML pages.

#### **GameMusicPlayer**

Plays the game music track on repeat.

#### **GameScenes**

Game Scenes enum.

#### **GameScreenController**

Game screen controller that has some startup on screen thing todo.

#### **GameScreenController.SidebarLoaded**

#### **GameScreenModel**

Game Screen Model.

#### **GameSoundPlayer**

plays an audio clip as a sound effect for the game.

#### **InfiniteFloatingAnimation**

Balatro style floating tile animation.

#### **LetterTileController**

Controls the layout of the letterUIModel.

#### **LetterTileGroup**

tile group that observes an observable list and updates the ui Tile Controller's nodes into their EmptyTileSlotController nodes.

**LetterTileModel**

Represents the letter tiles the player plays holds, redraws.

**LevelController**

Controller for the level screen.

**LevelModel**

Represents the level model.

**LevelScoreSequence**

Animates each level tile popping up and the score text increasing for every tile in the word row.

**Logger**

com.example.project.services.Logger for Application and to show up in test and build CI workflow.

**LoginController**

Controller for the login scene.

**LoginModel**

Login model class.

**MainMenuController**

controller for the main menu.

**MainMenuModel**

main menu model class

**PageLoader**

Interface for page loading.

**PasswordHasher**

Hashes passwords.

**PopupController<P>**

Parent class for PopupControllers

**PopupLoader**

class to load the popup and initialize it with a controller

**PopupModel**

Parent class for popup windows

**RootLayoutController**

Root layout that remains the root scene of the application to switch scenes within.

**SceneManager**

Represents the scene manager class.

**ScoreChimePlayer**

plays a note from the F major scale ascending in pitch each time for a nice score combo sound.

**ScoreTimeline**

Class for total score counter animation

**ScrabbleTileProvider**

Static class to store the scrabble base letter value.

**ScrabbleTileProvider.ScrabbleTileData****Session**

Game Session.

**ShopController**

Shop view controller.

**ShopModel**

Shop Model.

**SidebarController**

The sidebar on shop and level with the current round info.

### **SQLiteConnection**

Abstract SQLiteConnection class.

### **SQLiteDictionaryConnection**

Connection to the dictionary.db

### **SQLiteUsersConnection**

Connection to the users.db

### **SVGIcons**

a static class to hold some icons for the application.

### **TextEmphasisAnimation**

An animation to quickly enlarge text and then return it to its original scale.

### **TileController<T>**

Parent class of TileControllers.

### **TileControllerFactory**

class to load the tile model controller for that tile model type.

### **TileGroup<modelType,controllerType>**

Tile Group abstract class holds a row or column of tiles.

### **TileModel**

Parent tile class for objects in the scene that are tiles.

### **UpgradeEffects**

upgrade effect methods

### **UpgradeTileController**

Controls the upgrade tile view fxml file.

### **UpgradeTileGroup**

tile group that observes an observable list and updates the ui nodes for some game tiles.

### **UpgradeTileModel**

Represents the upgrade tiles sold at the shop.

### **UpgradeTileModel.UpgradeBuilder**

builder class for the upgrades

### **UpgradeTiles**

Static class to store all the possible upgrade tiles available in the game.

### **User**

Represents a user in the application that can log in.

### **UsersDAO**

SQLite Users database.

## All Packages

Package Summary
Package
Description
<a href="#">com.example.project</a>
<a href="#">com.example.project.controllers</a>
<a href="#">com.example.project.controllers.gameScreens</a>
<a href="#">com.example.project.controllers.gameScreens.animations</a>
<a href="#">com.example.project.controllers.popupControllers</a>
<a href="#">com.example.project.controllers.tiles</a>
<a href="#">com.example.project.models</a>
<a href="#">com.example.project.models.gameScreens</a>
<a href="#">com.example.project.models.popups</a>
<a href="#">com.example.project.models.tileGroups</a>
<a href="#">com.example.project.models.tiles</a>
<a href="#">com.example.project.services</a>
<a href="#">com.example.project.services.shopItems</a>
<a href="#">com.example.project.services.sound</a>
<a href="#">com.example.project.services.sqlite</a>
<a href="#">com.example.project.services.sqlite.dAOs</a>

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Class LoginController

```
java.lang.Object
  com.example.project.controllers.gameScreens.GameScreenController
    com.example.project.controllers.gameScreens.LoginController
```

```
public class LoginController
extends GameScreenController
```

Controller for the login scene.

### Nested Class Summary

**Nested classes/interfaces inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
GameScreenController.SidebarLoaded
```

### Field Summary

**Fields inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
logger
```

### Constructor Summary

#### Constructors

**Constructor**

**Description**

```
LoginController()
```

No arg constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

```
void
```

```
initialize()
```

FXML initialise function called once when the .fxml is loaded on application launch.

```
protected void
```

```
onLoginButtonClick()
```

```
void
```

```
onSceneChangedToThis()
```

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

```
protected void
```

```
onSignupButtonClick()
```

## Methods inherited from class com.example.project.controllers.gameScreens.GameScreenController

loadSidebar

## Methods inherited from class java.lang.Object

clone<sup>✉</sup>, equals<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, toString<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

## Constructor Details

### LoginController

public LoginController()

No arg constructor.

## Method Details

### onSceneChangedToThis

public void onSceneChangedToThis()

#### Description copied from class: GameScreenController

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

#### Specified by:

onSceneChangedToThis in class GameScreenController

### initialize

public void initialize()

FXML initialise function called once when the .fxml is loaded on application launch.

### onLoginButtonClick

protected void onLoginButtonClick()

### onSignupButtonClick

protected void onSignupButtonClick()

## Uses of Package com.example.project.models.tileGroups

### Packages that use com.example.project.models.tileGroups

#### Package

#### Description

[com.example.project.controllers.gameScreens](#)

[com.example.project.models.tileGroups](#)

### Classes in com.example.project.models.tileGroups used by com.example.project.controllers.gameScreens

#### Class

#### Description

[UpgradeTileGroup](#)

tile group that observes an observable list and updates the ui nodes for some game tiles.

### Classes in com.example.project.models.tileGroups used by com.example.project.models.tileGroups

#### Class

#### Description

[TileGroup](#)

Tile Group abstract class holds a row or column of tiles.

**Module** com.example.project  
**Package** com.example.project.models.tiles

## Class EmptyTileSlotModel

java.lang.Object ↗  
  com.example.project.models.tiles.TileModel  
    com.example.project.models.tiles.EmptyTileSlotModel

```
public class EmptyTileSlotModel  
extends TileModel
```

An empty tile slot model. Empty tile slots store a letter tile.

### Field Summary

#### Fields inherited from class com.example.project.models.tiles.TileModel

ClackSoundPlayer, CORNER\_RADIUS, hoverSoundPlayer

### Constructor Summary

#### Constructors

##### Constructor

##### Description

`EmptyTileSlotModel()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

`String` ↗

`getFXMLPath()`

gets the fxml path.

`javafx.beans.property.SimpleObjectProperty<LetterTileModel>`

`getLetterTileSlotProperty()`

`void`

`setTile(LetterTileModel tile)`

Set a new tile.

#### Methods inherited from class com.example.project.models.tiles.TileModel

`getClackSoundPlayer`, `getHoverSoundPlayer`

#### Methods inherited from class java.lang.Object

`clone` ↗, `equals` ↗, `finalize` ↗, `getClass` ↗, `hashCode` ↗, `notify` ↗, `notifyAll` ↗, `toString` ↗, `wait` ↗, `wait` ↗, `wait` ↗

### Constructor Details

#### EmptyTileSlotModel

```
public EmptyTileSlotModel()
```

## Method Details

### getLetterTileSlotProperty

```
public javafx.beans.property.SimpleObjectProperty<LetterTileModel> getLetterTileSlotProperty()
```

**Returns:**

### setTile

```
public void setTile(LetterTileModel tile)
```

Set a new tile.

**Parameters:**

tile - letter tile.

### getFXMLPath

```
public String✉ getFXMLPath()
```

**Description copied from class:** [TileModel](#)

gets the fxml path.

**Specified by:**

[getFXMLPath](#) in class [TileModel](#)

**Returns:**

string path.

**Module** com.example.project  
**Package** com.example.project.models.tiles

## Class LetterTileModel

java.lang.Object  
  com.example.project.models.tiles.TileModel  
    com.example.project.models.tiles.LetterTileModel

```
public class LetterTileModel  
extends TileModel
```

Represents the letter tiles the player plays holds, redraws.

### Field Summary

#### Fields inherited from class com.example.project.models.tiles.TileModel

ClackSoundPlayer, CORNER\_RADIUS, hoverSoundPlayer

### Constructor Summary

#### Constructors

##### Constructor

##### Description

LetterTileModel(Character newLetter)

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

String

getFXMLPath()

gets the fxml path.

Character

getLetter()

Integer

getValue()

#### Methods inherited from class com.example.project.models.tiles.TileModel

getClackSoundPlayer, getHoverSoundPlayer

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

### Constructor Details

#### LetterTileModel

```
public LetterTileModel(Character2 newLetter)
```

**Parameters:**

newLetter - letter

## Method Details

### getLetter

```
public Character2 getLetter()
```

**Returns:**

returns the tiles character.

### getValue

```
public Integer2 getValue()
```

**Returns:**

base value of scrabble letter score.

### getFXMLPath

```
public String2 getFXMLPath()
```

**Description copied from class:** [TileModel](#)

gets the fxml path.

**Specified by:**

[getFXMLPath](#) in class [TileModel](#)

**Returns:**

string path.

**Module** com.example.project

**Package** com.example.project.models.tiles

## Record Class ScrabbleTileProvider.ScrabbleTileData

java.lang.Object<sup>2</sup>  
  java.lang.Record<sup>2</sup>  
    com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData

### Record Components:

value - value of scrabble letter

frequency - population of tile

### Enclosing class:

ScrabbleTileProvider

---

```
public static record ScrabbleTileProvider.ScrabbleTileData(int value, int frequency)
extends Record2
```

## Constructor Summary

### Constructors

#### Constructor

#### Description

**ScrabbleTileData**(int value, int frequency)

Creates an instance of a ScrabbleTileData record class.

## Method Summary

### All Methods

### Instance Methods

### Concrete Methods

#### Modifier and Type

#### Method

#### Description

final boolean

**equals**(Object<sup>2</sup> o)

Indicates whether some other object is "equal to" this one.

int

**frequency**()

Returns the value of the frequency record component.

final int

**hashCode**()

Returns a hash code value for this object.

final String<sup>2</sup>

**toString**()

Returns a string representation of this record class.

int

**value**()

Returns the value of the value record component.

#### Methods inherited from class java.lang.Object<sup>2</sup>

clone<sup>2</sup>, finalize<sup>2</sup>, getClass<sup>2</sup>, notify<sup>2</sup>, notifyAll<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>

## Constructor Details

### ScrabbleTileData

```
public ScrabbleTileData(int value,  
                      int frequency)
```

Creates an instance of a ScrabbleTileData record class.

**Parameters:**

value - the value for the `value` record component

frequency - the value for the `frequency` record component

## Method Details

### toString

```
public final String toString()
```

Returns a string representation of this record class. The representation contains the name of the class, followed by the name and value of each of the record components.

**Specified by:**

`toString` in class `Record`

**Returns:**

a string representation of this object

### hashCode

```
public final int hashCode()
```

Returns a hash code value for this object. The value is derived from the hash code of each of the record components.

**Specified by:**

`hashCode` in class `Record`

**Returns:**

a hash code value for this object

### equals

```
public final boolean equals(Object o)
```

Indicates whether some other object is "equal to" this one. The objects are equal if the other object is of the same class and if all the record components are equal. All components in this record class are compared with '=='.

**Specified by:**

`equals` in class `Record`

**Parameters:**

o - the object with which to compare

**Returns:**

true if this object is the same as the o argument; false otherwise.

### value

```
public int value()
```

Returns the value of the `value` record component.

**Returns:**

the value of the `value` record component

### frequency

```
public int frequency()
```

Returns the value of the `frequency` record component.

**Returns:**

the value of the `frequency` record component

**Module** com.example.project  
**Package** com.example.project.models.tiles

## Class ScrabbleTileProvider

java.lang.Object  
com.example.project.models.tiles.ScrabbleTileProvider

```
public class ScrabbleTileProvider  
extends Object
```

Static class to store the scrabble base letter value.

### Nested Class Summary

#### Nested Classes

##### Modifier and Type

##### Class

##### Description

static final record

ScrabbleTileProvider.ScrabbleTileData

### Constructor Summary

#### Constructors

##### Constructor

##### Description

ScrabbleTileProvider()

Constructor.

### Method Summary

#### All Methods

#### Static Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

Character

drawRandomTile()

int

getValue(Character letter)

Returns Integer of the letter according to scrabble

protected static void

seedRandomNumberGenerator(long newSeed)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

## ScrabbleTileProvider

```
public ScrabbleTileProvider()
```

Constructor.

## Method Details

### seedRandomNumberGenerator

```
protected static void seedRandomNumberGenerator(long newSeed)
```

### getValue

```
public int getValue(Character letter)
```

Returns Integer of the letter according to scrabble

**Parameters:**

letter - letter to get value of

**Returns:**

Integer value of the letter, or 0 if not found

### drawRandomTile

```
public Character drawRandomTile()
```

**Returns:**

randomly selected tile based on CDF

**Module** com.example.project  
**Package** com.example.project.models.tiles

## Class TileModel

java.lang.Object<sup>↳</sup>  
    com.example.project.models.tiles.TileModel

### Direct Known Subclasses:

EmptyTileSlotModel, LetterTileModel, UpgradeTileModel

---

```
public abstract class TileModel
extends Object↳
```

Parent tile class for objects in the scene that are tiles.

## Field Summary

### Fields

#### Modifier and Type

#### Field

#### Description

protected final GameSoundPlayer

ClackSoundPlayer

static final int

CORNER\_RADIUS

corner radius (rounded corners).

protected final GameSoundPlayer

hoverSoundPlayer

## Constructor Summary

### Constructors

#### Constructor

#### Description

TileModel()

## Method Summary

### All Methods

### Instance Methods

### Abstract Methods

### Concrete Methods

#### Modifier and Type

#### Method

#### Description

GameSoundPlayer

getClackSoundPlayer()

gets the sound player that plays the click sound.

abstract String<sup>↳</sup>

getFXMLPath()

gets the fxml path.

## GameSoundPlayer

### getHoverSoundPlayer()

gets the hover sound player

## Methods inherited from class java.lang.Object

clone<sup>✉</sup>, equals<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, toString<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

## Field Details

### hoverSoundPlayer

```
protected final GameSoundPlayer hoverSoundPlayer
```

### ClackSoundPlayer

```
protected final GameSoundPlayer ClackSoundPlayer
```

### CORNER\_RADIUS

```
public static final int CORNER_RADIUS
```

corner radius (rounded corners).

#### See Also:

Constant Field Values

## Constructor Details

### TileModel

```
public TileModel()
```

## Method Details

### getFXMLPath

```
public abstract String✉ getFXMLPath()
```

gets the fxml path.

#### Returns:

string path.

### getHoverSoundPlayer

```
public GameSoundPlayer getHoverSoundPlayer()
```

gets the hover sound player

#### Returns:

sound player

### getClackSoundPlayer

```
public GameSoundPlayer getClackSoundPlayer()
```

gets the sound player that plays the click sound.

**Returns:**

sound player.

**Module** com.example.project  
**Package** com.example.project.models.tiles

## Class UpgradeEffects

java.lang.Object  
com.example.project.models.tiles.UpgradeEffects

```
public class UpgradeEffects  
extends Object
```

upgrade effect methods

### Constructor Summary

#### Constructors

Constructor

Description

`UpgradeEffects()`

### Method Summary

#### All Methods

#### Static Methods

#### Concrete Methods

Modifier and Type

Method

Description

static void

`braceletEffect()`

Add +1 to score multiplier for every consecutive letter alphabetical order.

static void

`buttonEffect()`

adds 5 to the word score for every x in the tile rack.

static void

`coinEffect()`

20% Chance the total word score is doubled

static void

`diceEffect()`

Value is doubled for a random letter in the played word

static void

`glassesEffect()`

Adds +2 to the score multiplier for every identical pair of letters next to each other

static void

`mirrorEffect()`

If played word is a palindrome, doubles word score.

#### Methods inherited from class java.lang.Object

`clone()`, `equals()`, `finalize()`, `getClass()`, `hashCode()`, `notify()`, `notifyAll()`, `toString()`, `wait()`, `wait()`, `wait()`

### Constructor Details

## **UpgradeEffects**

```
public UpgradeEffects()
```

### **Method Details**

#### **glassesEffect**

```
public static void glassesEffect()
```

Adds +2 to the score multiplier for every identical pair of letters next to each other

#### **diceEffect**

```
public static void diceEffect()
```

Value is doubled for a random letter in the played word

#### **coinEffect**

```
public static void coinEffect()
```

20% Chance the total word score is doubled

#### **mirrorEffect**

```
public static void mirrorEffect()
```

If played word is a palindrome, doubles word score.

#### **braceletEffect**

```
public static void braceletEffect()
```

Add +1 to score multiplier for every consecutive letter alphabetical order.

#### **buttonEffect**

```
public static void buttonEffect()
```

adds 5 to the word score for every x in the tile rack.

**Module** com.example.project

**Package** com.example.project.models.tiles

## Class UpgradeTileModel.UpgradeBuilder

java.lang.Object<sup>✉</sup>  
com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder

**Enclosing class:**

UpgradeTileModel

---

```
public static class UpgradeTileModel.UpgradeBuilder  
extends Object✉
```

builder class for the upgrades

### Constructor Summary

#### Constructors

**Constructor**

**Description**

`UpgradeBuilder()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

`UpgradeTileModel`

`build()`

build upgrade

`UpgradeTileModel.UpgradeBuilder`

`cost(double cost)`

upgrade cost builder

`UpgradeTileModel.UpgradeBuilder`

`description(String✉ description)`

upgrade description builder

`UpgradeTileModel.UpgradeBuilder`

`imagePath(String✉ imagePath)`

upgrade image path builder

`UpgradeTileModel.UpgradeBuilder`

`name(String✉ name)`

upgrade name builder

`UpgradeTileModel.UpgradeBuilder`

`upgradeEffect(Runnable✉ upgradeEffect)`

upgrade effect builder

#### Methods inherited from class java.lang.Object<sup>✉</sup>

`clone✉, equals✉, finalize✉, getClass✉, hashCode✉, notify✉, notifyAll✉, toString✉, wait✉, wait✉, wait✉`

## Constructor Details

### UpgradeBuilder

```
public UpgradeBuilder()
```

## Method Details

### description

```
public UpgradeTileModel.UpgradeBuilder description(String2 description)
```

upgrade description builder

**Parameters:**

description - new upgrade description

**Returns:**

description

### name

```
public UpgradeTileModel.UpgradeBuilder name(String2 name)
```

upgrade name builder

**Parameters:**

name - new upgrade name

**Returns:**

name

### imagePath

```
public UpgradeTileModel.UpgradeBuilder imagePath(String2 imagePath)
```

upgrade image path builder

**Parameters:**

imagePath - new upgrade image path

**Returns:**

image path

### cost

```
public UpgradeTileModel.UpgradeBuilder cost(double cost)
```

upgrade cost builder

**Parameters:**

cost - new upgrade cost

**Returns:**

cost

### upgradeEffect

```
public UpgradeTileModel.UpgradeBuilder upgradeEffect(Runnable2 upgradeEffect)
```

upgrade effect builder

**Parameters:**

upgradeEffect - new upgrade effect

**Returns:**

upgrade effect

## build

```
public UpgradeTileModel build()
```

build upgrade

**Returns:**

new upgrade

**Module** com.example.project  
**Package** com.example.project.models.tiles

## Class UpgradeTileModel

java.lang.Object ↗  
    com.example.project.models.tiles.TileModel  
        com.example.project.models.tiles.UpgradeTileModel

```
public class UpgradeTileModel  
extends TileModel
```

Represents the upgrade tiles sold at the shop.

### Nested Class Summary

#### Nested Classes

##### Modifier and Type

##### Class

##### Description

static class

**UpgradeTileModel.UpgradeBuilder**

builder class for the upgrades

### Field Summary

#### Fields inherited from class com.example.project.models.tiles.TileModel

ClackSoundPlayer, CORNER\_RADIUS, hoverSoundPlayer

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**UpgradeTileModel(UpgradeTileModel.UpgradeBuilder builder)**

public upgrade constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

**String** ↗

**getAbilityImagePath()**

gets upgrade tile image path.

**double**

**getCost()**

gets the cost

**String** ↗

**getDescription()**

gets the description.

#### **String**

**getFXMLPath()**

gets the fxml path.

#### **String**

**getName()**

gets the name.

#### **Runnable**

**getUpgradeEffect()**

gets the effect for this tile.

### **Methods inherited from class com.example.project.models.tiles.TileModel**

getClackSoundPlayer, getHoverSoundPlayer

### **Methods inherited from class java.lang.Object**

clone<sup>✉</sup>, equals<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, toString<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

## **Constructor Details**

### **UpgradeTileModel**

**public UpgradeTileModel(UpgradeTileModel.UpgradeBuilder builder)**

public upgrade constructor.

#### **Parameters:**

builder - upgrade builder

## **Method Details**

### **getName**

**public String getName()**

gets the name.

#### **Returns:**

return name.

### **getDescription**

**public String getDescription()**

gets the description.

#### **Returns:**

description.

### **getCost**

**public double getCost()**

gets the cost

#### **Returns:**

return cost.

## **getAbilityImagePath**

```
public String✉ getAbilityImagePath()
```

gets upgrade tile image path.

**Returns:**

returns loaded Image for this tile.

## **getUpgradeEffect**

```
public Runnable✉ getUpgradeEffect()
```

gets the effect for this tile.

**Returns:**

returns the runnable effect for this tile.

## **getFXMLPath**

```
public String✉ getFXMLPath()
```

**Description copied from class: TileModel**

gets the fxml path.

**Specified by:**

getFXMLPath in class TileModel

**Returns:**

string path.

## Uses of Class com.example.project.models.tiles.EmptyTileSlotModel

### Packages that use EmptyTileSlotModel

#### Package

#### Description

com.example.project.controllers.tiles

### Uses of EmptyTileSlotModel in com.example.project.controllers.tiles

#### Methods in com.example.project.controllers.tiles with parameters of type EmptyTileSlotModel

##### Modifier and Type

##### Method

##### Description

void

`EmptyTileSlotController.bind(EmptyTileSlotModel modelToUse)`

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Class MainMenuController

```
java.lang.Object
    com.example.project.controllers.gameScreens.GameScreenController
        com.example.project.controllers.gameScreens.MainMenuController
```

```
public class MainMenuController
extends GameScreenController
```

controller for the main menu.

### Nested Class Summary

**Nested classes/interfaces inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
GameScreenController.SidebarLoaded
```

### Field Summary

**Fields inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
logger
```

### Constructor Summary

#### Constructors

**Constructor**

**Description**

```
MainMenuController()
```

No arg constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

```
void
```

```
initialize()
```

FXML initialise function called once when the .fxml is loaded.

```
protected void
```

```
onDailyRewardClicked()
```

```
protected void
```

```
onFastForwardClick()
```

```
protected void
```

```
onLogoutButtonClick()
```

```
void  
onSceneChangedToThis()
```

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

```
protected void  
onStartButtonClick()
```

## Methods inherited from class com.example.project.controllers.gameScreens.GameScreenController

```
loadSidebar
```

## Methods inherited from class java.lang.Object

```
clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()
```

## Constructor Details

### MainMenuController

```
public MainMenuController()
```

No arg constructor.

## Method Details

### onSceneChangedToThis

```
public void onSceneChangedToThis()
```

#### Description copied from class: GameScreenController

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

#### Specified by:

onSceneChangedToThis in class GameScreenController

### initialize

```
public void initialize()
```

FXML initialise function called once when the .fxml is loaded.

### onStartButtonClick

```
protected void onStartButtonClick()
```

### onLogoutButtonClick

```
protected void onLogoutButtonClick()
```

### onDailyRewardClicked

```
protected void onDailyRewardClicked()
```

### onFastForwardClick

```
protected void onFastForwardClick()
```



## Uses of Class com.example.project.models.tiles.LetterTileModel

### Packages that use LetterTileModel

#### Package

#### Description

com.example.project.controllers.tiles

com.example.project.models.gameScreens

com.example.project.models.tileGroups

com.example.project.models.tiles

### Uses of LetterTileModel in com.example.project.controllers.tiles

#### Methods in com.example.project.controllers.tiles that return LetterTileModel

##### Modifier and Type

##### Method

##### Description

LetterTileModel

LetterTileController.getModel()

#### Methods in com.example.project.controllers.tiles with parameters of type LetterTileModel

##### Modifier and Type

##### Method

##### Description

void

LetterTileController.bind(LetterTileModel tile)

LetterTileController

TileControllerFactory.createLetterTileController(LetterTileModel lt)

### Uses of LetterTileModel in com.example.project.models.gameScreens

#### Methods in com.example.project.models.gameScreens that return types with arguments of type LetterTileModel

##### Modifier and Type

##### Method

##### Description

javafx.beans.property.ReadOnlyListProperty<LetterTileModel>

LevelModel.getRedrawWindowTilesProperty()

javafx.beans.property.ReadOnlyListProperty<LetterTileModel>

LevelModel.getTileRackTilesProperty()

```
javafx.beans.property.ReadOnlyListProperty<LetterTileModel>
```

```
LevelModel.getWordWindowTilesProperty()
```

### Methods in com.example.project.models.gameScreens with parameters of type LetterTileModel

#### Modifier and Type

#### Method

#### Description

protected void

```
LevelModel.addTileToRack(LetterTileModel tile)
```

protected void

```
LevelModel.addTileToRedrawWindow(LetterTileModel tile)
```

protected void

```
LevelModel.addTileToWordWindow(LetterTileModel tile)
```

void

```
LevelModel.addToCombo(LetterTileModel tile)
```

add combo sum and multiCombo

boolean

```
LevelModel.tryMoveTile(LetterTileModel tile)
```

determines where tile should go and moves it

## Uses of LetterTileModel in com.example.project.models.tileGroups

### Constructor parameters in com.example.project.models.tileGroups with type arguments of type LetterTileModel

#### Modifier

#### Constructor

#### Description

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<  
<LetterTileController> onClickAction)
```

Constructor

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<  
<LetterTileController> onClickHandler, List<Runnable> afterSyncActions)
```

Constructor

protected

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<  
<LetterTileController> onClickHandler, List<Runnable> afterSyncActions, TileControllerFactory factory)
```

## Uses of LetterTileModel in com.example.project.models.tiles

### Methods in com.example.project.models.tiles that return types with arguments of type LetterTileModel

#### Modifier and Type

**Method****Description**

```
javafx.beans.property.SimpleObjectProperty<LetterTileModel>
```

```
EmptyTileSlotModel.getLetterTileSlotProperty()
```

**Methods in com.example.project.models.tiles with parameters of type LetterTileModel****Modifier and Type****Method****Description**

```
void
```

```
EmptyTileSlotModel.setTile(LetterTileModel tile)
```

Set a new tile.

## Uses of Record Class

**com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData**

No usage of com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData

---

Copyright © 2025. All rights reserved.

## **Uses of Class com.example.project.models.tiles.ScrabbleTileProvider**

No usage of com.example.project.models.tiles.ScrabbleTileProvider

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.models.tiles.TileModel

### Packages that use TileModel

#### Package

#### Description

com.example.project.controllers.tiles

com.example.project.models.tileGroups

com.example.project.models.tiles

### Uses of TileModel in com.example.project.controllers.tiles

#### Classes in com.example.project.controllers.tiles with type parameters of type TileModel

##### Modifier and Type

##### Class

##### Description

class

TileController<T extends TileModel>

Parent class of TileControllers.

#### Fields in com.example.project.controllers.tiles declared as TileModel

##### Modifier and Type

##### Field

##### Description

protected T

TileController.model

#### Methods in com.example.project.controllers.tiles with type parameters of type TileModel

##### Modifier and Type

##### Method

##### Description

<C extends TileController<T>, T extends TileModel>  
C

TileControllerFactory.createTileController(T tile, Class<?> controllerType)

Returns controller of type.

### Uses of TileModel in com.example.project.models.tileGroups

#### Classes in com.example.project.models.tileGroups with type parameters of type TileModel

##### Modifier and Type

##### Class

##### Description

class

TileGroup<modelType extends TileModel, controllerType extends TileController<modelType>>

Tile Group abstract class holds a row or column of tiles.

## Uses of *TileModel* in *com.example.project.models.tiles*

### Subclasses of *TileModel* in *com.example.project.models.tiles*

#### Modifier and Type

##### Class

##### Description

class

#### [EmptyTileSlotModel](#)

An empty tile slot model.

class

#### [LetterTileModel](#)

Represents the letter tiles the player plays holds, redraws.

class

#### [UpgradeTileModel](#)

Represents the upgrade tiles sold at the shop.

## Uses of Class **com.example.project.models.tiles.UpgradeEffects**

No usage of com.example.project.models.tiles.UpgradeEffects

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder

### Packages that use UpgradeTileModel.UpgradeBuilder

#### Package

#### Description

com.example.project.models.tiles

### Uses of UpgradeTileModel.UpgradeBuilder in com.example.project.models.tiles

#### Methods in com.example.project.models.tiles that return UpgradeTileModel.UpgradeBuilder

##### Modifier and Type

##### Method

##### Description

**UpgradeTileModel.UpgradeBuilder**

**UpgradeTileModel.UpgradeBuilder.cost(double cost)**

upgrade cost builder

**UpgradeTileModel.UpgradeBuilder**

**UpgradeTileModel.UpgradeBuilder.description(String<sup>✉</sup> description)**

upgrade description builder

**UpgradeTileModel.UpgradeBuilder**

**UpgradeTileModel.UpgradeBuilder.imagePath(String<sup>✉</sup> imagePath)**

upgrade image path builder

**UpgradeTileModel.UpgradeBuilder**

**UpgradeTileModel.UpgradeBuilder.name(String<sup>✉</sup> name)**

upgrade name builder

**UpgradeTileModel.UpgradeBuilder**

**UpgradeTileModel.UpgradeBuilder.upgradeEffect(Runnable<sup>✉</sup> upgradeEffect)**

upgrade effect builder

#### Constructors in com.example.project.models.tiles with parameters of type UpgradeTileModel.UpgradeBuilder

##### Modifier

##### Constructor

##### Description

**UpgradeTileModel(UpgradeTileModel.UpgradeBuilder builder)**

public upgrade constructor.

## Uses of Class com.example.project.models.tiles.UpgradeTileModel

### Packages that use UpgradeTileModel

#### Package

#### Description

com.example.project.controllers.tiles

com.example.project.models.gameScreens

com.example.project.models.tileGroups

com.example.project.models.tiles

com.example.project.services

com.example.project.services.shopItems

### Uses of UpgradeTileModel in com.example.project.controllers.tiles

#### Methods in com.example.project.controllers.tiles with parameters of type UpgradeTileModel

##### Modifier and Type

##### Method

##### Description

void

`UpgradeTileController.bind(UpgradeTileModel tileModel)`

### Uses of UpgradeTileModel in com.example.project.models.gameScreens

#### Methods in com.example.project.models.gameScreens that return types with arguments of type UpgradeTileModel

##### Modifier and Type

##### Method

##### Description

`javafx.beans.property.ListProperty<UpgradeTileModel>`

`ShopModel.currentShopItemsProperty()`

Current items in the shop row that you can buy.

`javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel>`

`ShopModel.playersUpgradesProperty()`

get what upgrades the player currently has.

#### Methods in com.example.project.models.gameScreens with parameters of type UpgradeTileModel

##### Modifier and Type

##### Method

##### Description

boolean

`ShopModel.canPurchase(UpgradeTileModel tile)`

Determines whether the player can afford to purchase the specified upgrade tile.

void

**ShopModel.purchase(UpgradeTileModel tileClickedOn)**

This should attempt to purchase an upgrade tile from the shop.

## Uses of UpgradeTileModel in com.example.project.models.tileGroups

### Constructor parameters in com.example.project.models.tileGroups with type arguments of type UpgradeTileModel

**Modifier**

**Constructor**

**Description**

**UpgradeTileGroup(javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> observedList)**

Constructor with no additional sync action parameter.

**UpgradeTileGroup(javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> observedList, Consumer<  
<UpgradeTileController>> onClickAction)**

Constructor.

## Uses of UpgradeTileModel in com.example.project.models.tiles

### Methods in com.example.project.models.tiles that return UpgradeTileModel

**Modifier and Type**

**Method**

**Description**

**UpgradeTileModel**

**UpgradeTileModel.UpgradeBuilder.build()**

build upgrade

## Uses of UpgradeTileModel in com.example.project.services

### Methods in com.example.project.services that return types with arguments of type UpgradeTileModel

**Modifier and Type**

**Method**

**Description**

**javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel>**

**Session.getUpgradeTilesProperty()**

gets upgrade tile property

### Methods in com.example.project.services with parameters of type UpgradeTileModel

**Modifier and Type**

**Method**

**Description**

void

**Session.addUpgrade(UpgradeTileModel upgrade)**

Adds an upgrade tile to the player's collection.

## Constructor parameters in `com.example.project.services` with type arguments of type `UpgradeTileModel`

### Modifier

### Constructor

### Description

protected

```
Session(int newHandSize, int newWordViewSize, int newRedrawWindowSize,  
javafx.collections.ObservableList<UpgradeTileModel> newUpgrades, User newUser, int newMoney,  
int newLevelsBeaten, int currentLevelRequirement, int newFirstLevelsRequirement, int newInitialMoney)
```

Constructor for injecting values in for unit test.

## Uses of `UpgradeTileModel` in `com.example.project.services.shopItems`

### Methods in `com.example.project.services.shopItems` that return `UpgradeTileModel`

#### Modifier and Type

#### Method

#### Description

static `UpgradeTileModel`

`UpgradeTiles.getRandomUpgradeTile()`

Gets random upgrade tile.

static `UpgradeTileModel`

`UpgradeTiles.getTile(int index)`

Gets the upgrade tile at index.

## Package com.example.project.models.tiles

package com.example.project.models.tiles

### Related Packages

#### Package

#### Description

[com.example.project.models](#)

[com.example.project.models.gameScreens](#)

[com.example.project.models.popups](#)

[com.example.project.models.tileGroups](#)

### All Classes and Interfaces

### Classes

### Record Classes

#### Class

#### Description

[EmptyTileSlotModel](#)

An empty tile slot model.

[LetterTileModel](#)

Represents the letter tiles the player plays holds, redraws.

[ScrabbleTileProvider](#)

Static class to store the scrabble base letter value.

[ScrabbleTileProvider.ScrabbleTileData](#)

[TileModel](#)

Parent tile class for objects in the scene that are tiles.

[UpgradeEffects](#)

upgrade effect methods

[UpgradeTileModel](#)

Represents the upgrade tiles sold at the shop.

[UpgradeTileModel.UpgradeBuilder](#)

builder class for the upgrades

## Hierarchy For Package com.example.project.models.tiles

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
  - [com.example.project.models.tiles.ScrabbleTileProvider](#)
  - [com.example.project.models.tiles.TileModel](#)
    - [com.example.project.models.tiles.EmptyTileSlotModel](#)
    - [com.example.project.models.tiles.LetterTileModel](#)
    - [com.example.project.models.tiles.UpgradeTileModel](#)
  - [com.example.project.models.tiles.UpgradeEffects](#)
  - [com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder](#)

### Record Class Hierarchy

- [java.lang.Object](#)
  - [java.lang.Record](#)
    - [com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData](#)

---

Copyright © 2025. All rights reserved.

## Uses of Package com.example.project.models.tiles

### Packages that use com.example.project.models.tiles

#### Package

#### Description

com.example.project.controllers.tiles

com.example.project.models.gameScreens

com.example.project.models.tileGroups

com.example.project.models.tiles

com.example.project.services

com.example.project.services.shopItems

### Classes in com.example.project.models.tiles used by com.example.project.controllers.tiles

#### Class

#### Description

**EmptyTileSlotModel**

An empty tile slot model.

**LetterTileModel**

Represents the letter tiles the player plays holds, redraws.

**TileModel**

Parent tile class for objects in the scene that are tiles.

**UpgradeTileModel**

Represents the upgrade tiles sold at the shop.

### Classes in com.example.project.models.tiles used by com.example.project.models.gameScreens

#### Class

#### Description

**LetterTileModel**

Represents the letter tiles the player plays holds, redraws.

**UpgradeTileModel**

Represents the upgrade tiles sold at the shop.

### Classes in com.example.project.models.tiles used by com.example.project.models.tileGroups

#### Class

#### Description

**LetterTileModel**

Represents the letter tiles the player plays holds, redraws.

**TileModel**

Parent tile class for objects in the scene that are tiles.

**UpgradeTileModel**

Represents the upgrade tiles sold at the shop.

### Classes in com.example.project.models.tiles used by com.example.project.models.tiles

#### Class

#### Description

##### **LetterTileModel**

Represents the letter tiles the player plays holds, redraws.

##### **TileModel**

Parent tile class for objects in the scene that are tiles.

##### **UpgradeTileModel**

Represents the upgrade tiles sold at the shop.

##### **UpgradeTileModel.UpgradeBuilder**

builder class for the upgrades

### Classes in com.example.project.models.tiles used by com.example.project.services

#### Class

#### Description

##### **UpgradeTileModel**

Represents the upgrade tiles sold at the shop.

### Classes in com.example.project.models.tiles used by com.example.project.services.shopItems

#### Class

#### Description

##### **UpgradeTileModel**

Represents the upgrade tiles sold at the shop.

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Class ShopController

```
java.lang.Object
  com.example.project.controllers.gameScreens.GameScreenController
    com.example.project.controllers.gameScreens.ShopController
```

```
public class ShopController
extends GameScreenController
```

Shop view controller.

### Nested Class Summary

**Nested classes/interfaces inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
GameScreenController.SidebarLoaded
```

### Field Summary

**Fields inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
logger
```

### Constructor Summary

#### Constructors

**Modifier**

**Constructor**

**Description**

```
ShopController()
```

no arg constructor.

protected

```
ShopController(ShopModel model, UpgradeTileGroup players, UpgradeTileGroup shopItems, Logger logger)
```

protected constructor for unit testing with mock model injection.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

void

```
initialize()
```

FXML initialise function called once when the .fxml is loaded on application launch.

protected void

```
onNextLevelPressed()
```

void

### **onSceneChangedToThis()**

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

```
protected void
```

```
onUpgradeClicked(UpgradeTileController controller)
```

### **Methods inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
loadSidebar
```

### **Methods inherited from class java.lang.Object**

```
clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()
```

## **Constructor Details**

### **ShopController**

```
public ShopController()
```

no arg constructor.

### **ShopController**

```
protected ShopController(ShopModel model,  
                         UpgradeTileGroup players,  
                         UpgradeTileGroup shopItems,  
                         Logger logger)
```

protected constructor for unit testing with mock model injection.

## **Method Details**

### **initialize**

```
public void initialize()
```

FXML initialise function called once when the .fxml is loaded on application launch.

### **onSceneChangedToThis**

```
public void onSceneChangedToThis()
```

#### **Description copied from class: GameScreenController**

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

#### **Specified by:**

onSceneChangedToThis in class GameScreenController

### **onUpgradeClicked**

```
protected void onUpgradeClicked(UpgradeTileController controller)
```

#### **Parameters:**

controller - upgrade tile ui element clicked on.

### **onNextLevelPressed**

```
protected void onNextLevelPressed()
```

---

Copyright © 2025. All rights reserved.

Module com.example.project

## Package com.example.project

package com.example.project

### Related Packages

#### Package

#### Description

[com.example.project.controllers](#)

[com.example.project.models](#)

[com.example.project.services](#)

### Classes

#### Class

#### Description

#### Application

The entry point of the application.

## Hierarchy For Package com.example.project

### Package Hierarchies:

All Packages

### Class Hierarchy

- [java.lang.Object](#) ↗
  - [javafx.application.Application](#)
    - [com.example.project.Application](#)

## **Uses of Package com.example.project**

No usage of com.example.project

---

Copyright © 2025. All rights reserved.

**Module** com.example.project  
**Package** com.example.project.services

## Class FXMLPageLoader

java.lang.Object<sup>2</sup>  
    com.example.project.services.FXMLPageLoader

### All Implemented Interfaces:

PageLoader

```
public class FXMLPageLoader
extends Object2
implements PageLoader
```

FXML Loader wrapper class for the project to load the game screen FXML pages.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**FXMLPageLoader()**

constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

<T> T

**getController()**

Get the .fxml pages controller

javafx.scene.Parent

**load(String<sup>2</sup> fxmlPath)**

Load method to load fxml page.

#### Methods inherited from class java.lang.Object<sup>2</sup>

clone<sup>2</sup>, equals<sup>2</sup>, finalize<sup>2</sup>, getClass<sup>2</sup>, hashCode<sup>2</sup>, notify<sup>2</sup>, notifyAll<sup>2</sup>, toString<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>

### Constructor Details

#### FXMLPageLoader

```
public FXMLPageLoader()
constructor.
```

### Method Details

#### load

```
public javafx.scene.Parent load(String2 fxmlPath)
```

```
throws IOException
```

#### Description copied from interface: PageLoader

Load method to load fxml page.

##### Specified by:

[load](#) in interface [PageLoader](#)

##### Parameters:

fxmlPath - path to .fxml mfile.

##### Returns:

returns the fxml file root node.

##### Throws:

[IOException](#) - exception on failing.

## getController

```
public <T> T getController()
```

#### Description copied from interface: PageLoader

Get the .fxml pages controller

##### Specified by:

[getController](#) in interface [PageLoader](#)

##### Type Parameters:

T - controller type.

##### Returns:

page's controller.

**Module** com.example.project

**Package** com.example.project.services

## Enum Class GameScenes

java.lang.Object  
  java.lang.Enum<GameScenes>  
    com.example.project.services.GameScenes

### All Implemented Interfaces:

Serializable, Comparable<GameScenes>, Constable

---

public enum GameScenes  
extends Enum<GameScenes>

Game Scenes enum.

### Nested Class Summary

#### **Nested classes/interfaces inherited from class java.lang.Enum**

Enum.EnumDesc<E> extends Enum<E>>

### Enum Constant Summary

#### Enum Constants

Enum Constant

Description

DAILY\_REWARD

Daily Reward Spinner Screen.

LEVEL

level Screen

LOGIN

Login Screen.

MAINMENU

Main Menu Screen.

SHOP

Shop screen.

### Method Summary

#### All Methods

#### Static Methods

#### Concrete Methods

Modifier and Type

Method

Description

static GameScenes

valueOf(String name)

Returns the enum constant of this class with the specified name.

static GameScenes[]

values()

Returns an array containing the constants of this enum class, in the order they are declared.

#### Methods inherited from class java.lang.Enum

clone<sup>2</sup>, compareTo<sup>2</sup>, describeConstable<sup>2</sup>, equals<sup>2</sup>, finalize<sup>2</sup>, getDeclaringClass<sup>2</sup>, hashCode<sup>2</sup>, name<sup>2</sup>, ordinal<sup>2</sup>,  
toString<sup>2</sup>, valueOf<sup>2</sup>

#### Methods inherited from class java.lang.Object<sup>2</sup>

getClass<sup>2</sup>, notify<sup>2</sup>, notifyAll<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>

### Enum Constant Details

#### LOGIN

public static final GameScenes LOGIN

Login Screen.

#### MAINMENU

public static final GameScenes MAINMENU

Main Menu Screen.

#### DAILY\_REWARD

public static final GameScenes DAILY\_REWARD

Daily Reward Spinner Screen.

#### LEVEL

public static final GameScenes LEVEL

level Screen

#### SHOP

public static final GameScenes SHOP

Shop screen.

### Method Details

#### values

public static GameScenes[] values()

Returns an array containing the constants of this enum class, in the order they are declared.

**Returns:**

an array containing the constants of this enum class, in the order they are declared

#### valueOf

public static GameScenes valueOf(String<sup>2</sup> name)

Returns the enum constant of this class with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this class. (Extraneous whitespace characters are not permitted.)

**Parameters:**

name - the name of the enum constant to be returned.

**Returns:**

the enum constant with the specified name

**Throws:**

IllegalArgumentException<sup>2</sup> - if this enum class has no constant with the specified name

[NullPointerException](#) - if the argument is null

**Module** com.example.project  
**Package** com.example.project.services

## Class Logger

java.lang.Object<sup>✉</sup>  
    com.example.project.services.Logger

```
public class Logger  
extends Object✉
```

com.example.project.services.Logger for Application and to show up in test and build CI workflow.

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

**Logger()**

Default constructor writes to console does not capture.

**Logger(ByteArrayOutputStream<sup>✉</sup> capturedErrStream, ByteArrayOutputStream<sup>✉</sup> capturedOutStream)**

Constructor for unit tests.

protected

**Logger(ByteArrayOutputStream<sup>✉</sup> capturedErrStream, ByteArrayOutputStream<sup>✉</sup> capturedOutStream, PrintStream<sup>✉</sup> mockRealErrOutput, PrintStream<sup>✉</sup> mockRealStdOutput)**

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

void

**clearLogs()**

Clears captured logs use in tests teardown.

**String<sup>✉</sup>**

**getErrorLogs()**

**String<sup>✉</sup>**

**getLogs()**

void

**logError(String<sup>✉</sup> message)**

Logs an error message to standard error.

void

**logMessage(String<sup>✉</sup> formattedMessage)**

Log message to System.out.

```
void  
setPrintToConsole(boolean value)
```

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

### Constructor Details

#### Logger

```
public Logger()
```

Default constructor writes to console does not capture. For the project files. Use below for unit tests.

#### Logger

```
public Logger(ByteArrayOutputStream capturedErrStream,  
             ByteArrayOutputStream capturedOutStream)
```

Constructor for unit tests. Logger with constructor to input the byte array output stream to write to. (for mocking a log to check get methods.)

**Parameters:**

`capturedErrStream` - byte array to store error logs.

`capturedOutStream` - byte array to store standard output logs.

#### Logger

```
protected Logger(ByteArrayOutputStream capturedErrStream,  
                 ByteArrayOutputStream capturedOutStream,  
                 PrintStream mockRealErrOutput,  
                 PrintStream mockRealStdOutput)
```

### Method Details

#### setPrintToConsole

```
public void setPrintToConsole(boolean value)
```

**Parameters:**

`value` - if this logger will also print to the console.

#### getErrorLogs

```
public String getErrorLogs()
```

**Returns:**

Gets the error log messages.

#### getLogs

```
public String getLogs()
```

**Returns:**

returns the standard log messages.

#### logError

```
public void logError(String2 message)
```

Logs an error message to standard error. And adds a newline.

The message can include format specifiers like in `String.format(String, Object...)`.

**Parameters:**

message - the error message format string (e.g., "Failed to connect to %s")

## clearLogs

```
public void clearLogs()
```

Clears captured logs use in tests teardown.

## logMessage

```
public void logMessage(String2 formattedMessage)
```

Log message to System.out.

**Parameters:**

formattedMessage - message.

**Module** com.example.project  
**Package** com.example.project.services

## Interface PageLoader

### All Known Implementing Classes:

FXMLPageLoader

public interface PageLoader

Interface for page loading.

### Method Summary

[All Methods](#)   [Instance Methods](#)   [Abstract Methods](#)

#### Modifier and Type

#### Method

#### Description

<T> T

**getController()**

Get the .fxml pages controller

javafx.scene.Parent

**load(String<sup>✉</sup> fxmlPath)**

Load method to load fxml page.

### Method Details

#### load

```
javafx.scene.Parent load(String✉ fxmlPath)
    throws IOException
```

Load method to load fxml page.

##### Parameters:

fxmlPath - path to .fxml mfile.

##### Returns:

returns the fxml file root node.

##### Throws:

IOException<sup>✉</sup> - exception on failing.

#### getController

<T> T getController()

Get the .fxml pages controller

##### Type Parameters:

T - controller type.

##### Returns:

page's controller.

**Module** com.example.project  
**Package** com.example.project.services

## Class PasswordHasher

java.lang.Object  
  com.example.project.services.PasswordHasher

```
public class PasswordHasher  
extends Object
```

Hashes passwords.

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

**PasswordEncoder()**

public constructor.

protected

**PasswordHasher(Logger logger)**

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

boolean

**doesPasswordMatch(String candidate, String hashed)**

Verifies a candidate password against a hashed password.

**String**

**hashPassword(String rawPassword)**

Hashes a raw password using BCrypt.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### PasswordEncoder

```
public PasswordHasher()
```

public constructor.

#### PasswordEncoder

```
protected PasswordHasher(Logger logger)
```

## Method Details

### hashPassword

```
public String2 hashPassword(String2 rawPassword)
```

Hashes a raw password using BCrypt.

**Parameters:**

rawPassword - the plain text password

**Returns:**

the hashed password string

### doesPasswordMatch

```
public boolean doesPasswordMatch(String2 candidate,  
                                String2 hashed)
```

Verifies a candidate password against a hashed password.

**Parameters:**

candidate - the plain text password entered by the user

hashed - the hashed password stored in the database

**Returns:**

true if the password matches, false otherwise

**Module** com.example.project  
**Package** com.example.project.services

## Class PopupLoader

java.lang.Object  
com.example.project.services.PopupLoader

```
public class PopupLoader  
extends Object
```

class to load the popup and initialize it with a controller

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

**PopupLoader()**

Constructor.

protected

**PopupLoader(FXMLLoader loader)**

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

<C extends PopupController<P>, P extends PopupModel>

C

**createPopupController(P popupModel)**

create popup controller.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### PopupLoader

```
protected PopupLoader(FXMLLoader loader)
```

#### PopupLoader

```
public PopupLoader()
```

Constructor.

### Method Details

## createPopupController

```
public <C extends PopupController<P>,
      P extends PopupModel>
C createPopupController(P popupModel)
```

create popup controller.

### Type Parameters:

C - class of popup controller.

P - class of popup.

### Parameters:

popupModel - popup object.

### Returns:

returns controller of the tile.

**Module** com.example.project  
**Package** com.example.project.services

## Class SVGIcons

java.lang.Object<sup>2</sup>  
com.example.project.services.SVGIcons

```
public class SVGIcons  
extends Object2
```

a static class to hold some icons for the application.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

SVGIcons()

### Method Summary

#### All Methods

#### Static Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

static javafx.scene.shape.SVGPath

**getCogIcon()**

gets a cog icon.

#### Methods inherited from class java.lang.Object<sup>2</sup>

clone<sup>2</sup>, equals<sup>2</sup>, finalize<sup>2</sup>, getClass<sup>2</sup>, hashCode<sup>2</sup>, notify<sup>2</sup>, notifyAll<sup>2</sup>, toString<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>

### Constructor Details

#### SVGIcons

```
public SVGIcons()
```

### Method Details

#### getCogIcon

```
public static javafx.scene.shape.SVGPath getCogIcon()
```

gets a cog icon.

##### Returns:

icon SVG.

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Class SidebarController

java.lang.Object  
com.example.project.controllers.gameScreens.SidebarController

```
public class SidebarController  
extends Object
```

The sidebar on shop and level with the current round info.

### Constructor Summary

#### Constructors

**Constructor**

**Description**

**SidebarController()**

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

void

**bindPersistentInfo()**

Binds things used in multiple screens, money, redraws plays.

javafx.scene.control.Label

**getComboLabel()**

Returns combo label.

javafx.scene.control.Label

**getCurrentScoreLabel()**

current score label.

javafx.scene.control.Label

**getMultiplierLabel()**

word score multiplier label.

javafx.scene.control.Label

**getRawPointsLabel()**

raw points label to be multiplied by multiplier.

void

**onlyShopShopInfo()**

Only show things relevant to upgrades or the shop.

void

**setupProperties(LevelModel levelModel)**

Setup sync able properties.

**Methods inherited from class java.lang.Object**

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructor Details

### SidebarController

```
public SidebarController()
```

## Method Details

### bindPersistentInfo

```
public void bindPersistentInfo()
```

Binds things used in multiple screens, money, redraws plays.

### onlyShopShopInfo

```
public void onlyShopShopInfo()
```

Only show things relevant to upgrades or the shop. Money, redraws, plays.

### getComboLabel

```
public javafx.scene.control.Label getComboLabel()
```

Returns combo label.

**Returns:**

combo label.

### setupProperties

```
public void setupProperties(LevelModel levelModel)
```

Setup sync able properties.

**Parameters:**

levelModel - level model.

### getRawPointsLabel

```
public javafx.scene.control.Label getRawPointsLabel()
```

raw points label to be multiplied by multiplier.

**Returns:**

label.

### getCurrentScoreLabel

```
public javafx.scene.control.Label getCurrentScoreLabel()
```

current score label.

**Returns:**

label.

### getmultiplierLabel

```
public javafx.scene.control.Label getmultiplierLabel()
```

word score multiplier label.

**Returns:**

label.

**Module** com.example.project  
**Package** com.example.project.services

## Class SceneManager

java.lang.Object<sup>↳</sup>  
    com.example.project.services.SceneManager

```
public class SceneManager  
extends Object↳
```

Represents the scene manager class.

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

**SceneManager(SceneManager newInstance)**

Constructor to inject an instance.

protected

**SceneManager(SceneManager newInstance, RootLayoutController newRootController, Map<GameScenes, GameScreenController> controllersToUse, Map<GameScenes, javafx.scene.Parent> pagesToUse)**

Constructor with injection for tests

### Method Summary

#### All Methods

#### Static Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

static SceneManager

**getInstance()**

void

**initialise(RootLayoutController theRootController, PageLoader loader)**

loads the fxml of all the scenes on launch to switch to via scene manaager.

void

**switchScene(GameScenes type)**

Switch between scenes specifiy scene type of type from GameScenes.

#### Methods inherited from class java.lang.Object<sup>↳</sup>

clone<sup>↳</sup>, equals<sup>↳</sup>, finalize<sup>↳</sup>, getClass<sup>↳</sup>, hashCode<sup>↳</sup>, notify<sup>↳</sup>, notifyAll<sup>↳</sup>, toString<sup>↳</sup>, wait<sup>↳</sup>, wait<sup>↳</sup>, wait<sup>↳</sup>

### Constructor Details

#### SceneManager

**public SceneManager(SceneManager newInstance)**

Constructor to inject an instance.

**Parameters:**

newInstance - new instance.

## SceneManager

```
protected SceneManager(SceneManager newInstance,  
                      RootLayoutController newRootController,  
                      Map<GameScenes, GameScreenController> controllersToUse,  
                      Map<GameScenes, javafx.scene.Parent> pagesToUse)
```

Constructor with injection for tests

**Parameters:**

newInstance - instance

newRootController - root controller

controllersToUse - controllers

pagesToUse - pages

## Method Details

### getInstance

```
public static SceneManager getInstance()
```

**Returns:**

Gets the programs Scene Manager instance.

### initialise

```
public void initialise(RootLayoutController theRootController,  
                      PageLoader loader)
```

loads the fxml of all the scenes on launch to switch to via scene manager.

**Parameters:**

theRootController - the root Controller for the rootLayout scene.

loader - FXMLLoader

### switchScene

```
public void switchScene(GameScenes type)
```

Switch between scenes specify scene type of type from GameScenes.

**Parameters:**

type - scene.

**Module** com.example.project  
**Package** com.example.project.services

## Class Session

java.lang.Object<sup>↳</sup>  
    com.example.project.services.Session

```
public class Session  
extends Object↳
```

Game Session. holds info of the current session.

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

**Session()**

Default constructor.

protected

**Session(int newHandSize, int newWordViewSize, int newRedrawWindowSize,  
javafx.collections.ObservableList<UpgradeTileModel> newUpgrades, User newUser, int newMoney,  
int newLevelsBeaten, int currentLevelRequirement, int newFirstLevelsRequirement, int newInitialMoney)**

Constructor for injecting values in for unit test.

### Method Summary

#### All Methods

#### Static Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

void

**addMoney(int amount)**

Adds money to the player's balance.

void

**addUpgrade(UpgradeTileModel upgrade)**

Adds an upgrade tile to the player's collection.

javafx.beans.property.ReadOnlyIntegerWrapper

**getCurrentPlays()**

gets the current plays.

javafx.beans.property.ReadOnlyIntegerWrapper

**getCurrentRedraws()**

gets the redraws property.

int

**getHandSize()**

gets hand size.

static Session

**getInstance()**

Gets session.

#### **LocalDate**<sup>2</sup>

##### **getLastRewardDate()**

Gets the last date the player claimed their daily reward.

#### **LevelModel**

##### **getLevelModel()**

javafx.beans.property.ReadOnlyIntegerWrapper

##### **getLevelRequirement()**

protected int

##### **getLevelsBeaten()**

javafx.beans.property.IntegerProperty

##### **getMoneyProperty()**

Returns the read-only money property for binding to UI components.

#### **Integer**<sup>2</sup>

##### **getRedrawWindowSize()**

gets redraw window size (number of slots)

javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel>

##### **getUpgradeTilesProperty()**

gets upgrade tile property

int

##### **getWordWindowSize()**

gets word size.

boolean

##### **hasClaimedRewardToday()**

Checks if the player already claimed today's reward.

void

##### **resetGame()**

Resets the current session when you lose

void

##### **resetMoney()**

Resets the player's money to the initial state (e.g.

void

##### **resetPlaysRedraws()**

Reset the plays and redraws.

void

##### **setLastRewardDate(LocalDate<sup>2</sup> date)**

Sets the date the daily reward was last claimed.

void

##### **setLevelModel(LevelModel levelModel)**

sets the session's level model

void

##### **setUser(User newUser)**

set new user.

boolean

**spendMoney(int amount)**

This attempts to spend the specified amount of money from the player's account.

void

**updateLevelInfo()**

Increments how many points are required to beat the level.

#### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructor Details

### Session

```
protected Session(int newHandSize,  
                 int newWordViewSize,  
                 int newRedrawWindowSize,  
                 javafx.collections.ObservableList<UpgradeTileModel> newUpgrades,  
                 User newUser,  
                 int newMoney,  
                 int newLevelsBeaten,  
                 int currentLevelRequirement,  
                 int newFirstLevelsRequirement,  
                 int newInitialMoney)
```

Constructor for injecting values in for unit test.

#### Parameters:

newHandSize - hand size.

newWordViewSize - word length allowed.

newRedrawWindowSize - redraw window size.

newUpgrades - upgrade tiles.

newUser - user.

newMoney - money.

newLevelsBeaten - levels beaten.

currentLevelRequirement - current level requirement.

newFirstLevelsRequirement - first level requirement.

### Session

```
public Session()
```

Default constructor.

## Method Details

### getLastRewardDate

```
public LocalDate getLastRewardDate()
```

Gets the last date the player claimed their daily reward.

#### Returns:

the date the reward was last claimed, or null if never claimed

### getInstance

```
public static Session getInstance()
```

Gets session.

**Returns:**

session instance.

## getLevelRequirement

```
public javafx.beans.property.ReadOnlyIntegerWrapper getLevelRequirement()
```

**Returns:**

points required for the play to score at least to beat the level.

## getLevelsBeaten

```
protected int getLevelsBeaten()
```

## setLevelModel

```
public void setLevelModel(LevelModel levelModel)
```

sets the session's level model

**Parameters:**

levelModel - the current level model

## getLevelModel

```
public LevelModel getLevelModel()
```

**Returns:**

the current level model

## getMoneyProperty

```
public javafx.beans.property.IntegerProperty getMoneyProperty()
```

Returns the read-only money property for binding to UI components. This allows UI elements to automatically update when the players money changes.

**Returns:**

ReadOnlyIntegerProperty representing the player's current money amount

## addUpgrade

```
public void addUpgrade(UpgradeTileModel upgrade)
```

Adds an upgrade tile to the player's collection. This will automatically update all UI displays bound to the upgrades property.

**Parameters:**

upgrade - this upgrades the tile to add it to the players collection

## spendMoney

```
public boolean spendMoney(int amount)
```

This attempts to spend the specified amount of money from the player's account. This operation should only succeed if the player has enough money.

**Parameters:**

amount - the amount of money to spend (cannot be negative)

**Returns:**

true if the transaction was successful (player had enough money), false if the player is a brokey

**Throws:**

IllegalArgumentException<sup>4</sup> - if the amount is negative

## setUser

```
public void setUser(User newUser)
```

set new user.

**Parameters:**

newUser - user that logged in.

## getHandSize

```
public int getHandSize()
```

gets hand size.

**Returns:**

returns number of tiles allowed in hand.

## getWordWindowSize

```
public int getWordWindowSize()
```

gets word size.

**Returns:**

return int word size.

## getUpgradeTilesProperty

```
public javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> getUpgradeTilesProperty()
```

gets upgrade tile property

**Returns:**

upgrade tiles model list

## updateLevelInfo

```
public void updateLevelInfo()
```

Increments how many points are required to beat the level.

## resetGame

```
public void resetGame()
```

Resets the current session when you lose

## getRedrawWindowSize

```
public Integer5 getRedrawWindowSize()
```

gets redraw window size (number of slots)

**Returns:**

return int redraw window size.

## getCurrentPlays

```
public javafx.beans.property.ReadOnlyIntegerWrapper getCurrentPlays()
```

gets the current plays.

**Returns:**

current plays remaining.

### **getCurrentRedraws**

```
public javafx.beans.property.ReadOnlyIntegerWrapper getCurrentRedraws()  
gets the redraws property.
```

**Returns:**

the current redraws.

### **resetPlaysRedraws**

```
public void resetPlaysRedraws()  
Reset the plays and redraws.
```

### **addMoney**

```
public void addMoney(int amount)  
Adds money to the player's balance.
```

**Parameters:**

amount - amount to add

### **setLastRewardDate**

```
public void setLastRewardDate(LocalDate2 date)  
Sets the date the daily reward was last claimed.
```

**Parameters:**

date - LocalDate of the reward claim

### **hasClaimedRewardToday**

```
public boolean hasClaimedRewardToday()  
Checks if the player already claimed today's reward.  
  
Returns:  
true if already claimed today
```

### **resetMoney**

```
public void resetMoney()  
Resets the player's money to the initial state (e.g. 0).
```

## Uses of Class com.example.project.services.FXMLPageLoader

### Packages that use FXMLPageLoader

#### Package

#### Description

com.example.project.controllers.tiles

com.example.project.services

### Uses of FXMLPageLoader in com.example.project.controllers.tiles

#### Constructors in com.example.project.controllers.tiles with parameters of type FXMLPageLoader

##### Modifier

##### Constructor

##### Description

protected

`TileControllerFactory(FXMLPageLoader loader)`

### Uses of FXMLPageLoader in com.example.project.services

#### Constructors in com.example.project.services with parameters of type FXMLPageLoader

##### Modifier

##### Constructor

##### Description

protected

`PopupLoader(FXMLPageLoader loader)`

## Uses of Enum Class

### com.example.project.services.GameScenes

#### Packages that use GameScenes

##### Package

##### Description

com.example.project.services

#### Uses of GameScenes in com.example.project.services

#### Methods in com.example.project.services that return GameScenes

##### Modifier and Type

##### Method

##### Description

static GameScenes

**GameScenes.valueOf(String<sup>2</sup> name)**

Returns the enum constant of this class with the specified name.

static GameScenes[]

**GameScenes.values()**

Returns an array containing the constants of this enum class, in the order they are declared.

#### Methods in com.example.project.services with parameters of type GameScenes

##### Modifier and Type

##### Method

##### Description

void

**SceneManager.switchScene(GameScenes type)**

Switch between scenes specify scene type of type from GameScenes.

#### Constructor parameters in com.example.project.services with type arguments of type GameScenes

##### Modifier

##### Constructor

##### Description

protected

**SceneManager(SceneManager newInstance, RootLayoutController newRootController, Map<sup>2</sup><GameScenes, GameScreenController> controllersToUse, Map<sup>2</sup><GameScenes, javafx.scene.Parent> pagesToUse)**

Constructor with injection for tests

## Uses of Class com.example.project.services.Logger

### Packages that use Logger

#### Package

#### Description

com.example.project.controllers.gameScreens

com.example.project.models.gameScreens

com.example.project.services

com.example.project.services.sound

com.example.project.services.sqlite.dAOs

## Uses of Logger in com.example.project.controllers.gameScreens

### Fields in com.example.project.controllers.gameScreens declared as Logger

#### Modifier and Type

#### Field

#### Description

protected `Logger`

`GameScreenController.logger`

### Constructors in com.example.project.controllers.gameScreens with parameters of type Logger

#### Modifier

#### Constructor

#### Description

protected

`ShopController(ShopModel model, UpgradeTileGroup players, UpgradeTileGroup shopItems, Logger logger)`

protected constructor for unit testing with mock model injection.

## Uses of Logger in com.example.project.models.gameScreens

### Fields in com.example.project.models.gameScreens declared as Logger

#### Modifier and Type

#### Field

#### Description

protected `Logger`

`GameScreenModel.logger`

### Constructors in com.example.project.models.gameScreens with parameters of type Logger

#### Modifier

#### Constructor

#### Description

protected  
`LevelModel(Session session, Logger logger)`

protected  
`ShopModel(Session session, Logger logger)`  
Constructor for tests.

## Uses of `Logger` in `com.example.project.services`

### Constructors in `com.example.project.services` with parameters of type `Logger`

**Modifier**

**Constructor**

**Description**

protected  
`PasswordHasher(Logger logger)`

## Uses of `Logger` in `com.example.project.services.sound`

### Constructors in `com.example.project.services.sound` with parameters of type `Logger`

**Modifier**

**Constructor**

**Description**

protected  
`GameSoundPlayer(String filePath, Logger logger)`  
  
protected  
`GameSoundPlayer(String filePath, Logger logger, float gainAmount)`

## Uses of `Logger` in `com.example.project.services.sqlite.dAOs`

### Constructors in `com.example.project.services.sqlite.dAOs` with parameters of type `Logger`

**Modifier**

**Constructor**

**Description**

`DictionaryDAO(Connection connection, Logger logger)`  
constructor with injection for tests.  
  
protected  
`UsersDAO(PasswordHasher hasher, Connection connection, Logger logger)`

`UsersDAO(Connection connection, Logger logger)`  
Constructor with injection for unit tests.

## Uses of Interface com.example.project.services.PageLoader

### Packages that use PageLoader

#### Package

#### Description

com.example.project.services

## Uses of PageLoader in com.example.project.services

### Classes in com.example.project.services that implement PageLoader

#### Modifier and Type

#### Class

#### Description

class

### FXMLPageLoader

FXML Loader wrapper class for the project to load the game screen FXML pages.

### Methods in com.example.project.services with parameters of type PageLoader

#### Modifier and Type

#### Method

#### Description

void

**SceneManager.initialise(RootLayoutController theRootController, PageLoader loader)**

loads the fxml of all the scenes on launch to switch to via scene manaager.

## Uses of Class com.example.project.services.PasswordHasher

### Packages that use PasswordHasher

#### Package

#### Description

com.example.project.models.gameScreens

com.example.project.services.sqlite.dAOs

### Uses of PasswordHasher in com.example.project.models.gameScreens

#### Constructors in com.example.project.models.gameScreens with parameters of type PasswordHasher

##### Modifier

##### Constructor

##### Description

`LoginModel(Session session, UsersDAO dao, PasswordHasher passwordHasher)`

constructor.

### Uses of PasswordHasher in com.example.project.services.sqlite.dAOs

#### Constructors in com.example.project.services.sqlite.dAOs with parameters of type PasswordHasher

##### Modifier

##### Constructor

##### Description

protected

`UsersDAO(PasswordHasher hasher, Connection✉ connection, Logger logger)`

## **Uses of Class com.example.project.services.PopupLoader**

No usage of com.example.project.services.PopupLoader

---

Copyright © 2025. All rights reserved.

## **Uses of Class com.example.project.services.SVGIcons**

No usage of com.example.project.services.SVGIcons

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.services.SceneManager

### Packages that use SceneManager

#### Package

#### Description

com.example.project.services

### Uses of SceneManager in com.example.project.services

#### Methods in com.example.project.services that return SceneManager

##### Modifier and Type

##### Method

##### Description

static SceneManager

SceneManager.getInstance()

#### Constructors in com.example.project.services with parameters of type SceneManager

##### Modifier

##### Constructor

##### Description

SceneManager(SceneManager newInstance)

Constructor to inject an instance.

protected

SceneManager(SceneManager newInstance, RootLayoutController newRootController, Map<GameScenes, GameScreenController> controllersToUse, Map<GameScenes, javafx.scene.Parent> pagesToUse)

Constructor with injection for tests

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens.animations

## Class AnimationWrapper

java.lang.Object<sup>↳</sup>  
com.example.project.controllers.gameScreens.animations.AnimationWrapper

**Direct Known Subclasses:**

LevelScoreSequence, TextEmphasisAnimation

---

```
public abstract class AnimationWrapper  
extends Object↳
```

Animation wrapper class for our projects animations that get used.

### Field Summary

#### Fields

**Modifier and Type**

**Field**

**Description**

protected final javafx.animation.SequentialTransition

**sequentialAnimation**

### Constructor Summary

#### Constructors

**Modifier**

**Constructor**

**Description**

protected

**AnimationWrapper()**

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

javafx.collections.ObservableList<javafx.animation.Animation>

**getChildren()**

returns animations children.

void

**play()**

play.

void

**setOnFinished(javafx.event.EventHandler<javafx.event.ActionEvent> handler)**

Set on finished.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Details

### sequentialAnimation

```
protected final javafx.animation.SequentialTransition sequentialAnimation
```

## Constructor Details

### AnimationWrapper

```
protected AnimationWrapper()
```

## Method Details

### play

```
public void play()
```

play.

### setOnFinished

```
public void setOnFinished(javafx.event.EventHandler<javafx.event.ActionEvent> handler)
```

Set on finished.

#### Parameters:

handler - actions to do.

### getChildren

```
public javafx.collections.ObservableList<javafx.animation.Animation> getChildren()
```

returns animations children.

#### Returns:

list of animations.

## Uses of Class com.example.project.services.Session

### Packages that use Session

#### Package

#### Description

com.example.project.models.gameScreens

com.example.project.services

### Uses of Session in com.example.project.models.gameScreens

#### Fields in com.example.project.models.gameScreens declared as Session

##### Modifier and Type

##### Field

##### Description

protected Session

GameScreenModel.session

#### Constructors in com.example.project.models.gameScreens with parameters of type Session

##### Modifier

##### Constructor

##### Description

protected

GameScreenModel(Session session)

LevelModel(Session session)

Default constructor.

protected

LevelModel(Session session, Logger logger)

LoginModel(Session session, UsersDAO dao, PasswordHasher passwordHasher)

constructor.

MainMenuModel(Session session)

constructor.

ShopModel(Session session)

Constructor

protected

ShopModel(Session session, Logger logger)

Constructor for tests.

## Uses of Session in com.example.project.services

### Methods in com.example.project.services that return Session

#### Modifier and Type

#### Method

#### Description

static Session

**Session.getInstance()**

Gets session.

## Package com.example.project.services

package com.example.project.services

### Related Packages

#### Package

#### Description

[com.example.project](#)

[com.example.project.services.shopItems](#)

[com.example.project.services.sound](#)

[com.example.project.services.sqlite](#)

[com.example.project.controllers](#)

[com.example.project.models](#)

### All Classes and Interfaces

### Interfaces

### Classes

### Enum Classes

#### Class

#### Description

**FXMLPageLoader**

FXML Loader wrapper class for the project to load the game screen FXML pages.

**GameScenes**

Game Scenes enum.

**Logger**

com.example.project.services.Logger for Application and to show up in test and build CI workflow.

**PageLoader**

Interface for page loading.

**PasswordHasher**

Hashes passwords.

**PopupLoader**

class to load the popup and initialize it with a controller

**SceneManager**

Represents the scene manager class.

**Session**

Game Session.

**SVGIcons**

a static class to hold some icons for the application.

## Hierarchy For Package com.example.project.services

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#) ↗
  - [com.example.project.services.FXMLPageLoader](#) (implements [com.example.project.services.PageLoader](#))
  - [com.example.project.services.Logger](#)
  - [com.example.project.services.PasswordHasher](#)
  - [com.example.project.services.PopupLoader](#)
  - [com.example.project.services.SceneManager](#)
  - [com.example.project.services.Session](#)
  - [com.example.project.services.SVGIcons](#)

### Interface Hierarchy

- [com.example.project.services.PageLoader](#)

### Enum Class Hierarchy

- [java.lang.Object](#) ↗
  - [java.lang.Enum](#) ↗<E> (implements [java.lang.Comparable](#) ↗<T>, [java.lang.constant.Constable](#) ↗, [java.io.Serializable](#) ↗)
    - [com.example.project.services.GameScenes](#)

## Uses of Package com.example.project.services

### Packages that use com.example.project.services

Package	Description
com.example.project.controllers.gameScreens	
com.example.project.controllers.tiles	
com.example.project.models.gameScreens	
com.example.project.services	
com.example.project.services.sound	
com.example.project.services.sqlite.dAOs	

### Classes in com.example.project.services used by com.example.project.controllers.gameScreens

Class	Description
Logger	com.example.project.services.Logger for Application and to show up in test and build CI workflow.

### Classes in com.example.project.services used by com.example.project.controllers.tiles

Class	Description
FXMLPageLoader	FXML Loader wrapper class for the project to load the game screen FXML pages.

### Classes in com.example.project.services used by com.example.project.models.gameScreens

Class	Description
Logger	com.example.project.services.Logger for Application and to show up in test and build CI workflow.
PasswordHasher	Hashes passwords.
Session	Game Session.

### Classes in com.example.project.services used by com.example.project.services

Class	Description
FXMLPageLoader	FXML Loader wrapper class for the project to load the game screen FXML pages.
GameScenes	

Game Scenes enum.

#### **Logger**

com.example.project.services.Logger for Application and to show up in test and build CI workflow.

#### **PageLoader**

Interface for page loading.

#### **SceneManager**

Represents the scene manager class.

#### **Session**

Game Session.

### **Classes in com.example.project.services used by com.example.project.services.sound**

#### **Class**

#### **Description**

#### **Logger**

com.example.project.services.Logger for Application and to show up in test and build CI workflow.

### **Classes in com.example.project.services used by com.example.project.services.sqlite.dAOs**

#### **Class**

#### **Description**

#### **Logger**

com.example.project.services.Logger for Application and to show up in test and build CI workflow.

#### **PasswordHasher**

Hashes passwords.

**Module** com.example.project  
**Package** com.example.project.services.shopItems

## Class UpgradeTiles

java.lang.Object  
    com.example.project.services.shopItems.UpgradeTiles

```
public class UpgradeTiles  
extends Object
```

Static class to store all the possible upgrade tiles available in the game.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

UpgradeTiles()

### Method Summary

#### All Methods

#### Static Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

static UpgradeTileModel

getRandomUpgradeTile()

Gets random upgrade tile.

static UpgradeTileModel

getTile(int index)

Gets the upgrade tile at index.

#### Methods inherited from class java.lang.Object

clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()

### Constructor Details

#### UpgradeTiles

```
public UpgradeTiles()
```

### Method Details

#### getRandomUpgradeTile

```
public static UpgradeTileModel getRandomUpgradeTile()
```

Gets random upgrade tile.

##### Returns:

returns upgrade tile.

## getTile

```
public static UpgradeTileModel getTile(int index)
```

Gets the upgrade tile at index.

**Parameters:**

index - tile to get.

**Returns:**

an upgrade tile model.

## **Uses of Class com.example.project.services.shopItems.UpgradeTiles**

No usage of com.example.project.services.shopItems.UpgradeTiles

---

Copyright © 2025. All rights reserved.

Module com.example.project

## Package com.example.project.services.shopItems

package com.example.project.services.shopItems

### Related Packages

#### Package

#### Description

[com.example.project.services](#)

[com.example.project.services.sound](#)

[com.example.project.services.sqlite](#)

### Classes

#### Class

#### Description

[UpgradeTiles](#)

Static class to store all the possible upgrade tiles available in the game.

Copyright © 2025. All rights reserved.

## Hierarchy For Package com.example.project.services.shopItems

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#) ↗
  - [com.example.project.services.shopItems.UpgradeTiles](#)

---

Copyright © 2025. All rights reserved.

## **Uses of Package com.example.project.services.shopItems**

No usage of com.example.project.services.shopItems

---

Copyright © 2025. All rights reserved.

**Module** com.example.project  
**Package** com.example.project.services.sound

## Class GameMusicPlayer

```
java.lang.Object
  com.example.project.services.sound.GameSoundPlayer
    com.example.project.services.sound.GameMusicPlayer
```

```
public class GameMusicPlayer
extends GameSoundPlayer
```

Plays the game music track on repeat.

### Field Summary

#### Fields inherited from class com.example.project.services.sound.GameSoundPlayer

clip

### Constructor Summary

#### Constructors

##### Constructor

##### Description

[GameMusicPlayer\(\)](#)

Create new instance of game music player.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

void

[playGameMusicLoop\(\)](#)

play game music loop continuously.

#### Methods inherited from class com.example.project.services.sound.GameSoundPlayer

[mute](#), [replay](#), [unMute](#)

#### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Details

#### GameMusicPlayer

```
public GameMusicPlayer()
```

Create new instance of game music player.

### Method Details

## **playGameMusicLoop**

```
public void playGameMusicLoop()
```

play game music loop continuously.

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens.animations

## Class InfiniteFloatingAnimation

java.lang.Object  
com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation

```
public class InfiniteFloatingAnimation  
extends Object
```

Balatro style floating tile animation.

### Constructor Summary

#### Constructors

**Constructor**

**Description**

`InfiniteFloatingAnimation()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

`void`

`apply(javafx.scene.Node node, double intervalSeconds)`

apply to node.

#### Methods inherited from class java.lang.Object

`clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()`

### Constructor Details

#### InfiniteFloatingAnimation

```
public InfiniteFloatingAnimation()
```

### Method Details

#### apply

```
public void apply(javafx.scene.Node node,  
                  double intervalSeconds)
```

apply to node.

**Parameters:**

node - node.

intervalSeconds - time between changing directions.



**Module** com.example.project  
**Package** com.example.project.services.sound

## Class GameSoundPlayer

java.lang.Object<sup>2</sup>  
    com.example.project.services.sound.GameSoundPlayer

### Direct Known Subclasses:

GameMusicPlayer

---

```
public class GameSoundPlayer
extends Object2
```

plays an audio clip as a sound effect for the game.

## Field Summary

### Fields

#### Modifier and Type

#### Field

#### Description

protected final Clip<sup>2</sup>

clip

## Constructor Summary

### Constructors

#### Modifier

#### Constructor

#### Description

GameSoundPlayer(String<sup>2</sup> filePath)

creates a new instance of GameSoundPlayer

GameSoundPlayer(String<sup>2</sup> filePath, float gainAmount)

creates a new instance of GameSoundPlayer

protected

GameSoundPlayer(String<sup>2</sup> filePath, Logger logger)

protected

GameSoundPlayer(String<sup>2</sup> filePath, Logger logger, float gainAmount)

## Method Summary

### All Methods

### Instance Methods

### Concrete Methods

#### Modifier and Type

#### Method

#### Description

void

```
mute()
```

set clip volume to mute.

```
void
```

```
replay()
```

Reset clip and then play.

```
void
```

```
unMute()
```

Set clip to its default volume.

#### Methods inherited from class java.lang.Object

```
clone✉, equals✉, finalize✉, getClass✉, hashCode✉, notify✉, notifyAll✉, toString✉, wait✉, wait✉, wait✉
```

### Field Details

#### clip

```
protected final Clip✉ clip
```

### Constructor Details

#### GameSoundPlayer

```
public GameSoundPlayer(String✉ filePath)
```

creates a new instance of GameSoundPlayer

##### Parameters:

filePath - filepath to sound.

#### GameSoundPlayer

```
protected GameSoundPlayer(String✉ filePath,  
                           Logger logger)
```

#### GameSoundPlayer

```
protected GameSoundPlayer(String✉ filePath,  
                           Logger logger,  
                           float gainAmount)
```

#### GameSoundPlayer

```
public GameSoundPlayer(String✉ filePath,  
                           float gainAmount)
```

creates a new instance of GameSoundPlayer

##### Parameters:

filePath - filepath to sound.

gainAmount - volume to set the clip at

### Method Details

#### replay

```
public void replay()
```

Reset clip and then play.

## **mute**

```
public void mute()
```

set clip volume to mute.

## **unMute**

```
public void unMute()
```

Set clip to its default volume.

## **Uses of Class com.example.project.services.sound.GameMusicPlayer**

No usage of com.example.project.services.sound.GameMusicPlayer

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.services.sound.GameSoundPlayer

### Packages that use GameSoundPlayer

#### Package

#### Description

com.example.project.models.popups

com.example.project.models.tiles

com.example.project.services.sound

### Uses of GameSoundPlayer in com.example.project.models.popups

#### Fields in com.example.project.models.popups declared as GameSoundPlayer

##### Modifier and Type

##### Field

##### Description

protected final GameSoundPlayer

PopupModel.paperSoundPlayer

protected final GameSoundPlayer

PopupModel.reversePaperSoundPlayer

#### Methods in com.example.project.models.popups that return GameSoundPlayer

##### Modifier and Type

##### Method

##### Description

GameSoundPlayer

PopupModel.getPaperSoundPlayer()

gets paper sound player

GameSoundPlayer

PopupModel.getReversePaperSoundPlayer()

gets reversed paper sound player

### Uses of GameSoundPlayer in com.example.project.models.tiles

#### Fields in com.example.project.models.tiles declared as GameSoundPlayer

##### Modifier and Type

##### Field

##### Description

protected final GameSoundPlayer

TileModel.clackSoundPlayer

protected final GameSoundPlayer

TileModel.hoverSoundPlayer

## Methods in com.example.project.models.tiles that return GameSoundPlayer

### Modifier and Type

### Method

### Description

**GameSoundPlayer**

**TileModel.getClickSoundPlayer()**

gets the sound player that plays the click sound.

**GameSoundPlayer**

**TileModel.getHoverSoundPlayer()**

gets the hover sound player

## Uses of GameSoundPlayer in com.example.project.services.sound

## Subclasses of GameSoundPlayer in com.example.project.services.sound

### Modifier and Type

### Class

### Description

class

**GameMusicPlayer**

Plays the game music track on repeat.

Module com.example.project

## Package com.example.project.services.sound

package com.example.project.services.sound

### Related Packages

#### Package

#### Description

[com.example.project.services](#)

[com.example.project.services.shopItems](#)

[com.example.project.services.sqlite](#)

### Classes

#### Class

#### Description

[GameMusicPlayer](#)

Plays the game music track on repeat.

[GameSoundPlayer](#)

plays an audio clip as a sound effect for the game.

---

Copyright © 2025. All rights reserved.

## Hierarchy For Package com.example.project.services.sound

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#) ↗
  - [com.example.project.services.sound.GameSoundPlayer](#)
    - [com.example.project.services.sound.GameMusicPlayer](#)

## Uses of Package com.example.project.services.sound

### Packages that use com.example.project.services.sound

#### Package

#### Description

com.example.project.models.popups

com.example.project.models.tiles

com.example.project.services.sound

### Classes in com.example.project.services.sound used by com.example.project.models.popups

#### Class

#### Description

**GameSoundPlayer**

plays an audio clip as a sound effect for the game.

### Classes in com.example.project.services.sound used by com.example.project.models.tiles

#### Class

#### Description

**GameSoundPlayer**

plays an audio clip as a sound effect for the game.

### Classes in com.example.project.services.sound used by com.example.project.services.sound

#### Class

#### Description

**GameSoundPlayer**

plays an audio clip as a sound effect for the game.

**Module** com.example.project  
**Package** com.example.project.services.sqlite

## Class SQLiteConnection

java.lang.Object<sup>↳</sup>  
    com.example.project.services.sqlite.SQLiteConnection

### Direct Known Subclasses:

SQLiteDictionaryConnection, SQLiteUsersConnection

---

```
public abstract class SQLiteConnection
extends Object↳
```

Abstract SQLiteConnection class.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**SQLiteConnection()**

### Method Summary

#### All Methods

#### Instance Methods

#### Abstract Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

protected abstract String<sup>↳</sup>

**getDatabasePath()**

Each subclass provides its database path.

##### Connection<sup>↳</sup>

**getInstance()**

Gets the database connection from an existing readonly resource .db

protected abstract Connection<sup>↳</sup>

**getSQLiteInstance()**

protected abstract void

**setSQLiteInstance(Connection<sup>↳</sup> newInstance)**

#### Methods inherited from class java.lang.Object<sup>↳</sup>

clone<sup>↳</sup>, equals<sup>↳</sup>, finalize<sup>↳</sup>, getClass<sup>↳</sup>, hashCode<sup>↳</sup>, notify<sup>↳</sup>, notifyAll<sup>↳</sup>, toString<sup>↳</sup>, wait<sup>↳</sup>, wait<sup>↳</sup>, wait<sup>↳</sup>

### Constructor Details

#### SQLiteConnection

public SQLiteConnection()

## **Method Details**

### **getSQLLiteInstance**

```
protected abstract Connection2 getSQLLiteInstance()
```

### **setSQLLiteInstance**

```
protected abstract void setSQLLiteInstance(Connection2 newInstance)
```

### **getDatabasePath**

```
protected abstract String2 getDatabasePath()
```

Each subclass provides its database path.

### **getInstance**

```
public Connection2 getInstance()
```

Gets the database connection from an existing readonly resource .db

**Returns:**

returns the connection is already made or creates new one.

**Module** com.example.project

**Package** com.example.project.services.sqlite

## Class SQLiteDictionaryConnection

```
java.lang.Object
    com.example.project.services.sqlite.SQLiteConnection
        com.example.project.services.sqlite.SQLiteDictionaryConnection
```

```
public class SQLiteDictionaryConnection
extends SQLiteConnection
```

Connection to the dictionary.db

### Constructor Summary

#### Constructors

##### Constructor

##### Description

`SQLiteDictionaryConnection()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

`protected String`

`getDatabasePath()`

Each subclass provides its database path.

`protected Connection`

`getSQLiteInstance()`

`protected void`

`setSQLiteInstance(Connection`

#### Methods inherited from class com.example.project.services.sqlite.SQLiteConnection

`getInstance`

#### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Details

#### SQLiteDictionaryConnection

```
public SQLiteDictionaryConnection()
```

### Method Details

## **getSQLLiteInstance**

```
protected Connection2 getSQLLiteInstance()
```

**Specified by:**

[getSQLLiteInstance](#) in class [SQLiteConnection](#)

## **setSQLLiteInstance**

```
protected void setSQLLiteInstance(Connection2 newInstance)
```

**Specified by:**

[setSQLLiteInstance](#) in class [SQLiteConnection](#)

## **getDatabasePath**

```
protected String2 getDatabasePath()
```

**Description copied from class: [SQLiteConnection](#)**

Each subclass provides its database path.

**Specified by:**

[getDatabasePath](#) in class [SQLiteConnection](#)

**Module** com.example.project

**Package** com.example.project.services.sqlite

## Class SQLiteUsersConnection

```
java.lang.Object
    com.example.project.services.sqlite.SQLiteConnection
        com.example.project.services.sqlite.SQLiteUsersConnection
```

```
public class SQLiteUsersConnection
extends SQLiteConnection
```

Connection to the users.db

### Constructor Summary

#### Constructors

##### Constructor

##### Description

`SQLiteUsersConnection()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

`protected String`

`getDatabasePath()`

Each subclass provides its database path.

`protected Connection`

`getSQLiteInstance()`

`protected void`

`setSQLiteInstance(Connection`

#### Methods inherited from class com.example.project.services.sqlite.SQLiteConnection

`getInstance`

#### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Details

#### SQLiteUsersConnection

```
public SQLiteUsersConnection()
```

### Method Details

## getSQLLiteInstance

```
protected Connection2 getSQLLiteInstance()
```

### Specified by:

getSQLLiteInstance in class [SQLiteConnection](#)

## setSQLLiteInstance

```
protected void setSQLLiteInstance(Connection2 newInstance)
```

### Specified by:

setSQLLiteInstance in class [SQLiteConnection](#)

## getDatabasePath

```
protected String2 getDatabasePath()
```

### Description copied from class: [SQLiteConnection](#)

Each subclass provides its database path.

### Specified by:

getDatabasePath in class [SQLiteConnection](#)

## Uses of Class com.example.project.services.sqlite.SQLiteConnection

### Packages that use `SQLiteConnection`

#### Package

#### Description

[com.example.project.services.sqlite](#)

## Uses of `SQLiteConnection` in com.example.project.services.sqlite

### Subclasses of `SQLiteConnection` in com.example.project.services.sqlite

#### Modifier and Type

#### Class

#### Description

class

[SQLiteDictionaryConnection](#)

Connection to the dictionary.db

class

[SQLiteUsersConnection](#)

Connection to the users.db

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens.animations

## Class LevelScoreSequence

```
java.lang.Object
    com.example.project.controllers.gameScreens.animations.AnimationWrapper
        com.example.project.controllers.gameScreens.animations.LevelScoreSequence
```

```
public class LevelScoreSequence
extends AnimationWrapper
```

Animates each level tile popping up and the score text increasing for every tile in the word row.

### Field Summary

#### Fields inherited from

class com.example.project.controllers.gameScreens.animations.AnimationWrapper

sequentialAnimation

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**LevelScoreSequence**(List<LetterTileController> wordTileControllers, **LevelModel** levelModel, javafx.scene.control.Label comboCountLabel, javafx.scene.control.Label multiplierLabel)

Constructor.

### Method Summary

#### Methods inherited from

class com.example.project.controllers.gameScreens.animations.AnimationWrapper

getChildren, play, setOnFinished

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### LevelScoreSequence

```
public LevelScoreSequence(List<LetterTileController> wordTileControllers,
                           LevelModel levelModel,
                           javafx.scene.control.Label comboCountLabel,
                           javafx.scene.control.Label multiplierLabel)
```

Constructor.

##### Parameters:

wordTileControllers - tiles to animate.

levelModel - level model.

comboCountLabel - the text label to add the sum to.

multiplierLabel - the text label to add the multi to.



## Uses of Class

### **com.example.project.services.sqlite.SQLiteDictionaryConnection**

No usage of com.example.project.services.sqlite.SQLiteDictionaryConnection

---

Copyright © 2025. All rights reserved.

## **Uses of Class com.example.project.services.sqlite.SQLiteUsersConnection**

No usage of com.example.project.services.sqlite.SQLiteUsersConnection

---

Copyright © 2025. All rights reserved.

**Module** com.example.project  
**Package** com.example.project.services.sqlite.dAOs

## Class DictionaryDAO

java.lang.Object  
    com.example.project.services.sqlite.dAOs.DictionaryDAO

```
public class DictionaryDAO  
extends Object
```

The SQLite Dictionary. The connection returned from SQLiteDictionaryConnection().getInstance() is always the same.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**DictionaryDAO()**

Constructor for this class SQLLiteDictionary.

**DictionaryDAO(Connection<sup>2</sup> connection, Logger logger)**

constructor with injection for tests.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

**String<sup>2</sup>**

**getWordDefinition(String<sup>2</sup> wordToFind)**

gets the words definition.

**boolean**

**isWordInDictionary(String<sup>2</sup> wordToCheck)**

gets if word is in dictionary.

#### Methods inherited from class java.lang.Object<sup>2</sup>

**clone<sup>2</sup>, equals<sup>2</sup>, finalize<sup>2</sup>, getClass<sup>2</sup>, hashCode<sup>2</sup>, notify<sup>2</sup>, notifyAll<sup>2</sup>, toString<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>, wait<sup>2</sup>**

### Constructor Details

#### DictionaryDAO

```
public DictionaryDAO()
```

Constructor for this class SQLLiteDictionary.

#### DictionaryDAO

```
public DictionaryDAO(Connection2 connection,  
                      Logger logger)
```

constructor with injection for tests.

##### Parameters:

connection - mock connection.

logger - a mock logger.

## Method Details

### getWordDefinition

```
public String2 getWordDefinition(String2 wordToFind)
```

gets the words definition.

**Parameters:**

wordToFind - Word to get definition of.

**Returns:**

Returns the definition.

### isWordInDictionary

```
public boolean isWordInDictionary(String2 wordToCheck)
```

gets if word is in dictionary.

**Parameters:**

wordToCheck - Word to check.

**Returns:**

returns the boolean value indicating whether the word exists in our database.

**Module** com.example.project  
**Package** com.example.project.services.sqlite.dAOs

## Class UsersDAO

java.lang.Object  
    com.example.project.services.sqlite.dAOs.UsersDAO

```
public class UsersDAO  
extends Object
```

SQLite Users database. with a table `users` 2 columns. 'username', 'password'. Which are both defined as unique not null Strings in sqlite.

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

**UsersDAO()**

Constructor for this class SQLLiteDictionary.

protected

**UsersDAO(PasswordHasher hasher, Connection connection, Logger logger)**

**UsersDAO(Connection connection, Logger logger)**

Constructor with injection for unit tests.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

void

**addUser(User user)**

Adds user to the user.db.

boolean

**doesUserExist(String username)**

User

**getUser(String username)**

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

### Constructor Details

## UsersDAO

```
public UsersDAO()
```

Constructor for this class SQLiteDictionary.

## UsersDAO

```
protected UsersDAO(PasswordHasher hasher,  
                  Connection✉ connection,  
                  Logger logger)
```

## UsersDAO

```
public UsersDAO(Connection✉ connection,  
                  Logger logger)
```

Constructor with injection for unit tests.

**Parameters:**

connection - Connection

logger - logger.

## Method Details

### addUser

```
public void addUser(User user)
```

Adds user to the user.db. password will be hashed before storing to ensure greater security (no plain text passwords)

**Parameters:**

user - user to add.

### doesUserExist

```
public boolean doesUserExist(String✉ username)
```

**Parameters:**

username - username

**Returns:**

returns bool indicating whether use is in database already.

### getUser

```
public User getUser(String✉ username)
```

**Parameters:**

username - username.

**Returns:**

returns user with matching username.

## Uses of Class

### **com.example.project.services.sqlite.dAOs.DictionaryDAO**

No usage of com.example.project.services.sqlite.dAOs.DictionaryDAO

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.services.sqlite.dAOs.UsersDAO

### Packages that use UsersDAO

#### Package

#### Description

com.example.project.models.gameScreens

### Uses of UsersDAO in com.example.project.models.gameScreens

#### Constructors in com.example.project.models.gameScreens with parameters of type UsersDAO

##### Modifier

##### Constructor

##### Description

`LoginModel(Session session, UsersDAO dao, PasswordHasher passwordHasher)`

constructor.

Module com.example.project

## Package com.example.project.services.sqlite.dAOs

package com.example.project.services.sqlite.dAOs

### Related Packages

#### Package

#### Description

com.example.project.services.sqlite

### Classes

#### Class

#### Description

**DictionaryDAO**

The SQLite Dictionary.

**UsersDAO**

SQLite Users database.

## Hierarchy For Package com.example.project.services.sqlite.dAOs

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
  - [com.example.project.services.sqlite.dAOs.DictionaryDAO](#)
  - [com.example.project.services.sqlite.dAOs.UsersDAO](#)

## Uses of Package `com.example.project.services.sqlite.dAOs`

### Packages that use `com.example.project.services.sqlite.dAOs`

#### Package

#### Description

`com.example.project.models.gameScreens`

### Classes in `com.example.project.services.sqlite.dAOs` used by `com.example.project.models.gameScreens`

#### Class

#### Description

`UsersDAO`

SQLite Users database.

Module com.example.project

## Package com.example.project.services.sqlite

package com.example.project.services.sqlite

### Related Packages

#### Package

#### Description

[com.example.project.services](#)

[com.example.project.services.sqlite.dAOs](#)

[com.example.project.services.shopItems](#)

[com.example.project.services.sound](#)

### Classes

#### Class

#### Description

[SQLiteConnection](#)

Abstract SQLiteConnection class.

[SQLiteDictionaryConnection](#)

Connection to the dictionary.db

[SQLiteUsersConnection](#)

Connection to the users.db

---

Copyright © 2025. All rights reserved.

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens.animations

## Class ScoreTimeline

java.lang.Object  
com.example.project.controllers.gameScreens.animations.ScoreTimeline

```
public class ScoreTimeline  
extends Object
```

Class for total score counter animation

### Constructor Summary

#### Constructors

**Constructor**

**Description**

`ScoreTimeline()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

`javafx.animation.Timeline`

`animateTotalScore(int startScore, int endScore, javafx.scene.control.Label currentScoreLabel,  
long totalDurationMillis)`

#### Methods inherited from class java.lang.Object

`clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()`

### Constructor Details

#### ScoreTimeline

```
public ScoreTimeline()
```

### Method Details

#### animateTotalScore

```
public javafx.animation.Timeline animateTotalScore(int startScore,  
                                                 int endScore,  
                                                 javafx.scene.control.Label currentScoreLabel,  
                                                 long totalDurationMillis)
```

**Parameters:**

`startScore` - int from current total score

`endScore` - int from calculated current score

`currentScoreLabel` - Label for total score

`totalDurationMillis` - total duration

**Returns:**

timeline of total score counter

## Hierarchy For Package com.example.project.services.sqlite

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
- [com.example.project.services.sqlite.SQLiteConnection](#)
  - [com.example.project.services.sqlite.SQLiteDictionaryConnection](#)
  - [com.example.project.services.sqlite.SQLiteUsersConnection](#)

## Uses of Package com.example.project.services.sqlite

### Packages that use com.example.project.services.sqlite

#### Package

#### Description

com.example.project.services.sqlite

### Classes in com.example.project.services.sqlite used by com.example.project.services.sqlite

#### Class

#### Description

**SQLiteDatabase**

Abstract SQLiteDatabase class.

## Module com.example.project

module com.example.project

project dependencies.

### Packages

#### Exports

##### Package

##### Description

[com.example.project](#)

[com.example.project.controllers](#)

[com.example.project.controllers.gameScreens](#)

[com.example.project.controllers.gameScreens.animations](#)

[com.example.project.controllers.popupControllers](#)

[com.example.project.controllers.tiles](#)

[com.example.project.models](#)

[com.example.project.models.gameScreens](#)

[com.example.project.models.popups](#)

[com.example.project.models.tileGroups](#)

[com.example.project.models.tiles](#)

[com.example.project.services](#)

[com.example.project.services.shopItems](#)

[com.example.project.services.sound](#)

[com.example.project.services.sqlite](#)

[com.example.project.services.sqlite.dAOs](#)

# Constant Field Values

## Contents

com.example.\*

### com.example.\*

#### com.example.project.Application

##### Modifier and Type

##### Constant Field

##### Value

public static final int

##### HEIGHT

700

public static final String

##### TITLE

"WordPlay"

public static final int

##### WIDTH

1300

#### com.example.project.models.tiles.TileModel

##### Modifier and Type

##### Constant Field

##### Value

public static final int

##### CORNER\_RADIUS

20

# JavaDoc Help

- Navigation:
  - Search
- Kinds of Pages:
  - Module
  - Package
  - Class or Interface
  - Other Files
  - Use
  - Tree (Class Hierarchy)
  - Constant Field Values
  - All Packages
  - All Classes and Interfaces
  - Index

## Navigation

Starting from the [Overview](#) page, you can browse the documentation using the links in each page, and in the navigation bar at the top of each page. The [Index](#) and [Search](#) box allow you to navigate to specific declarations and summary pages, including: [All Packages](#), [All Classes and Interfaces](#)

## Search

You can search for definitions of modules, packages, types, fields, methods, system properties and other terms defined in the API. These items can be searched using part or all of the name, optionally using "camelCase" abbreviations, or multiple search terms separated by whitespace. Some examples:

- "j.l.obj" matches "java.lang.Object"
- "InpStr" matches "java.io.InputStream"
- "math exact long" matches "java.lang.Math.absExact(long)"

Refer to the [Javadoc Search Specification](#) for a full description of search features.

## Kinds of Pages

The following sections describe the different kinds of pages in this collection.

### Module

Each module has a page that contains a list of its packages, dependencies on other modules, and services, with a summary for each. These pages may contain the following categories:

- Packages
- Modules
- Services

### Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. These pages may contain the following categories:

- Interfaces
- Classes
- Enum Classes
- Exception Classes
- Annotation Interfaces

### Class or Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a declaration and description, member summary tables, and detailed member descriptions. Entries in each of these sections are omitted if they are empty or not applicable.

- Class Inheritance Diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
- Class or Interface Declaration
- Class or Interface Description
- Nested Class Summary
- Enum Constant Summary
- Field Summary
- Property Summary
- Constructor Summary
- Method Summary
- Required Element Summary
- Optional Element Summary
- Enum Constant Details
- Field Details
- Property Details
- Constructor Details
- Method Details
- Element Details

**Note:** Annotation interfaces have required and optional elements, but not methods. Only enum classes have enum constants. The components of a record class are displayed as part of the declaration of the record class. Properties are a feature of JavaFX.

The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

## Other Files

Packages and modules may contain pages with additional information related to the declarations nearby.

## Use

Each documented package, class and interface has its own Use page. This page describes what packages, classes, methods, constructors and fields use any part of the given class or package. Given a class or interface A, its Use page includes subclasses of A, fields declared as A, methods that return A, and methods and constructors with parameters of type A. You can access this page by first going to the package, class or interface, then clicking on the USE link in the navigation bar.

## Tree (Class Hierarchy)

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. Classes are organized by inheritance structure starting with `java.lang.Object`. Interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on TREE displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking on TREE displays the hierarchy for only that package.

## Constant Field Values

The [Constant Field Values](#) page lists the static final fields and their values.

## All Packages

The [All Packages](#) page contains an alphabetic index of all packages contained in the documentation.

## All Classes and Interfaces

The [All Classes and Interfaces](#) page contains an alphabetic index of all classes and interfaces contained in the documentation, including annotation interfaces, enum classes, and record classes.

## Index

The [Index](#) contains an alphabetic index of all classes, interfaces, constructors, methods, and fields in the documentation, as well as summary pages such as [All Packages](#), [All Classes and Interfaces](#).

---

*This help file applies to API documentation generated by the standard doclet.*

---

# Index

A B C D E F G H I L M N O P R S T U V W

All Classes and Interfaces | All Packages | Constant Field Values

## A

**addMoney(int)** - Method in class com.example.project.services.Session  
    Adds money to the player's balance.

**addTileToRack(LetterTileModel)** - Method in class com.example.project.models.gameScreens.LevelModel

**addTileToRedrawWindow(LetterTileModel)** - Method in class com.example.project.models.gameScreens.LevelModel

**addTileToWordWindow(LetterTileModel)** - Method in class com.example.project.models.gameScreens.LevelModel

**addToCombo(LetterTileModel)** - Method in class com.example.project.models.gameScreens.LevelModel  
    add combo sum and multiCombo

**addUpgrade(UpgradeTileModel)** - Method in class com.example.project.services.Session  
    Adds an upgrade tile to the player's collection.

**addUser(User)** - Method in class com.example.project.services.sqlite.dAOs.UsersDAO  
    Adds user to the user.db.

**animateTotalScore(int, int, Label, long)** - Method in class com.example.project.controllers.gameScreens.animations.ScoreTimeline

**AnimationWrapper** - Class in com.example.project.controllers.gameScreens.animations  
    Animation wrapper class for our projects animations that get used.

**AnimationWrapper()** - Constructor for class com.example.project.controllers.gameScreens.animations.AnimationWrapper

**Application** - Class in com.example.project  
    The entry point of the application.

**Application()** - Constructor for class com.example.project.Application

**apply(Node, double)** - Method in class com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation  
    apply to node.

**applyReward(DailyRewardType)** - Static method in class com.example.project.models.gameScreens.DailyRewardModel  
    Applies the effect of the given reward to the session and marks it as claimed.

## B

**bind(EmptyTileSlotModel)** - Method in class com.example.project.controllers.tiles.EmptyTileSlotController

**bind(LetterTileModel)** - Method in class com.example.project.controllers.tiles.LetterTileController

**bind(UpgradeTileModel)** - Method in class com.example.project.controllers.tiles.UpgradeTileController

**bind(T)** - Method in class com.example.project.controllers.tiles.TileController  
    Binds tile to the controller as its model.

**bindPersistentInfo()** - Method in class com.example.project.controllers.gameScreens.SidebarController  
    Binds things used in multiple screens, money, redraws plays.

**braceletEffect()** - Static method in class com.example.project.models.tiles.UpgradeEffects  
    Add +1 to score multiplier for every consecutive letter alphabetical order.

**build()** - Method in class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder  
    build upgrade

**buttonEffect()** - Static method in class com.example.project.models.tiles.UpgradeEffects  
    adds 5 to the word score for every x in the tile rack.

## C

**calcTotalWordScore()** - Method in class com.example.project.models.gameScreens.LevelModel

**canPurchase(UpgradeTileModel)** - Method in class com.example.project.models.gameScreens.ShopModel  
    Determines whether the player can afford to purchase the specified upgrade tile.

**ClackSoundPlayer** - Variable in class com.example.project.models.tiles.TileModel

**clearLogs()** - Method in class com.example.project.services.Logger  
    Clears captured logs use in tests teardown.

**clip** - Variable in class com.example.project.services.sound.GameSoundPlayer

**coinEffect()** - Static method in class com.example.project.models.tiles.UpgradeEffects  
    20% Chance the total word score is doubled

com.example.project - module com.example.project  
    project dependencies.

com.example.project - package com.example.project

com.example.project.controllers - package com.example.project.controllers

com.example.project.controllers.gameScreens - package com.example.project.controllers.gameScreens  
com.example.project.controllers.gameScreens.animations - package com.example.project.controllers.gameScreens.animations  
com.example.project.controllers.popupControllers - package com.example.project.controllers.popupControllers  
com.example.project.controllers.tiles - package com.example.project.controllers.tiles  
com.example.project.models - package com.example.project.models  
com.example.project.models.gameScreens - package com.example.project.models.gameScreens  
com.example.project.models.popups - package com.example.project.models.popups  
com.example.project.models.tileGroups - package com.example.project.models.tileGroups  
com.example.project.models.tiles - package com.example.project.models.tiles  
com.example.project.services - package com.example.project.services  
com.example.project.services.shopItems - package com.example.project.services.shopItems  
com.example.project.services.sound - package com.example.project.services.sound  
com.example.project.services.sqlite - package com.example.project.services.sqlite  
com.example.project.services.sqlite.dAOs - package com.example.project.services.sqlite.dAOs

**container** - Variable in class com.example.project.models.tileGroups.TileGroup

**controller()** - Method in record class com.example.project.controllers.gameScreens.GameScreenController.SidebarLoaded  
Returns the value of the controller record component.

**CORNER\_RADIUS** - Static variable in class com.example.project.models.tiles.TileModel  
corner radius (rounded corners).

**cost(double)** - Method in class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder  
upgrade cost builder

**createLetterTileController(LetterTileModel)** - Method in class com.example.project.controllers.tiles.TileControllerFactory

**createPopupController(P)** - Method in class com.example.project.services.PopupLoader  
create popup controller.

**createTileController(T, Class<C>)** - Method in class com.example.project.controllers.tiles.TileControllerFactory  
Returns controller of type.

**currentShopItemsProperty()** - Method in class com.example.project.models.gameScreens.ShopModel  
Current items in the shop row that you can buy.

## D

**DAILY\_REWARD** - Enum constant in enum class com.example.project.services.GameScenes  
Daily Reward Spinner Screen.

**Daily\_Reward\_Won\_1Dollar** - Enum constant in enum class com.example.project.models.gameScreens.DailyRewardType  
Player wins \$1

**Daily\_Reward\_Won\_5Dollars** - Enum constant in enum class com.example.project.models.gameScreens.DailyRewardType  
Player wins \$5

**Daily\_Reward\_Won\_NOTHING** - Enum constant in enum class com.example.project.models.gameScreens.DailyRewardType  
Player wins nothing

**DailyRewardController** - Class in com.example.project.controllers.gameScreens  
Controller for the daily reward popup screen.

**DailyRewardController()** - Constructor for class com.example.project.controllers.gameScreens.DailyRewardController

**DailyRewardModel** - Class in com.example.project.models.gameScreens  
Model for handling daily reward logic and state.

**DailyRewardModel()** - Constructor for class com.example.project.models.gameScreens.DailyRewardModel

**DailyRewardType** - Enum Class in com.example.project.models.gameScreens  
Enum representing the possible daily reward outcomes.

**DefinitionController** - Class in com.example.project.controllers.popupControllers  
Controller for definition popup

**DefinitionController()** - Constructor for class com.example.project.controllers.popupControllers.DefinitionController

**DefinitionPopup** - Class in com.example.project.models.popups  
Represents Definition window called when a word is played

**DefinitionPopup()** - Constructor for class com.example.project.models.popups.DefinitionPopup

**definitionProperty()** - Method in class com.example.project.models.popups.DefinitionPopup

**description(String)** - Method in class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder  
upgrade description builder

**diceEffect()** - Static method in class com.example.project.models.tiles.UpgradeEffects  
Value is doubled for a random letter in the played word

**DictionaryDAO** - Class in com.example.project.services.sqlite.dAOs

The SQLite Dictionary.

**DictionaryDAO()** - Constructor for class com.example.project.services.sqlite.dAOs.DictionaryDAO

Constructor for this class SQLLiteDictionary.

**DictionaryDAO(Connection, Logger)** - Constructor for class com.example.project.services.sqlite.dAOs.DictionaryDAO

constructor with injection for tests.

**doesPasswordMatch(String, String)** - Method in class com.example.project.services.PasswordHasher

Verifies a candidate password against a hashed password.

**doesUserExist(String)** - Method in class com.example.project.services.sqlite.dAOs.UsersDAO

**drawRandomTile()** - Method in class com.example.project.models.tiles.ScrabbleTileProvider

## E

**EmptyTileSlotController** - Class in com.example.project.controllers.tiles

Empty Tile controller for EmptyTile view.

**EmptyTileSlotController()** - Constructor for class com.example.project.controllers.tiles.EmptyTileSlotController

**EmptyTileSlotModel** - Class in com.example.project.models.tiles

An empty tile slot model.

**EmptyTileSlotModel()** - Constructor for class com.example.project.models.tiles.EmptyTileSlotModel

**equals(Object)** - Method in record class com.example.project.controllers.gameScreens.GameScreenController SidebarLoaded

Indicates whether some other object is "equal to" this one.

**equals(Object)** - Method in record class com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData

Indicates whether some other object is "equal to" this one.

## F

**frequency()** - Method in record class com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData

Returns the value of the frequency record component.

**FXMLPageLoader** - Class in com.example.project.services

FXML Loader wrapper class for the project to load the game screen FXML pages.

**FXMLPageLoader()** - Constructor for class com.example.project.services.FXMLPageLoader

constructor.

## G

**GameMusicPlayer** - Class in com.example.project.services.sound

Plays the game music track on repeat.

**GameMusicPlayer()** - Constructor for class com.example.project.services.sound.GameMusicPlayer

Create new instance of game music player.

**GameScenes** - Enum Class in com.example.project.services

Game Scenes enum.

**GameScreenController** - Class in com.example.project.controllers.gameScreens

Game screen controller that has some startup on screen thing todo.

**GameScreenController()** - Constructor for class com.example.project.controllers.gameScreens.GameScreenController

**GameScreenController.SidebarLoaded** - Record Class in com.example.project.controllers.gameScreens

**GameScreenModel** - Class in com.example.project.models.gameScreens

Game Screen Model.

**GameScreenModel(Session)** - Constructor for class com.example.project.models.gameScreens.GameScreenModel

**GameSoundPlayer** - Class in com.example.project.services.sound

plays an audio clip as a sound effect for the game.

**GameSoundPlayer(String)** - Constructor for class com.example.project.services.sound.GameSoundPlayer

creates a new instance of GameSoundPlayer

**GameSoundPlayer(String, float)** - Constructor for class com.example.project.services.sound.GameSoundPlayer

creates a new instance of GameSoundPlayer

**GameSoundPlayer(String, Logger)** - Constructor for class com.example.project.services.sound.GameSoundPlayer

**GameSoundPlayer(String, Logger, float)** - Constructor for class com.example.project.services.sound.GameSoundPlayer

**getAbilityImagePath()** - Method in class com.example.project.models.tiles.UpgradeTileModel

gets upgrade tile image path.

**getChildren()** - Method in class com.example.project.controllers.gameScreens.animations.AnimationWrapper

returns animations children.

**getClickSoundPlayer()** - Method in class com.example.project.models.tiles.TileModel

gets the sound player that plays the click sound.

**getCogIcon()** - Static method in class com.example.project.services.SVGIcons

gets a cog icon.

**getComboLabel()** - Method in class com.example.project.controllers.gameScreens.SidebarController

Returns combo label.

**getController()** - Method in class com.example.project.services.FXMLPageLoader

**getController()** - Method in interface com.example.project.services.PageLoader  
Get the .fxml pages controller

**getControllers()** - Method in class com.example.project.models.tileGroups.TileGroup  
Gets the groups tile controllers.

**getCost()** - Method in class com.example.project.models.tiles.UpgradeTileModel  
gets the cost

**getCurrentPlays()** - Method in class com.example.project.services.Session  
gets the current plays.

**getCurrentRedraws()** - Method in class com.example.project.services.Session  
gets the redraws property.

**getCurrentScoreLabel()** - Method in class com.example.project.controllers.gameScreens.SidebarController  
current score label.

**getCurrentWord()** - Method in class com.example.project.models.gameScreens.LevelModel  
Gets the current word formed by tiles in the word area

**getDatabasePath()** - Method in class com.example.project.services.sqlite.SQLiteConnection  
Each subclass provides its database path.

**getDatabasePath()** - Method in class com.example.project.services.sqlite.SQLiteDictionaryConnection

**getDatabasePath()** - Method in class com.example.project.services.sqlite.SQLiteUsersConnection

**getDescription()** - Method in class com.example.project.models.tiles.UpgradeTileModel  
gets the description.

**getErrorLogs()** - Method in class com.example.project.services.Logger

**getFXMLPath()** - Method in class com.example.project.models.popups.DefinitionPopup

**getFXMLPath()** - Method in class com.example.project.models.popups.PopupModel  
gets FXML path

**getFXMLPath()** - Method in class com.example.project.models.tiles.EmptyTileSlotModel

**getFXMLPath()** - Method in class com.example.project.models.tiles.LetterTileModel

**getFXMLPath()** - Method in class com.example.project.models.tiles.TileModel  
gets the fxml path.

**getFXMLPath()** - Method in class com.example.project.models.tiles.UpgradeTileModel

**getHandSize()** - Method in class com.example.project.services.Session  
gets hand size.

**getHighscore()** - Method in class com.example.project.models.User  
get highscore.

**getHoverSoundPlayer()** - Method in class com.example.project.models.tiles.TileModel  
gets the hover sound player

**getInstance()** - Static method in class com.example.project.services.SceneManager

**getInstance()** - Static method in class com.example.project.services.Session  
Gets session.

**getInstance()** - Method in class com.example.project.services.sqlite.SQLiteConnection  
Gets the database connection from an existing readonly resource .db

**getIsDefinitionActive()** - Method in class com.example.project.models.popups.DefinitionPopup

**getIsRedrawActive()** - Method in class com.example.project.models.gameScreens.LevelModel  
gets the redraw property.

**getLastRewardDate()** - Method in class com.example.project.services.Session  
Gets the last date the player claimed their daily reward.

**getLetter()** - Method in class com.example.project.models.tiles.LetterTileModel

**getLetterTileSlotProperty()** - Method in class com.example.project.models.tiles.EmptyTileSlotModel

**getLevelModel()** - Method in class com.example.project.services.Session

**getLevelRequirement()** - Method in class com.example.project.services.Session

**getLevelsBeaten()** - Method in class com.example.project.services.Session

**getLogs()** - Method in class com.example.project.services.Logger

**getModel()** - Method in class com.example.project.controllers.popupControllers.DefinitionController

**getModel()** - Method in class com.example.project.controllers.popupControllers.PopupController  
gets the popup model.

**getModel()** - Method in class com.example.project.controllers.tiles.LetterTileController

**getModel()** - Method in class com.example.project.controllers.tiles.TileController  
gets the tile model.

**getMoneyProperty()** - Method in class com.example.project.services.Session  
Returns the read-only money property for binding to UI components.

**getmultiplierLabel()** - Method in class com.example.project.controllers.gameScreens.SidebarController

word score multiplier label.

**getName()** - Method in class com.example.project.models.tiles.UpgradeTileModel  
gets the name.

**getPaperSoundPlayer()** - Method in class com.example.project.models.popups.PopupModel  
gets paper sound player

**getPassword()** - Method in class com.example.project.models.User  
get password.

**getPlayersCurrentPoints()** - Method in class com.example.project.models.gameScreens.LevelModel

**getPlayersTotalPoints()** - Method in class com.example.project.models.gameScreens.LevelModel

**getRandomUpgradeTile()** - Static method in class com.example.project.services.shopItems.UpgradeTiles  
Gets random upgrade tile.

**getRawPointsLabel()** - Method in class com.example.project.controllers.gameScreens SidebarController  
raw points label to be multiplied by multiplier.

**getRedrawWindowSize()** - Method in class com.example.project.services.Session  
gets redraw window size (number of slots)

**getRedrawWindowTilesProperty()** - Method in class com.example.project.models.gameScreens.LevelModel

**getReversePaperSoundPlayer()** - Method in class com.example.project.models.popups.PopupModel  
gets reversed paper sound player

**getRoot()** - Method in class com.example.project.controllers.tiles.EmptyTileSlotController

**getRoot()** - Method in class com.example.project.controllers.tiles.LetterTileController

**getRoot()** - Method in class com.example.project.controllers.tiles.TileController  
root ui node.

**getRoot()** - Method in class com.example.project.controllers.tiles.UpgradeTileController

**getSQLiteInstance()** - Method in class com.example.project.services.sqlite.SQLiteConnection

**getSQLiteInstance()** - Method in class com.example.project.services.sqlite.SQLiteDictionaryConnection

**getSQLiteInstance()** - Method in class com.example.project.services.sqlite.SQLiteUsersConnection

**getStack()** - Method in class com.example.project.controllers.popupControllers.DefinitionController

**getStack()** - Method in class com.example.project.controllers.popupControllers.PopupController  
root ui node.

**getTile(int)** - Static method in class com.example.project.services.shopItems.UpgradeTiles  
Gets the upgrade tile at index.

**getTileRackTilesProperty()** - Method in class com.example.project.models.gameScreens.LevelModel

**getTileScoreSoundPlayer()** - Method in class com.example.project.models.gameScreens.LevelModel  
Gets the tile score sound effect player.

**getUpgradeEffect()** - Method in class com.example.project.models.tiles.UpgradeTileModel  
gets the effect for this tile.

**getUpgradeTilesProperty()** - Method in class com.example.project.services.Session  
gets upgrade tile property

**getUser(String)** - Method in class com.example.project.services.sqlite.dAOs.UsersDAO

**getUsername()** - Method in class com.example.project.models.User  
get username.

**getValue()** - Method in class com.example.project.models.tiles.LetterTileModel

**getValue(Character)** - Method in class com.example.project.models.tiles.ScrabbleTileProvider  
Returns Integer of the letter according to scrabble

**getWelcomeTextProperty()** - Method in class com.example.project.models.gameScreens.LoginModel  
Gets the info text property.

**getWordDefinition(String)** - Method in class com.example.project.services.sqlite.dAOs.DictionaryDAO  
gets the words definition.

**getWordWindowSize()** - Method in class com.example.project.services.Session  
gets word size.

**getWordWindowTilesProperty()** - Method in class com.example.project.models.gameScreens.LevelModel

**glassesEffect()** - Static method in class com.example.project.models.tiles.UpgradeEffects  
Adds +2 to the score multiplier for every identical pair of letters next to each other

## H

**hasClaimedRewardToday()** - Method in class com.example.project.services.Session  
Checks if the player already claimed today's reward.

**hashCode()** - Method in record class com.example.project.controllers.gameScreens.GameScreenController SidebarLoaded  
Returns a hash code value for this object.

**hashCode()** - Method in record class com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData  
Returns a hash code value for this object.

**hashPassword(String)** - Method in class com.example.project.services.PasswordHasher

Hashes a raw password using BCrypt.

**hasLost()** - Method in class com.example.project.models.gameScreens.LevelModel  
true if player has lost

**hasWon()** - Method in class com.example.project.models.gameScreens.LevelModel  
returns true if player has won.

**HEIGHT** - Static variable in class com.example.project.Application

**hoverSoundPlayer** - Variable in class com.example.project.models.tiles.TileModel

## I

**imagePath(String)** - Method in class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder  
upgrade image path builder

**imageView** - Variable in class com.example.project.controllers.tiles.UpgradeTileController

**InfiniteFloatingAnimation** - Class in com.example.project.controllers.gameScreens.animations  
Balatro style floating tile animation.

**InfiniteFloatingAnimation()** - Constructor for class  
com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation

**initialise(RootLayoutController, PageLoader)** - Method in class com.example.project.services.SceneManager  
loads the fxml of all the scenes on launch to switch to via scene manager.

**initialize()** - Method in class com.example.project.controllers.gameScreens.DailyRewardController  
Initialises the daily reward view by resetting the reward label.

**initialize()** - Method in class com.example.project.controllers.gameScreens.LevelController

**initialize()** - Method in class com.example.project.controllers.gameScreens.LoginController  
FXML initialise function called once when the .fxml is loaded on application launch.

**initialize()** - Method in class com.example.project.controllers.gameScreens.MainMenuController  
FXML initialise function called once when the .fxml is loaded.

**initialize()** - Method in class com.example.project.controllers.gameScreens.ShopController  
FXML initialise function called once when the .fxml is loaded on application launch.

**initialize(DefinitionPopup)** - Method in class com.example.project.controllers.popupControllers.DefinitionController

**initialize(P)** - Method in class com.example.project.controllers.popupControllers.PopupController  
sets elements of the ui

**isCurrentWordValid()** - Method in class com.example.project.models.gameScreens.LevelModel  
returns true if word is in dictionary.

**isWordInDictionary(String)** - Method in class com.example.project.services.sqlite.dAOs.DictionaryDAO  
gets if word is in dictionary.

## L

**LetterTileController** - Class in com.example.project.controllers.tiles  
Controls the layout of the letterUIModel.

**LetterTileController()** - Constructor for class com.example.project.controllers.tiles.LetterTileController

**LetterTileGroup** - Class in com.example.project.models.tileGroups  
tile group that observes an observable list and updates the ui Tile Controller's nodes into their EmptyTileSlotController nodes.

**LetterTileGroup(int, Pane, ReadOnlyListProperty<LetterTileModel>, Consumer<LetterTileController>)** - Constructor for  
class com.example.project.models.tileGroups.LetterTileGroup  
Constructor

**LetterTileGroup(int, Pane, ReadOnlyListProperty<LetterTileModel>, Consumer<LetterTileController>, List<Runnable>)** -  
Constructor for class com.example.project.models.tileGroups.LetterTileGroup  
Constructor

**LetterTileGroup(int, Pane, ReadOnlyListProperty<LetterTileModel>, Consumer<LetterTileController>, List<Runnable>, TileControllerFactory)** - Constructor for class com.example.project.models.tileGroups.LetterTileGroup

**LetterTileModel** - Class in com.example.project.models.tiles  
Represents the letter tiles the player plays holds, redraws.

**LetterTileModel(Character)** - Constructor for class com.example.project.models.tiles.LetterTileModel

**LEVEL** - Enum constant in enum class com.example.project.services.GameScenes  
level Screen

**LevelController** - Class in com.example.project.controllers.gameScreens  
Controller for the level screen.

**LevelController()** - Constructor for class com.example.project.controllers.gameScreens.LevelController  
Constructor only called once each time application opened.

**LevelController(LevelModel)** - Constructor for class com.example.project.controllers.gameScreens.LevelController

**LevelModel** - Class in com.example.project.models.gameScreens  
Represents the level model.

**LevelModel(Session)** - Constructor for class com.example.project.models.gameScreens.LevelModel  
Default constructor.

**LevelModel(Session, Logger)** - Constructor for class com.example.project.models.gameScreens.LevelModel

**LevelScoreSequence** - Class in com.example.project.controllers.gameScreens.animations

Animates each level tile popping up and the score text increasing for every tile in the word row.

**LevelScoreSequence(List<LetterTileController>, LevelModel, Label, Label)** - Constructor for class com.example.project.controllers.gameScreens.animations.LevelScoreSequence  
Constructor.

**load(String)** - Method in class com.example.project.services.FXMLPageLoader

**load(String)** - Method in interface com.example.project.services.PageLoader

Load method to load fxml page.

**loadSidebar()** - Method in class com.example.project.controllers.gameScreens.GameScreenController

**logError(String)** - Method in class com.example.project.services.Logger

Logs an error message to standard error.

**logger** - Variable in class com.example.project.controllers.gameScreens.GameScreenController

**logger** - Variable in class com.example.project.models.gameScreens.GameScreenModel

**Logger** - Class in com.example.project.services

com.example.project.services.Logger for Application and to show up in test and build CI workflow.

**Logger()** - Constructor for class com.example.project.services.Logger  
Default constructor writes to console does not capture.

**Logger(ByteArrayOutputStream, ByteArrayOutputStream)** - Constructor for class com.example.project.services.Logger  
Constructor for unit tests.

**Logger(ByteArrayOutputStream, ByteArrayOutputStream, PrintStream, PrintStream)** - Constructor for class com.example.project.services.Logger  
Constructor for unit tests.

**LOGIN** - Enum constant in enum class com.example.project.services.GameScenes

Login Screen.

**LoginController** - Class in com.example.project.controllers.gameScreens

Controller for the login scene.

**LoginController()** - Constructor for class com.example.project.controllers.gameScreens.LoginController  
No arg constructor.

**LoginModel** - Class in com.example.project.models.gameScreens

Login model class.

**LoginModel(Session, UsersDAO, PasswordHasher)** - Constructor for class com.example.project.models.gameScreens.LoginModel  
constructor.

**logMessage(String)** - Method in class com.example.project.services.Logger  
Log message to System.out.

## M

**main(String[])** - Static method in class com.example.project.Application

**MAINMENU** - Enum constant in enum class com.example.project.services.GameScenes

Main Menu Screen.

**MainMenuController** - Class in com.example.project.controllers.gameScreens  
controller for the main menu.

**MainMenuController()** - Constructor for class com.example.project.controllers.gameScreens.MainMenuController  
No arg constructor.

**MainMenuModel** - Class in com.example.project.models.gameScreens  
main menu model class

**MainMenuModel(Session)** - Constructor for class com.example.project.models.gameScreens.MainMenuModel  
constructor.

**mirrorEffect()** - Static method in class com.example.project.models.tiles.UpgradeEffects  
If played word is a palindrome, doubles word score.

**model** - Variable in class com.example.project.controllers.popupControllers.PopupController

**model** - Variable in class com.example.project.controllers.tiles.TileController

**mute()** - Method in class com.example.project.services.sound.GameSoundPlayer  
set clip volume to mute.

## N

**name(String)** - Method in class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder  
upgrade name builder

**node()** - Method in record class com.example.project.controllers.gameScreens.GameScreenController.SidebarLoaded  
Returns the value of the node record component.

## O

**onClickAction** - Variable in class com.example.project.models.tileGroups.TileGroup

**onConfirmRedrawButton()** - Method in class com.example.project.controllers.gameScreens.LevelController  
Handle redraw confirm button.

**onDailyRewardClicked()** - Method in class com.example.project.controllers.gameScreens.MainMenuController

**onFastForwardClick()** - Method in class com.example.project.controllers.gameScreens.MainMenuController

**onLoginButtonClick()** - Method in class com.example.project.controllers.gameScreens.LoginController

**onLoginClicked(String, String)** - Method in class com.example.project.models.gameScreens.LoginModel

**onLogoutButtonClick()** - Method in class com.example.project.controllers.gameScreens.MainMenuController

**onLogoutClicked()** - Method in class com.example.project.models.gameScreens.MainMenuModel  
switches to the login screen

**onLostLevel()** - Method in class com.example.project.models.gameScreens.LevelModel  
Called when the level has been lost.

**onlyShopShopInfo()** - Method in class com.example.project.controllers.gameScreens.SidebarController  
Only show things relevant to upgrades or the shop.

**onNextLevelPressed()** - Method in class com.example.project.controllers.gameScreens.ShopController

**onNextLevelPressed()** - Method in class com.example.project.models.gameScreens.ShopModel  
exists shop and increments level requirement for the next level.

**onPlayButton()** - Method in class com.example.project.controllers.gameScreens.LevelController  
Handle play button

**onRedrawButton()** - Method in class com.example.project.controllers.gameScreens.LevelController  
redraw button opens or cancels the redraw.

**onSceneChangedToThis()** - Method in class com.example.project.controllers.gameScreens.DailyRewardController

**onSceneChangedToThis()** - Method in class com.example.project.controllers.gameScreens.GameScreenController  
Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

**onSceneChangedToThis()** - Method in class com.example.project.controllers.gameScreens.LevelController

**onSceneChangedToThis()** - Method in class com.example.project.controllers.gameScreens.LoginController

**onSceneChangedToThis()** - Method in class com.example.project.controllers.gameScreens.MainMenuController

**onSceneChangedToThis()** - Method in class com.example.project.controllers.gameScreens.ShopController

**onSignupButtonClick()** - Method in class com.example.project.controllers.gameScreens.LoginController

**onSignUpClicked(String, String)** - Method in class com.example.project.models.gameScreens.LoginModel  
Adds user to database.

**onSkipButton()** - Method in class com.example.project.controllers.gameScreens.LevelController

**onStartButtonClick()** - Method in class com.example.project.controllers.gameScreens.MainMenuController

**onStartClicked()** - Method in class com.example.project.models.gameScreens.MainMenuModel  
switches to the level screen

**onUpgradeClicked(UpgradeTileController)** - Method in class com.example.project.controllers.gameScreens.ShopController

**onWonLevel()** - Method in class com.example.project.models.gameScreens.LevelModel  
Called when level has been won.

## P

**PageLoader** - Interface in com.example.project.services  
Interface for page loading.

**paperSoundPlayer** - Variable in class com.example.project.models.popups.PopupModel

**PasswordHasher** - Class in com.example.project.services  
Hashes passwords.

**PasswordHasher()** - Constructor for class com.example.project.services.PasswordHasher  
public constructor.

**PasswordHasher(Logger)** - Constructor for class com.example.project.services.PasswordHasher

**play()** - Method in class com.example.project.controllers.gameScreens.animations.AnimationWrapper  
play.

**playersUpgradesProperty()** - Method in class com.example.project.models.gameScreens.ShopModel  
get what upgrades the player currently has.

**playGameMusicLoop()** - Method in class com.example.project.services.sound.GameMusicPlayer  
play game music loop continuously.

**playNextNote()** - Method in class com.example.project.models.gameScreens.ScoreChimePlayer  
plays the note then increments the int index.

**playTiles()** - Method in class com.example.project.models.gameScreens.LevelModel  
clears the word row tiles.

**PopupController<P>** - Class in com.example.project.controllers.popupControllers  
Parent class for PopupControllers

**PopupController()** - Constructor for class com.example.project.controllers.popupControllers.PopupController

**PopupLoader** - Class in com.example.project.services  
class to load the popup and initialize it with a controller

**PopupLoader()** - Constructor for class com.example.project.services.PopupLoader  
Constructor.

**PopupLoader(FXMLLoader)** - Constructor for class com.example.project.services.PopupLoader

**PopupModel** - Class in com.example.project.models.popups

Parent class for popup windows

**PopupModel()** - Constructor for class com.example.project.models.popups.PopupModel

**purchase(UpgradeTileModel)** - Method in class com.example.project.models.gameScreens.ShopModel

This should attempt to purchase an upgrade tile from the shop.

## R

**redrawTiles()** - Method in class com.example.project.models.gameScreens.LevelModel

Redraws tiles into the tile rack and removes from redraw window.

**regenerateShopItems()** - Method in class com.example.project.models.gameScreens.ShopModel

Create new shop items.

**replay()** - Method in class com.example.project.services.sound.GameSoundPlayer

Reset clip and then play.

**reset()** - Method in class com.example.project.models.gameScreens.ScoreChimePlayer

Resets back to 0.

**resetCombo()** - Method in class com.example.project.models.gameScreens.LevelModel

Resets counts for sum and multi in combo

**resetGame()** - Method in class com.example.project.services.Session

Resets the current session when you lose

**resetMoney()** - Method in class com.example.project.services.Session

Resets the player's money to the initial state (e.g.

**resetPlaysRedraws()** - Method in class com.example.project.services.Session

Reset the plays and redraws.

**returnRedrawTilesToTheRack()** - Method in class com.example.project.models.gameScreens.LevelModel

Sends the selected redraw tiles back to the rack

**reversePaperSoundPlayer** - Variable in class com.example.project.models.popups.PopupModel

**rollReward()** - Static method in class com.example.project.models.gameScreens.DailyRewardModel

Randomly selects a daily reward.

**root** - Variable in class com.example.project.controllers.tiles.EmptyTileSlotController

**root** - Variable in class com.example.project.controllers.tiles.UpgradeTileController

**RootLayoutController** - Class in com.example.project.controllers

Root layout that remains the root scene of the application to switch scenes within.

**RootLayoutController()** - Constructor for class com.example.project.controllers.RootLayoutController

## S

**SceneManager** - Class in com.example.project.services

Represents the scene manager class.

**SceneManager(SceneManager)** - Constructor for class com.example.project.services.SceneManager

Constructor to inject an instance.

**SceneManager(SceneManager, RootLayoutController, Map<GameScenes, GameScreenController>, Map<GameScenes, Parent>)** - Constructor for class com.example.project.services.SceneManager

Constructor with injection for tests

**ScoreChimePlayer** - Class in com.example.project.models.gameScreens

Plays a note from the F major scale ascending in pitch each time for a nice score combo sound.

**ScoreChimePlayer()** - Constructor for class com.example.project.models.gameScreens.ScoreChimePlayer

**ScoreTimeline** - Class in com.example.project.controllers.gameScreens.animations

Class for total score counter animation

**ScoreTimeline()** - Constructor for class com.example.project.controllers.gameScreens.animations.ScoreTimeline

**ScrabbleTileData(int, int)** - Constructor for record class com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData

Creates an instance of a ScrabbleTileData record class.

**ScrabbleTileProvider** - Class in com.example.project.models.tiles

Static class to store the scrabble base letter value.

**ScrabbleTileProvider()** - Constructor for class com.example.project.models.tiles.ScrabbleTileProvider

Constructor.

**ScrabbleTileProvider.ScrabbleTileData** - Record Class in com.example.project.models.tiles

**seedRandomNumberGenerator(long)** - Static method in class com.example.project.models.tiles.ScrabbleTileProvider

**sequentialAnimation** - Variable in class com.example.project.controllers.gameScreens.animations.AnimationWrapper

**session** - Variable in class com.example.project.models.gameScreens.GameScreenModel

**Session** - Class in com.example.project.services

Game Session.

**Session()** - Constructor for class com.example.project.services.Session

Default constructor.

**Session(int, int, int, ObservableList<UpgradeTileModel>, User, int, int, int, int, int)** - Constructor for class com.example.project.services.Session

Constructor for injecting values in for unit test.

**setContent(Parent)** - Method in class com.example.project.controllers.RootLayoutController

**setIsDefinitionActive(boolean)** - Method in class com.example.project.models.popups.DefinitionPopup  
toggle isDefinitionActive

**setIsRedrawActive(boolean)** - Method in class com.example.project.models.gameScreens.LevelModel  
set redraw active.

**setLastRewardDate(LocalDate)** - Method in class com.example.project.services.Session  
Sets the date the daily reward was last claimed.

**setLetter(LetterTileController)** - Method in class com.example.project.controllers.tiles.EmptyTileSlotController  
updates the tile slot with a new letter tile controller.

**setLevelModel(LevelModel)** - Method in class com.example.project.services.Session  
sets the session's level model

**setModel(T)** - Method in class com.example.project.controllers.tiles.TileController  
Set the model this controller will observe.

**setOnFinished(EventHandler<ActionEvent>)** - Method in class com.example.project.controllers.gameScreens.animations.AnimationWrapper  
Set on finished.

**setPlayersScore(int)** - Method in class com.example.project.models.gameScreens.LevelModel

**setPopup(String)** - Method in class com.example.project.models.popups.DefinitionPopup

**setPrintToConsole(boolean)** - Method in class com.example.project.services.Logger

**setRoot(StackPane)** - Method in class com.example.project.controllers.tiles.EmptyTileSlotController  
Unit Test for EmptyTileControllerTests

**setSlotForLetterTile(StackPane)** - Method in class com.example.project.controllers.tiles.EmptyTileSlotController

**setSQLiteInstance(Connection)** - Method in class com.example.project.services.sqlite.SQLiteConnection

**setSQLiteInstance(Connection)** - Method in class com.example.project.services.sqlite.SQLiteDictionaryConnection

**setSQLiteInstance(Connection)** - Method in class com.example.project.services.sqlite.SQLiteUsersConnection

**setTile(LetterTileModel)** - Method in class com.example.project.models.tiles.EmptyTileSlotModel  
Set a new tile.

**setTotalScore(int)** - Method in class com.example.project.models.gameScreens.LevelModel

**setupNewLevel()** - Method in class com.example.project.models.gameScreens.LevelModel  
Initialise new level.

**setupProperties(LevelModel)** - Method in class com.example.project.controllers.gameScreens.SidebarController  
Setup sync able properties.

**setUser(User)** - Method in class com.example.project.services.Session  
set new user.

**setWordMulti(int)** - Method in class com.example.project.models.gameScreens.LevelModel  
sets the current word multiplier

**setWordPoints(int)** - Method in class com.example.project.models.gameScreens.LevelModel  
sets the current word points before multipliers

**SHOP** - Enum constant in enum class com.example.project.services.GameScenes  
Shop screen.

**ShopController** - Class in com.example.project.controllers.gameScreens  
Shop view controller.

**ShopController()** - Constructor for class com.example.project.controllers.gameScreens.ShopController  
no arg constructor.

**ShopController(ShopModel, UpgradeTileGroup, UpgradeTileGroup, Logger)** - Constructor for class com.example.project.controllers.gameScreens.ShopController  
protected constructor for unit testing with mock model injection.

**ShopModel** - Class in com.example.project.models.gameScreens  
Shop Model.

**ShopModel(Session)** - Constructor for class com.example.project.models.gameScreens.ShopModel  
Constructor

**ShopModel(Session, Logger)** - Constructor for class com.example.project.models.gameScreens.ShopModel  
Constructor for tests.

**SidebarController** - Class in com.example.project.controllers.gameScreens  
The sidebar on shop and level with the current round info.

**SidebarController()** - Constructor for class com.example.project.controllers.gameScreens.SidebarController

**SidebarLoaded(Parent, SidebarController)** - Constructor for record class com.example.project.controllers.gameScreens.GameScreenController.SideBarLoaded  
Creates an instance of a SidebarLoaded record class.

**slotForLetterTile** - Variable in class com.example.project.controllers.tiles.EmptyTileSlotController

**spendMoney(int)** - Method in class com.example.project.services.Session  
This attempts to spend the specified amount of money from the player's account.

**SQLiteConnection** - Class in com.example.project.services.sqlite  
Abstract SQLiteConnection class.

**SQLiteConnection()** - Constructor for class com.example.project.services.sqlite.SQLiteConnection

**SQLiteDictionaryConnection** - Class in com.example.project.services.sqlite  
Connection to the dictionary.db

**SQLiteDictionaryConnection()** - Constructor for class com.example.project.services.sqlite.SQLiteDictionaryConnection

**SQLiteUsersConnection** - Class in com.example.project.services.sqlite  
Connection to the users.db

**SQLiteUsersConnection()** - Constructor for class com.example.project.services.sqlite.SQLiteUsersConnection

**start(Stage)** - Method in class com.example.project.Application

**SVGIcons** - Class in com.example.project.services  
a static class to hold some icons for the application.

**SVGIcons()** - Constructor for class com.example.project.services.SVGIcons

**switchScene(GameScenes)** - Method in class com.example.project.services.SceneManager  
Switch between scenes specify scene type of type from GameScenes.

**syncTiles()** - Method in class com.example.project.models.tileGroups.TileGroup  
Sync tiles in this tile groups controller.

## T

**TextEmphasisAnimation** - Class in com.example.project.controllers.gameScreens.animations  
An animation to quickly enlarge text and then return it to its original scale.

**TextEmphasisAnimation(Label, Paint, Paint, Duration)** - Constructor for class  
com.example.project.controllers.gameScreens.animations.TextEmphasisAnimation  
create the animations for text growing then shrinking

**TileController<T>** - Class in com.example.project.controllers.tiles  
Parent class of TileControllers.

**TileController()** - Constructor for class com.example.project.controllers.tiles.TileController

**tileControllerFactory** - Variable in class com.example.project.models.tileGroups.TileGroup

**TileControllerFactory** - Class in com.example.project.controllers.tiles  
class to load the tile model controller for that tile model type.

**TileControllerFactory()** - Constructor for class com.example.project.controllers.tiles.TileControllerFactory  
Creates instance of TileControllerFactory.

**TileControllerFactory(FXMLLoaderPageLoader)** - Constructor for class com.example.project.controllers.tiles.TileControllerFactory

**tileControllers** - Variable in class com.example.project.models.tileGroups.TileGroup

**TileGroup<modelType,controllerType>** - Class in com.example.project.models.tileGroups  
Tile Group abstract class holds a row or column of tiles.

**TileModel** - Class in com.example.project.models.tiles  
Parent tile class for objects in the scene that are tiles.

**TileModel()** - Constructor for class com.example.project.models.tiles.TileModel

**TITLE** - Static variable in class com.example.project.Application

**toString()** - Method in record class com.example.project.controllers.gameScreens.GameScreenController.SidebarLoaded  
Returns a string representation of this record class.

**toString()** - Method in record class com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData  
Returns a string representation of this record class.

**tryMoveTile(LetterTileModel)** - Method in class com.example.project.models.gameScreens.LevelModel  
determines where tile should go and moves it

## U

**unMute()** - Method in class com.example.project.services.sound.GameSoundPlayer  
Set clip to its default volume.

**updateLevelInfo()** - Method in class com.example.project.services.Session  
Increments how many points are required to beat the level.

**updateVisuals()** - Method in class com.example.project.models.tileGroups.LetterTileGroup

**updateVisuals()** - Method in class com.example.project.models.tileGroups.TileGroup

**updateVisuals()** - Method in class com.example.project.models.tileGroups.UpgradeTileGroup

**UpgradeBuilder()** - Constructor for class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder

**upgradeEffect(Runnable)** - Method in class com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder  
upgrade effect builder

**UpgradeEffects** - Class in com.example.project.models.tiles  
upgrade effect methods

**UpgradeEffects()** - Constructor for class com.example.project.models.tiles.UpgradeEffects

**UpgradeTileController** - Class in com.example.project.controllers.tiles

Controls the upgrade tile view fxml file.

**UpgradeTileController()** - Constructor for class com.example.project.controllers.tiles.UpgradeTileController

**UpgradeTileGroup** - Class in com.example.project.models.tileGroups

tile group that observes an observable list and updates the ui nodes for some game tiles.

**UpgradeTileGroup(Pane, ReadOnlyListProperty<UpgradeTileModel>)** - Constructor for class com.example.project.models.tileGroups.UpgradeTileGroup

Constructor with no additional sync action parameter.

**UpgradeTileGroup(Pane, ReadOnlyListProperty<UpgradeTileModel>, Consumer<UpgradeTileController>)** - Constructor for class com.example.project.models.tileGroups.UpgradeTileGroup

Constructor.

**UpgradeTileModel** - Class in com.example.project.models.tiles

Represents the upgrade tiles sold at the shop.

**UpgradeTileModel(UpgradeTileModel.UpgradeBuilder)** - Constructor for class

com.example.project.models.tiles.UpgradeTileModel

public upgrade constructor.

**UpgradeTileModel.UpgradeBuilder** - Class in com.example.project.models.tiles

builder class for the upgrades

**UpgradeTiles** - Class in com.example.project.services.shopItems

Static class to store all the possible upgrade tiles available in the game.

**UpgradeTiles()** - Constructor for class com.example.project.services.shopItems.UpgradeTiles

**User** - Class in com.example.project.models

Represents a user in the application that can log in.

**User(String, String, int)** - Constructor for class com.example.project.models.User

Constructor.

**UsersDAO** - Class in com.example.project.services.sqlite.dAOs

SQLite Users database.

**UsersDAO()** - Constructor for class com.example.project.services.sqlite.dAOs.UsersDAO

Constructor for this class SQLiteDictionary.

**UsersDAO(PasswordHasher, Connection, Logger)** - Constructor for class com.example.project.services.sqlite.dAOs.UsersDAO

**UsersDAO(Connection, Logger)** - Constructor for class com.example.project.services.sqlite.dAOs.UsersDAO

Constructor with injection for unit tests.

## V

**value()** - Method in record class com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData

Returns the value of the value record component.

**valueOf(String)** - Static method in enum class com.example.project.models.gameScreens.DailyRewardType

Returns the enum constant of this class with the specified name.

**valueOf(String)** - Static method in enum class com.example.project.services.GameScenes

Returns the enum constant of this class with the specified name.

**values()** - Static method in enum class com.example.project.models.gameScreens.DailyRewardType

Returns an array containing the constants of this enum class, in the order they are declared.

**values()** - Static method in enum class com.example.project.services.GameScenes

Returns an array containing the constants of this enum class, in the order they are declared.

## W

**WIDTH** - Static variable in class com.example.project.Application

**wordMultiProperty()** - Method in class com.example.project.models.gameScreens.LevelModel

word multiplier.

**wordPointsProperty()** - Method in class com.example.project.models.gameScreens.LevelModel

**wordProperty()** - Method in class com.example.project.models.popups.DefinitionPopup

A B C D E F G H I L M N O P R S T U V W

All Classes and Interfaces | All Packages | Constant Field Values

## Module com.example.project

module com.example.project

project dependencies.

### Packages

#### Exports

##### Package

##### Description

[com.example.project](#)

[com.example.project.controllers](#)

[com.example.project.controllers.gameScreens](#)

[com.example.project.controllers.gameScreens.animations](#)

[com.example.project.controllers.popupControllers](#)

[com.example.project.controllers.tiles](#)

[com.example.project.models](#)

[com.example.project.models.gameScreens](#)

[com.example.project.models.popups](#)

[com.example.project.models.tileGroups](#)

[com.example.project.models.tiles](#)

[com.example.project.services](#)

[com.example.project.services.shopItems](#)

[com.example.project.services.sound](#)

[com.example.project.services.sqlite](#)

[com.example.project.services.sqlite.dAOs](#)

## Hierarchy For All Packages

### Package Hierarchies:

com.example.project, com.example.project.controllers, com.example.project.controllers.gameScreens, com.example.project.controllers.gameScreens.animations, com.example.project.controllers.popupControllers, com.example.project.controllers.tiles, com.example.project.models, com.example.project.models.gameScreens, com.example.project.models.popups, com.example.project.models.tileGroups, com.example.project.models.tiles, com.example.project.services, com.example.project.services.shopItems, com.example.project.services.sound, com.example.project.services.sqlite, com.example.project.services.sqlite.dAOs

## Class Hierarchy

- java.lang.Object<sup>2</sup>
  - com.example.project.controllers.gameScreens.animations.AnimationWrapper
    - com.example.project.controllers.gameScreens.animations.LevelScoreSequence
    - com.example.project.controllers.gameScreens.animations.TextEmphasisAnimation
  - javafx.application.Application
    - com.example.project.Application
  - com.example.project.models.gameScreens.DailyRewardModel
  - com.example.project.services.sqlite.dAOs.DictionaryDAO
  - com.example.project.services.FXMLPageLoader (implements com.example.project.services.PageLoader)
  - com.example.project.controllers.gameScreens.GameScreenController
    - com.example.project.controllers.gameScreens.DailyRewardController
    - com.example.project.controllers.gameScreens.LevelController
    - com.example.project.controllers.gameScreens.LoginController
    - com.example.project.controllers.gameScreens.MainMenuController
    - com.example.project.controllers.gameScreens.ShopController
  - com.example.project.models.gameScreens.GameScreenModel
    - com.example.project.models.gameScreens.LevelModel
    - com.example.project.models.gameScreens.LoginModel
    - com.example.project.models.gameScreens.MainMenuModel
    - com.example.project.models.gameScreens.ShopModel
  - com.example.project.services.GameSoundPlayer
    - com.example.project.services.sound.GameMusicPlayer
  - com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation
  - com.example.project.services.Logger
  - com.example.project.services.PasswordHasher
  - com.example.project.controllers.popupControllers.PopupController<P>
    - com.example.project.controllers.popupControllers.DefinitionController
  - com.example.project.services.PopupLoader
  - com.example.project.models.popups.PopupModel
    - com.example.project.models.popups.DefinitionPopup
  - com.example.project.controllers.RootLayoutController
  - com.example.project.services.SceneManager
  - com.example.project.models.gameScreens.ScoreChimePlayer
  - com.example.project.controllers.gameScreens.animations.ScoreTimeline
  - com.example.project.models.tiles.ScrabbleTileProvider
  - com.example.project.services.Session
  - com.example.project.controllers.gameScreens SidebarController
  - com.example.project.services.sqlite.SQLiteConnection
    - com.example.project.services.sqlite.SQLiteDictionaryConnection
    - com.example.project.services.sqlite.SQLiteUsersConnection
  - com.example.project.services.SVGIcons
  - com.example.project.controllers.tiles.TileController<T>
    - com.example.project.controllers.tiles.EmptyTileSlotController
    - com.example.project.controllers.tiles.LetterTileController
    - com.example.project.controllers.tiles.UpgradeTileController
  - com.example.project.controllers.tiles.TileControllerFactory
  - com.example.project.models.tileGroups.TileGroup<modelType, controllerType>
    - com.example.project.models.tileGroups.LetterTileGroup
    - com.example.project.models.tileGroups.UpgradeTileGroup
  - com.example.project.models.tiles.TileModel
    - com.example.project.models.tiles.EmptyTileSlotModel
    - com.example.project.models.tiles.LetterTileModel
    - com.example.project.models.tiles.UpgradeTileModel
  - com.example.project.models.tiles.UpgradeEffects
  - com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder
  - com.example.project.services.shopItems.UpgradeTiles
  - com.example.project.models.User
  - com.example.project.services.sqlite.dAOs.UsersDAO

## Interface Hierarchy

- com.example.project.services.PageLoader

## Enum Class Hierarchy

- java.lang.Object<sup>2</sup>
  - java.lang.Enum<sup>2</sup><E> (implements java.lang.Comparable<sup>2</sup><T>, java.lang.constant.Constable<sup>2</sup>, java.io.Serializable<sup>2</sup>)

- com.example.project.models.gameScreens.**DailyRewardType**
- com.example.project.services.**GameScenes**

## Record Class Hierarchy

- java.lang.**Object**<sup>2</sup>
  - java.lang.**Record**<sup>3</sup>
    - com.example.project.controllers.gameScreens.**GameScreenController.SidebarLoaded**
    - com.example.project.models.tiles.**ScrabbleTileProvider.ScrabbleTileData**

---

Copyright © 2025. All rights reserved.

## Search

Reset

► Additional resources

Loading search index...

---

Copyright © 2025. All rights reserved.

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens.animations

## Class TextEmphasisAnimation

```
java.lang.Object
    com.example.project.controllers.gameScreens.animations.AnimationWrapper
        com.example.project.controllers.gameScreens.animations.TextEmphasisAnimation
```

```
public class TextEmphasisAnimation
extends AnimationWrapper
```

An animation to quickly enlarge text and then return it to its original scale.

### Field Summary

#### Fields inherited from

class com.example.project.controllers.gameScreens.animations.AnimationWrapper

sequentialAnimation

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**TextEmphasisAnimation**(javafx.scene.control.Label textToAnimate, javafx.scene.paint.Paint colorAtStart, javafx.scene.paint.Paint changeColourAtEnd, javafx.util.Duration timePauseAfterAnim)

create the animations for text growing then shrinking

### Method Summary

#### Methods inherited from

class com.example.project.controllers.gameScreens.animations.AnimationWrapper

getChildren, play, setOnFinished

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### TextEmphasisAnimation

```
public TextEmphasisAnimation(javafx.scene.control.Label textToAnimate,
                            javafx.scene.paint.Paint colorAtStart,
                            javafx.scene.paint.Paint changeColourAtEnd,
                            javafx.util.Duration timePauseAfterAnim)
```

create the animations for text growing then shrinking

##### Parameters:

textToAnimate - Label.

colorAtStart - initial colour change.

changeColourAtEnd - colour change after finished.

timePauseAfterAnim - time delay after animation until animation finished.



## Uses of Class

**com.example.project.controllers.gameScreens.animations.AnimationWrapper**

### Packages that use AnimationWrapper

#### Package

#### Description

[com.example.project.controllers.gameScreens.animations](#)

## Uses of AnimationWrapper in com.example.project.controllers.gameScreens.animations

### Subclasses of AnimationWrapper in com.example.project.controllers.gameScreens.animations

#### Modifier and Type

#### Class

#### Description

class

#### LevelScoreSequence

Animates each level tile popping up and the score text increasing for every tile in the word row.

class

#### TextEmphasisAnimation

An animation to quickly enlarge text and then return it to its original scale.

**Module** com.example.project

**Package** com.example.project

## Class Application

```
java.lang.Object
    javafx.application.Application
        com.example.project.Application
```

```
public class Application
extends javafx.application.Application
```

The entry point of the application.

### Nested Class Summary

#### **Nested classes/interfaces inherited from class javafx.application.Application**

```
javafx.application.Application.Parameters
```

### Field Summary

#### Fields

**Modifier and Type**

**Field**

**Description**

```
static final int
```

**HEIGHT**

```
static final String
```

**TITLE**

```
static final int
```

**WIDTH**

#### **Fields inherited from class javafx.application.Application**

```
STYLESSHEET_CASPIAN, STYLESSHEET_MODENA
```

### Constructor Summary

#### Constructors

**Constructor**

**Description**

**Application()**

### Method Summary

#### All Methods

#### Static Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

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

void
start(javafx.stage.Stage stage)
```

#### Methods inherited from class javafx.application.Application

getHostServices, getParameters, getUserAgentStylesheet, init, launch, launch, notifyPreloader, setUserAgentStylesheet, stop

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Details

#### TITLE

public static final String TITLE

**See Also:**

Constant Field Values

#### WIDTH

public static final int WIDTH

**See Also:**

Constant Field Values

#### HEIGHT

public static final int HEIGHT

**See Also:**

Constant Field Values

### Constructor Details

#### Application

public Application()

### Method Details

#### start

public void start(javafx.stage.Stage stage)
throws IOException

**Specified by:**

start in class javafx.application.Application

**Throws:**

IOException

## main

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

### Parameters:

args - the command-line arguments passed to your program when it starts.

## Uses of Class

**com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation**

No usage of com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation

---

Copyright © 2025. All rights reserved.

## Uses of Class

### **com.example.project.controllers.gameScreens.animations.LevelScoreSequence**

No usage of com.example.project.controllers.gameScreens.animations.LevelScoreSequence

---

Copyright © 2025. All rights reserved.

## Uses of Class

### **com.example.project.controllers.gameScreens.animations.ScoreTimeline**

No usage of com.example.project.controllers.gameScreens.animations.ScoreTimeline

---

Copyright © 2025. All rights reserved.

## Uses of Class

**com.example.project.controllers.gameScreens.animations.TextEmphasisAnimation**

No usage of com.example.project.controllers.gameScreens.animations.TextEmphasisAnimation

---

Copyright © 2025. All rights reserved.

Module com.example.project

## Package com.example.project.controllers.gameScreens.animations

package com.example.project.controllers.gameScreens.animations

### Related Packages

#### Package

#### Description

com.example.project.controllers.gameScreens

### Classes

#### Class

#### Description

##### AnimationWrapper

Animation wrapper class for our projects animations that get used.

##### InfiniteFloatingAnimation

Balatro style floating tile animation.

##### LevelScoreSequence

Animates each level tile popping up and the score text increasing for every tile in the word row.

##### ScoreTimeline

Class for total score counter animation

##### TextEmphasisAnimation

An animation to quickly enlarge text and then return it to its original scale.

Copyright © 2025. All rights reserved.

## Hierarchy For Package com.example.project.controllers.gameScreens.animations

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
- [com.example.project.controllers.gameScreens.animations.AnimationWrapper](#)
  - [com.example.project.controllers.gameScreens.animations.LevelScoreSequence](#)
  - [com.example.project.controllers.gameScreens.animations.TextEmphasisAnimation](#)
- [com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation](#)
- [com.example.project.controllers.gameScreens.animations.ScoreTimeline](#)

## Uses of Package `com.example.project.controllers.gameScreens.animations`

### Packages that use `com.example.project.controllers.gameScreens.animations`

#### Package

#### Description

`com.example.project.controllers.gameScreens.animations`

### Classes in `com.example.project.controllers.gameScreens.animations` used by `com.example.project.controllers.gameScreens.animations`

#### Class

#### Description

`AnimationWrapper`

Animation wrapper class for our projects animations that get used.

## Uses of Class

### **com.example.project.controllers.gameScreens.DailyRewardController**

No usage of com.example.project.controllers.gameScreens.DailyRewardController

---

Copyright © 2025. All rights reserved.

## Uses of Record Class

[com.example.project.controllers.gameScreens.GameScreenController.SidebarLoaded](#)

### Packages that use GameScreenController.SidebarLoaded

#### Package

#### Description

[com.example.project.controllers.gameScreens](#)

### Uses of GameScreenController.SidebarLoaded in com.example.project.controllers.gameScreens

#### Methods in com.example.project.controllers.gameScreens that return GameScreenController.SidebarLoaded

##### Modifier and Type

##### Method

##### Description

protected [GameScreenController.SidebarLoaded](#)

[GameScreenController.loadSidebar\(\)](#)

## Uses of Class com.example.project.controllers.gameScreens.GameScreenController

### Packages that use GameScreenController

#### Package

#### Description

com.example.project.controllers.gameScreens

com.example.project.services

### Uses of GameScreenController in com.example.project.controllers.gameScreens

#### Subclasses of GameScreenController in com.example.project.controllers.gameScreens

##### Modifier and Type

##### Class

##### Description

class

##### DailyRewardController

Controller for the daily reward popup screen.

class

##### LevelController

Controller for the level screen.

class

##### LoginController

Controller for the login scene.

class

##### MainMenuController

controller for the main menu.

class

##### ShopController

Shop view controller.

### Uses of GameScreenController in com.example.project.services

#### Constructor parameters in com.example.project.services with type arguments of type GameScreenController

##### Modifier

##### Constructor

##### Description

protected

`SceneManager(SceneManager newInstance, RootLayoutController newRootController, Map<GameScenes, GameScreenController> controllersToUse, Map<GameScenes, javafx.scene.Parent> pagesToUse)`

Constructor with injection for tests

## **Uses of Class com.example.project.Application**

No usage of com.example.project.Application

---

Copyright © 2025. All rights reserved.

## Uses of Class

### **com.example.project.controllers.gameScreens.LevelController**

No usage of com.example.project.controllers.gameScreens.LevelController

---

Copyright © 2025. All rights reserved.

## Uses of Class

### **com.example.project.controllers.gameScreens.LoginController**

No usage of com.example.project.controllers.gameScreens.LoginController

---

Copyright © 2025. All rights reserved.

## Uses of Class

### **com.example.project.controllers.gameScreens.MainMenuController**

No usage of com.example.project.controllers.gameScreens.MainMenuController

---

Copyright © 2025. All rights reserved.

## Uses of Class

### **com.example.project.controllers.gameScreens.ShopController**

No usage of com.example.project.controllers.gameScreens.ShopController

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.controllers.gameScreens.SidebarController

### Packages that use SidebarController

#### Package

#### Description

com.example.project.controllers.gameScreens

### Uses of SidebarController in com.example.project.controllers.gameScreens

#### Methods in com.example.project.controllers.gameScreens that return SidebarController

##### Modifier and Type

##### Method

##### Description

SidebarController

GameScreenController.SidebarLoaded.controller()

Returns the value of the controller record component.

#### Constructors in com.example.project.controllers.gameScreens with parameters of type SidebarController

##### Modifier

##### Constructor

##### Description

protected

SidebarLoaded(javafx.scene.Parent node, SidebarController controller)

Creates an instance of a SidebarLoaded record class.

## Package com.example.project.controllers.gameScreens

package com.example.project.controllers.gameScreens

### Related Packages

#### Package

#### Description

[com.example.project.controllers](#)

[com.example.project.controllers.gameScreens.animations](#)

[com.example.project.controllers.popupControllers](#)

[com.example.project.controllers.tiles](#)

### All Classes and Interfaces

### Classes

### Record Classes

#### Class

#### Description

**DailyRewardController**

Controller for the daily reward popup screen.

**GameScreenController**

Game screen controller that has some startup on screen thing todo.

**GameScreenController.SidebarLoaded**

**LevelController**

Controller for the level screen.

**LoginController**

Controller for the login scene.

**MainMenuController**

controller for the main menu.

**ShopController**

Shop view controller.

**SidebarController**

The sidebar on shop and level with the current round info.

## Hierarchy For Package com.example.project.controllers.gameScreens

### Package Hierarchies:

All Packages

### Class Hierarchy

- [java.lang.Object](#)
  - [com.example.project.controllers.gameScreens.GameScreenController](#)
    - [com.example.project.controllers.gameScreens.DailyRewardController](#)
    - [com.example.project.controllers.gameScreens.LevelController](#)
    - [com.example.project.controllers.gameScreens.LoginController](#)
    - [com.example.project.controllers.gameScreens.MainMenuController](#)
    - [com.example.project.controllers.gameScreens.ShopController](#)
  - [com.example.project.controllers.gameScreens.SidebarController](#)

### Record Class Hierarchy

- [java.lang.Object](#)
  - [java.lang.Record](#)
    - [com.example.project.controllers.gameScreens.GameScreenController.SidebarLoaded](#)

---

Copyright © 2025. All rights reserved.

## Uses of Package com.example.project.controllers.gameScreens

### Packages that use com.example.project.controllers.gameScreens

#### Package

#### Description

com.example.project.controllers.gameScreens

com.example.project.services

### Classes in com.example.project.controllers.gameScreens used by com.example.project.controllers.gameScreens

#### Class

#### Description

**GameScreenController**

Game screen controller that has some startup on screen thing todo.

**GameScreenController.SidebarLoaded**

**SidebarController**

The sidebar on shop and level with the current round info.

### Classes in com.example.project.controllers.gameScreens used by com.example.project.services

#### Class

#### Description

**GameScreenController**

Game screen controller that has some startup on screen thing todo.

Module com.example.project

## Package com.example.project.controllers

package com.example.project.controllers

### Related Packages

#### Package

#### Description

[com.example.project](#)

[com.example.project.controllers.gameScreens](#)

[com.example.project.controllers.popupControllers](#)

[com.example.project.controllers.tiles](#)

[com.example.project.models](#)

[com.example.project.services](#)

### Classes

#### Class

#### Description

[RootLayoutController](#)

Root layout that remains the root scene of the application to switch scenes within.

Copyright © 2025. All rights reserved.

## Hierarchy For Package com.example.project.controllers

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#) ↗
  - [com.example.project.controllers.RootLayoutController](#)

---

Copyright © 2025. All rights reserved.

**Module** com.example.project  
**Package** com.example.project.controllers

## Class RootLayoutController

java.lang.Object  
    com.example.project.controllers.RootLayoutController

```
public class RootLayoutController  
extends Object
```

Root layout that remains the root scene of the application to switch scenes within.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

RootLayoutController()

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

void

setContent(javafx.scene.Parent page)

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### RootLayoutController

```
public RootLayoutController()
```

### Method Details

#### setContent

```
public void setContent(javafx.scene.Parent page)
```

##### Parameters:

page - Set page content to a game scene.

## Uses of Package com.example.project.controllers

### Packages that use com.example.project.controllers

#### Package

#### Description

com.example.project.services

### Classes in com.example.project.controllers used by com.example.project.services

#### Class

#### Description

[RootLayoutController](#)

Root layout that remains the root scene of the application to switch scenes within.

**Module** com.example.project

**Package** com.example.project.controllers.popupControllers

## Class DefinitionController

```
java.lang.Object
  com.example.project.controllers.popupControllers.PopupController<DefinitionPopup>
    com.example.project.controllers.popupControllers.DefinitionController
```

```
public class DefinitionController
extends PopupController<DefinitionPopup>
```

Controller for definition popup

### Field Summary

#### Fields inherited from class com.example.project.controllers.popupControllers.PopupController

model

### Constructor Summary

#### Constructors

##### Constructor

##### Description

DefinitionController()

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

DefinitionPopup

**getModel()**

gets the popup model.

javafx.scene.Node

**getStack()**

root ui node.

void

**initialize(DefinitionPopup model)**

sets elements of the ui

#### Methods inherited from class java.lang.Object

clone<sup>✉</sup>, equals<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, toString<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

### Constructor Details

#### DefinitionController

```
public DefinitionController()
```

## Method Details

### getStack

```
public javafx.scene.Node getStack()
```

**Description copied from class: PopupController**

root ui node.

**Specified by:**

[getStack](#) in class [PopupController<DefinitionPopup>](#)

**Returns:**

returns Node

### getModel

```
public DefinitionPopup getModel()
```

**Description copied from class: PopupController**

gets the popup model.

**Overrides:**

[getModel](#) in class [PopupController<DefinitionPopup>](#)

**Returns:**

model.

### initialize

```
public void initialize(DefinitionPopup model)
```

**Description copied from class: PopupController**

sets elements of the ui

**Specified by:**

[initialize](#) in class [PopupController<DefinitionPopup>](#)

**Parameters:**

model - popup to initialize

**Module** com.example.project

**Package** com.example.project.controllers.popupControllers

## Class PopupController<P extends PopupModel>

java.lang.Object  
com.example.project.controllers.PopupController<P>

### Type Parameters:

P - Popup model type

### Direct Known Subclasses:

DefinitionController

---

```
public abstract class PopupController<P extends PopupModel>
extends Object
```

Parent class for PopupControllers

## Field Summary

### Fields

#### Modifier and Type

#### Field

#### Description

protected P

model

## Constructor Summary

### Constructors

#### Constructor

#### Description

PopupController()

## Method Summary

### All Methods

### Instance Methods

### Abstract Methods

### Concrete Methods

#### Modifier and Type

#### Method

#### Description

P

getModel()

gets the popup model.

abstract javafx.scene.Node

getStack()

root ui node.

abstract void

initialize(P popup)

sets elements of the ui

## Methods inherited from class java.lang.Object

```
clone2, equals2, finalize2, getClass2, hashCode2, notify2, notifyAll2, toString2, wait2, wait2, wait2
```

## Field Details

### model

```
protected P extends PopupModel model
```

## Constructor Details

### PopupController

```
public PopupController()
```

## Method Details

### getStack

```
public abstract javafx.scene.Node getStack()
```

root ui node.

**Returns:**

root ui Node

### initialize

```
public abstract void initialize(P popup)
```

sets elements of the ui

**Parameters:**

popup - popup to initialize

### getModel

```
public P getModel()
```

gets the popup model.

**Returns:**

model.

## Uses of Class

### **com.example.project.controllers.popupControllers.DefinitionController**

No usage of com.example.project.controllers.popupControllers.DefinitionController

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.controllers.popupControllers.PopupController

### Packages that use PopupController

#### Package

#### Description

com.example.project.controllers.popupControllers

com.example.project.services

### Uses of PopupController in com.example.project.controllers.popupControllers

#### Subclasses of PopupController in com.example.project.controllers.popupControllers

##### Modifier and Type

##### Class

##### Description

class

##### DefinitionController

Controller for definition popup

### Uses of PopupController in com.example.project.services

#### Methods in com.example.project.services with type parameters of type PopupController

##### Modifier and Type

##### Method

##### Description

<C extends PopupController<P>, P extends PopupModel>

C

**PopupLoader.createPopupController(P popupModel)**

create popup controller.

Module com.example.project

## Package com.example.project.controllers.popupControllers

package com.example.project.controllers.popupControllers

### Related Packages

#### Package

#### Description

[com.example.project.controllers](#)

[com.example.project.controllers.gameScreens](#)

[com.example.project.controllers.tiles](#)

### Classes

#### Class

#### Description

[DefinitionController](#)

Controller for definition popup

[PopupController<P extends PopupModel>](#)

Parent class for PopupControllers

## Hierarchy For Package com.example.project.controllers.popupControllers

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
- [com.example.project.controllers.popupControllers.PopupController<P>](#)
  - [com.example.project.controllers.DefinitionController](#)

## Uses of Package com.example.project.controllers.popupControllers

### Packages that use com.example.project.controllers.popupControllers

Package	Description
com.example.project.controllers.popupControllers	
com.example.project.services	

### Classes in com.example.project.controllers.popupControllers used by com.example.project.controllers.popupController

Class	Description
PopupController	Parent class for PopupControllers

### Classes in com.example.project.controllers.popupControllers used by com.example.project.services

Class	Description
PopupController	Parent class for PopupControllers

**Module** com.example.project  
**Package** com.example.project.controllers.tiles

## Class EmptyTileSlotController

java.lang.Object  
  com.example.project.controllers.tiles.TileController<EmptyTileSlotModel>  
    com.example.project.controllers.tiles.EmptyTileSlotController

```
public class EmptyTileSlotController  
extends TileController<EmptyTileSlotModel>
```

Empty Tile controller for EmptyTile view.

### Field Summary

#### Fields

##### Modifier and Type

##### Field

##### Description

protected javafx.scene.layout.StackPane  
**root**

protected javafx.scene.layout.StackPane  
**slotForLetterTile**

### Fields inherited from class com.example.project.controllers.tiles.TileController

model

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**EmptyTileSlotController()**

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

void

**bind(EmptyTileSlotModel modelToUse)**

Binds tile to the controller as its model.

javafx.scene.Node

**getRoot()**

root ui node.

void

**setLetter(LetterTileController letterController)**

updates the tile slot with a new letter tile controller.

```
protected void  
setRoot(javafx.scene.layout.StackPane root)
```

Unit Test for EmptyTileControllerTests

```
protected void  
setSlotForLetterTile(javafx.scene.layout.StackPane slot)
```

## Methods inherited from class com.example.project.controllers.tiles.TileController

getModel, setModel

## Methods inherited from class java.lang.Object

clone<sup>✉</sup>, equals<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, toString<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

## Field Details

### root

```
protected javafx.scene.layout.StackPane root
```

### slotForLetterTile

```
protected javafx.scene.layout.StackPane slotForLetterTile
```

## Constructor Details

### EmptyTileSlotController

```
public EmptyTileSlotController()
```

## Method Details

### bind

```
public void bind(EmptyTileSlotModel modelToUse)
```

#### Description copied from class: TileController

Binds tile to the controller as its model.

##### Specified by:

bind in class TileController<EmptyTileSlotModel>

##### Parameters:

modelToUse - tile to bind.

### getRoot

```
public javafx.scene.Node getRoot()
```

#### Description copied from class: TileController

root ui node.

##### Specified by:

getRoot in class TileController<EmptyTileSlotModel>

##### Returns:

### **setRoot**

```
protected void setRoot/javafx.scene.layout.StackPane root)
```

Unit Test for EmptyTileControllerTests

**Parameters:**

root - injecting Stackpane manually

### **setSlotForLetterTile**

```
protected void setSlotForLetterTile/javafx.scene.layout.StackPane slot)
```

### **setLetter**

```
public void setLetter/LetterTileController letterController)
```

updates the tile slot with a new letter tile controller.

**Parameters:**

letterController - the letter tiles controller.

**Module** com.example.project  
**Package** com.example.project.controllers.tiles

## Class LetterTileController

```
java.lang.Object
  com.example.project.controllers.tiles.TileController<LetterTileModel>
    com.example.project.controllers.tiles.LetterTileController
```

```
public class LetterTileController
extends TileController<LetterTileModel>
```

Controls the layout of the letterUIModel. Which is a letter tile.

### Field Summary

#### Fields inherited from class com.example.project.controllers.tiles.TileController

model

### Constructor Summary

#### Constructors

##### Constructor

##### Description

`LetterTileController()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

`void`

`bind(LetterTileModel tile)`

Binds tile to the controller as its model.

`LetterTileModel`

`getModel()`

gets the tile model.

`javafx.scene.Node`

`getRoot()`

root ui node.

#### Methods inherited from class com.example.project.controllers.tiles.TileController

`setModel`

#### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Details

#### LetterTileController

```
public LetterTileController()
```

## Method Details

### getRoot

```
public javafx.scene.Node getRoot()
```

**Description copied from class:** [TileController](#)

root ui node.

**Specified by:**

[getRoot](#) in class [TileController<LetterTileModel>](#)

**Returns:**

root ui Node

### getModel

```
public LetterTileModel getModel()
```

**Description copied from class:** [TileController](#)

gets the tile model.

**Overrides:**

[getModel](#) in class [TileController<LetterTileModel>](#)

**Returns:**

model.

### bind

```
public void bind(LetterTileModel tile)
```

**Description copied from class:** [TileController](#)

Binds tile to the controller as its model.

**Specified by:**

[bind](#) in class [TileController<LetterTileModel>](#)

**Parameters:**

tile - tile to bind.

## Uses of Class com.example.project.controllers.RootLayoutController

### Packages that use RootLayoutController

#### Package

#### Description

com.example.project.services

## Uses of RootLayoutController in com.example.project.services

### Methods in com.example.project.services with parameters of type RootLayoutController

#### Modifier and Type

#### Method

#### Description

void

**SceneManager.initialise(RootLayoutController theRootController, PageLoader loader)**

loads the fxml of all the scenes on launch to switch to via scene manager.

### Constructors in com.example.project.services with parameters of type RootLayoutController

#### Modifier

#### Constructor

#### Description

protected

**SceneManager(SceneManager newInstance, RootLayoutController newRootController, Map<GameScenes, GameScreenController> controllersToUse, Map<GameScenes, javafx.scene.Parent> pagesToUse)**

Constructor with injection for tests

**Module** com.example.project

**Package** com.example.project.controllers.tiles

## Class TileController<T extends TileModel>

java.lang.Object<sup>2</sup>  
com.example.project.controllers.tiles.TileController<T>

### Type Parameters:

T - tile model type.

### Direct Known Subclasses:

EmptyTileSlotController, LetterTileController, UpgradeTileController

---

```
public abstract class TileController<T extends TileModel>
extends Object2
```

Parent class of TileControllers.

### Field Summary

#### Fields

##### Modifier and Type

##### Field

##### Description

protected T

model

### Constructor Summary

#### Constructors

##### Constructor

##### Description

TileController()

### Method Summary

#### All Methods

#### Instance Methods

#### Abstract Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

abstract void

**bind(T tile)**

Binds tile to the controller as its model.

T

**getModel()**

gets the tile model.

abstract javafx.scene.Node

**getRoot()**

root ui node.

void

## **setModel(T tile)**

Set the model this controller will observe.

## **Methods inherited from class java.lang.Object**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## **Field Details**

### **model**

`protected T extends TileModel model`

## **Constructor Details**

### **TileController**

`public TileController()`

## **Method Details**

### **getRoot**

`public abstract javafx.scene.Node getRoot()`

root ui node.

**Returns:**

root ui Node

### **bind**

`public abstract void bind(T tile)`

Binds tile to the controller as its model.

**Parameters:**

tile - tile to bind.

### **setModel**

`public void setModel(T tile)`

Set the model this controller will observe.

**Parameters:**

tile - tile model class.

### **getModel**

`public T getModel()`

gets the tile model.

**Returns:**

model.



**Module** com.example.project  
**Package** com.example.project.controllers.tiles

## Class TileControllerFactory

java.lang.Object  
    com.example.project.controllers.tiles.TileControllerFactory

```
public class TileControllerFactory  
extends Object
```

class to load the tile model controller for that tile model type.

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

`TileControllerFactory()`

Creates instance of `TileControllerFactory`.

protected

`TileControllerFactory(FXMLLoader loader)`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

`LetterTileController`

`createLetterTileController(LetterTileModel lt)`

`<C extends TileController<T>, T extends TileModel>`  
C

`createTileController(T tile, Class<?> controllerType)`

Returns controller of type.

#### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Details

#### TileControllerFactory

`public TileControllerFactory()`

Creates instance of `TileControllerFactory`.

#### TileControllerFactory

`protected TileControllerFactory(FXMLLoader loader)`

## Method Details

### createTileController

```
public <C extends TileController<T>,
      T extends TileModel>
C createTileController(T tile,
                      Class<?> controllerType)
```

Returns controller of type.

**Type Parameters:**

C - controller type.

T - tile type.

**Parameters:**

tile - tile model.

controllerType - controller type.

**Returns:**

return controller.

### createLetterTileController

```
public LetterTileController createLetterTileController(LetterTileModel lt)
```

**Parameters:**

lt - letter tile model.

**Returns:**

returns letter tile controller.

**Module** com.example.project  
**Package** com.example.project.controllers.tiles

## Class UpgradeTileController

```
java.lang.Object
    com.example.project.controllers.tiles.TileController<UpgradeTileModel>
        com.example.project.controllers.tiles.UpgradeTileController
```

```
public class UpgradeTileController
extends TileController<UpgradeTileModel>
```

Controls the upgrade tile view fxml file.

### Field Summary

#### Fields

##### Modifier and Type

##### Field

##### Description

protected javafx.scene.image.ImageView  
**imageView**

protected javafx.scene.layout.StackPane  
**root**

### Fields inherited from class com.example.project.controllers.tiles.TileController

model

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**UpgradeTileController()**

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

void

**bind(UpgradeTileModel tileModel)**

Binds tile to the controller as its model.

javafx.scene.Node

**getRoot()**

root ui node.

### Methods inherited from class com.example.project.controllers.tiles.TileController

`getModel, setModel`

#### Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Field Details

#### `root`

`protected javafx.scene.layout.StackPane root`

#### `imageView`

`protected javafx.scene.image.ImageView imageView`

### Constructor Details

#### `UpgradeTileController`

`public UpgradeTileController()`

### Method Details

#### `getRoot`

`public javafx.scene.Node getRoot()`

##### Description copied from class: `TileController`

root ui node.

##### Specified by:

`getRoot` in class `TileController<UpgradeTileModel>`

##### Returns:

root ui Node

#### `bind`

`public void bind(UpgradeTileModel tileModel)`

##### Description copied from class: `TileController`

Binds tile to the controller as its model.

##### Specified by:

`bind` in class `TileController<UpgradeTileModel>`

##### Parameters:

tileModel - tile to bind.

## Uses of Class

### **com.example.project.controllers.tiles.EmptyTileSlotController**

No usage of com.example.project.controllers.tiles.EmptyTileSlotController

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.controllers.tiles.LetterTileController

### Packages that use LetterTileController

#### Package

#### Description

com.example.project.controllers.gameScreens.animations

com.example.project.controllers.tiles

com.example.project.models.tileGroups

### Uses of LetterTileController in com.example.project.controllers.gameScreens.animations

#### Constructor parameters in com.example.project.controllers.gameScreens.animations with type arguments of type LetterTileController

##### Modifier

##### Constructor

##### Description

`LevelScoreSequence(List<LetterTileController> wordTileControllers, LevelModel levelModel, javafx.scene.control.Label comboCountLabel, javafx.scene.control.Label multiplierLabel)`

Constructor.

### Uses of LetterTileController in com.example.project.controllers.tiles

#### Methods in com.example.project.controllers.tiles that return LetterTileController

##### Modifier and Type

##### Method

##### Description

`LetterTileController`

`TileControllerFactory.createLetterTileController(LetterTileModel lt)`

#### Methods in com.example.project.controllers.tiles with parameters of type LetterTileController

##### Modifier and Type

##### Method

##### Description

`void`

`EmptyTileSlotController.setLetter(LetterTileController letterController)`

updates the tile slot with a new letter tile controller.

### Uses of LetterTileController in com.example.project.models.tileGroups

#### Constructor parameters in com.example.project.models.tileGroups with type arguments of type LetterTileController

##### Modifier

##### Constructor

## Description

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<  
<LetterTileController> onClickAction)
```

Constructor

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<  
<LetterTileController> onClickHandler, List<Runnable> afterSyncActions)
```

Constructor

protected

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<  
<LetterTileController> onClickHandler, List<Runnable> afterSyncActions, TileControllerFactory factory)
```

## Uses of Class com.example.project.controllers.tiles.TileController

### Packages that use TileController

#### Package

#### Description

com.example.project.controllers.tiles

com.example.project.models.tileGroups

## Uses of TileController in com.example.project.controllers.tiles

### Subclasses of TileController in com.example.project.controllers.tiles

#### Modifier and Type

#### Class

#### Description

class

#### EmptyTileSlotController

Empty Tile controller for EmptyTile view.

class

#### LetterTileController

Controls the layout of the letterUIModel.

class

#### UpgradeTileController

Controls the upgrade tile view fxml file.

### Methods in com.example.project.controllers.tiles with type parameters of type TileController

#### Modifier and Type

#### Method

#### Description

<C extends TileController<T>, T extends TileModel>

C

TileControllerFactory.createTileController(T tile, Class<?> controllerType)

Returns controller of type.

## Uses of TileController in com.example.project.models.tileGroups

### Classes in com.example.project.models.tileGroups with type parameters of type TileController

#### Modifier and Type

#### Class

#### Description

class

TileGroup<modelType extends TileModel, controllerType extends TileController<modelType>>

Tile Group abstract class holds a row or column of tiles.



## Uses of Class com.example.project.controllers.tiles.TileControllerFactory

### Packages that use TileControllerFactory

#### Package

#### Description

com.example.project.models.tileGroups

### Uses of TileControllerFactory in com.example.project.models.tileGroups

#### Fields in com.example.project.models.tileGroups declared as TileControllerFactory

##### Modifier and Type

##### Field

##### Description

protected TileControllerFactory

TileGroup.tileControllerFactory

#### Constructors in com.example.project.models.tileGroups with parameters of type TileControllerFactory

##### Modifier

##### Constructor

##### Description

protected

LetterTileGroup(int numberEmptyTileSlots, javafx.scene.layout.Pane container, javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<LetterTileController> onClickHandler, List<Runnable> afterSyncActions, TileControllerFactory factory)

## Uses of Class com.example.project.controllers.tiles.UpgradeTileController

### Packages that use UpgradeTileController

#### Package

#### Description

com.example.project.controllers.gameScreens

com.example.project.models.tileGroups

### Uses of UpgradeTileController in com.example.project.controllers.gameScreens

#### Methods in com.example.project.controllers.gameScreens with parameters of type UpgradeTileController

##### Modifier and Type

##### Method

##### Description

protected void

`ShopController.onUpgradeClicked(UpgradeTileController controller)`

### Uses of UpgradeTileController in com.example.project.models.tileGroups

#### Constructor parameters in com.example.project.models.tileGroups with type arguments of type UpgradeTileController

##### Modifier

##### Constructor

##### Description

`UpgradeTileGroup(javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> observedList, Consumer<  
<UpgradeTileController> onClickAction)`

Constructor.

Module com.example.project

## Package com.example.project.controllers.tiles

package com.example.project.controllers.tiles

### Related Packages

#### Package

#### Description

[com.example.project.controllers](#)

[com.example.project.controllers.gameScreens](#)

[com.example.project.controllers.popupControllers](#)

### Classes

#### Class

#### Description

[EmptyTileSlotController](#)

Empty Tile controller for EmptyTile view.

[LetterTileController](#)

Controls the layout of the letterUIModel.

[TileController<T extends TileModel>](#)

Parent class of TileControllers.

[TileControllerFactory](#)

class to load the tile model controller for that tile model type.

[UpgradeTileController](#)

Controls the upgrade tile view fxml file.

Copyright © 2025. All rights reserved.

## Hierarchy For Package com.example.project.controllers.tiles

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
- [com.example.project.controllers.tiles.TileController<T>](#)
  - [com.example.project.controllers.tiles.EmptyTileSlotController](#)
  - [com.example.project.controllers.tiles.LetterTileController](#)
  - [com.example.project.controllers.tiles.UpgradeTileController](#)
- [com.example.project.controllers.tiles.TileControllerFactory](#)

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Class DailyRewardController

java.lang.Object ↗  
    com.example.project.controllers.gameScreens.GameScreenController  
        com.example.project.controllers.gameScreens.DailyRewardController

```
public class DailyRewardController  
extends GameScreenController
```

Controller for the daily reward popup screen.

### Nested Class Summary

**Nested classes/interfaces inherited from class com.example.project.controllers.gameScreens.GameScreenController**

GameScreenController.SidebarLoaded

### Field Summary

**Fields inherited from class com.example.project.controllers.gameScreens.GameScreenController**

logger

### Constructor Summary

#### Constructors

Constructor

Description

DailyRewardController()

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

void

**initialize()**

Initialises the daily reward view by resetting the reward label.

void

**onSceneChangedToThis()**

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

**Methods inherited from class com.example.project.controllers.gameScreens.GameScreenController**

loadSidebar

**Methods inherited from class java.lang.Object ↗**

clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()

## **Constructor Details**

### **DailyRewardController**

```
public DailyRewardController()
```

## **Method Details**

### **initialize**

```
public void initialize()
```

Initialises the daily reward view by resetting the reward label.

### **onSceneChangedToThis**

```
public void onSceneChangedToThis()
```

**Description copied from class:** [GameScreenController](#)

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

**Specified by:**

onSceneChangedToThis in class GameScreenController

## Uses of Package com.example.project.controllers.tiles

### Packages that use com.example.project.controllers.tiles

Package	Description
com.example.project.controllers.gameScreens	

com.example.project.controllers.gameScreens.animations
--

com.example.project.controllers.tiles
---------------------------------------

com.example.project.models.tileGroups
---------------------------------------

### Classes in com.example.project.controllers.tiles used by com.example.project.controllers.gameScreens

Class	Description
<a href="#">UpgradeTileController</a>	
Controls the upgrade tile view fxml file.	

### Classes in com.example.project.controllers.tiles used by com.example.project.controllers.gameScreens.animations

Class	Description
<a href="#">LetterTileController</a>	
Controls the layout of the letterUIModel.	

### Classes in com.example.project.controllers.tiles used by com.example.project.controllers.tiles

Class	Description
<a href="#">LetterTileController</a>	
Controls the layout of the letterUIModel.	

### Classes in com.example.project.controllers.tiles used by com.example.project.models.tileGroups

Class	Description
<a href="#">LetterTileController</a>	
Controls the layout of the letterUIModel.	
<a href="#">TileController</a>	
Parent class of TileControllers.	
<a href="#">TileControllerFactory</a>	
class to load the tile model controller for that tile model type.	
<a href="#">UpgradeTileController</a>	
Controls the upgrade tile view fxml file.	



**Module** com.example.project  
**Package** com.example.project.models

## Class User

java.lang.Object<sup>✉</sup>  
    com.example.project.models.User

```
public class User  
extends Object✉
```

Represents a user in the application that can log in.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

User(String<sup>✉</sup> username, String<sup>✉</sup> password, int highscore)

Constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

int

getHighscore()

get highscore.

String<sup>✉</sup>

getPassword()

get password.

String<sup>✉</sup>

getUsername()

get username.

#### Methods inherited from class java.lang.Object<sup>✉</sup>

clone<sup>✉</sup>, equals<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, toString<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

### Constructor Details

#### User

```
public User(String✉ username,  
          String✉ password,  
          int highscore)
```

Constructor.

##### Parameters:

username - username.

password - password.

highscore - highscore.

## **Method Details**

### **getUsername**

```
public String¶ getUsername()
```

get username.

**Returns:**

username.

### **getPassword**

```
public String¶ getPassword()
```

get password.

**Returns:**

password.

### **getHighscore**

```
public int getHighscore()
```

get highscore.

**Returns:**

returns highscore.

## Uses of Class com.example.project.models.User

### Packages that use User

#### Package

#### Description

com.example.project.services

com.example.project.services.sqlite.dAOs

## Uses of User in com.example.project.services

### Methods in com.example.project.services with parameters of type User

#### Modifier and Type

#### Method

#### Description

void

**Session.setUser(User newUser)**

set new user.

### Constructors in com.example.project.services with parameters of type User

#### Modifier

#### Constructor

#### Description

protected

**Session(int newHandSize, int newWordViewSize, int newRedrawWindowSize,  
javafx.collections.ObservableList<UpgradeTileModel> newUpgrades, User newUser, int newMoney,  
int newLevelsBeaten, int currentLevelRequirement, int newFirstLevelsRequirement, int newInitialMoney)**

Constructor for injecting values in for unit test.

## Uses of User in com.example.project.services.sqlite.dAOs

### Methods in com.example.project.services.sqlite.dAOs that return User

#### Modifier and Type

#### Method

#### Description

User

**UsersDAO.getUser(String<sup>2</sup> username)**

### Methods in com.example.project.services.sqlite.dAOs with parameters of type User

#### Modifier and Type

#### Method

#### Description

void

**UsersDAO.addUser(User user)**

Adds user to the user.db.



**Module** com.example.project  
**Package** com.example.project.models.gameScreens

## Class DailyRewardModel

java.lang.Object  
    com.example.project.models.gameScreens.DailyRewardModel

```
public class DailyRewardModel  
extends Object
```

Model for handling daily reward logic and state.

### Constructor Summary

#### Constructors

##### Constructor

##### Description

DailyRewardModel()

### Method Summary

#### All Methods

#### Static Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

static void

applyReward(DailyRewardType reward)

Applies the effect of the given reward to the session and marks it as claimed.

static DailyRewardType

rollReward()

Randomly selects a daily reward.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### DailyRewardModel

```
public DailyRewardModel()
```

### Method Details

#### rollReward

```
public static DailyRewardType rollReward()
```

Randomly selects a daily reward.

##### Returns:

a randomly selected reward

## applyReward

```
public static void applyReward(DailyRewardType reward)
```

Applies the effect of the given reward to the session and marks it as claimed.

**Parameters:**

reward - the reward to apply

**Module** com.example.project  
**Package** com.example.project.models.gameScreens

## Enum Class DailyRewardType

java.lang.Object  
  java.lang.Enum<DailyRewardType>  
    com.example.project.models.gameScreens.DailyRewardType

### All Implemented Interfaces:

Serializable, Comparable<DailyRewardType>, Constable

```
public enum DailyRewardType  
extends Enum<DailyRewardType>
```

Enum representing the possible daily reward outcomes.

### Nested Class Summary

#### **Nested classes/interfaces inherited from class java.lang.Enum**

Enum.EnumDesc<E> extends Enum<E>>

### Enum Constant Summary

#### Enum Constants

##### Enum Constant

##### Description

**Daily\_Reward\_Won\_1Dollar**

Player wins \$1

**Daily\_Reward\_Won\_5Dollars**

Player wins \$5

**Daily\_Reward\_Won\_NOTHING**

Player wins nothing

### Method Summary

#### All Methods

#### Static Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

static DailyRewardType

**valueOf(String name)**

Returns the enum constant of this class with the specified name.

static DailyRewardType[]

**values()**

Returns an array containing the constants of this enum class, in the order they are declared.

#### Methods inherited from class java.lang.Enum

clone, compareTo, describeConstable, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

## Enum Constant Details

### Daily\_Reward\_Won\_1Dollar

```
public static final DailyRewardType Daily_Reward_Won_1Dollar
```

Player wins \$1

### Daily\_Reward\_Won\_5Dollars

```
public static final DailyRewardType Daily_Reward_Won_5Dollars
```

Player wins \$5

### Daily\_Reward\_Won\_NOTHING

```
public static final DailyRewardType Daily_Reward_Won_NOTHING
```

Player wins nothing

## Method Details

### values

```
public static DailyRewardType[] values()
```

Returns an array containing the constants of this enum class, in the order they are declared.

**Returns:**

an array containing the constants of this enum class, in the order they are declared

### valueOf

```
public static DailyRewardType valueOf(String¶ name)
```

Returns the enum constant of this class with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this class. (Extraneous whitespace characters are not permitted.)

**Parameters:**

name - the name of the enum constant to be returned.

**Returns:**

the enum constant with the specified name

**Throws:**

IllegalArgumentException<sup>¶</sup> - if this enum class has no constant with the specified name

NullPointerException<sup>¶</sup> - if the argument is null

**Module** com.example.project  
**Package** com.example.project.models.gameScreens

## Class GameScreenModel

java.lang.Object  
com.example.project.models.gameScreens.GameScreenModel

### Direct Known Subclasses:

LevelModel, LoginModel, MainMenuModel, ShopModel

---

```
public abstract class GameScreenModel
extends Object
```

Game Screen Model. extended by loginModel, shopModel, levelModel.

## Field Summary

### Fields

#### Modifier and Type

#### Field

#### Description

protected Logger

logger

protected Session

session

## Constructor Summary

### Constructors

#### Modifier

#### Constructor

#### Description

protected

GameScreenModel(Session session)

## Method Summary

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Details

### logger

protected Logger logger

### session

protected Session session

## **Constructor Details**

### **GameScreenModel**

```
protected GameScreenModel(Session session)
```

**Module** com.example.project

**Package** com.example.project.models.gameScreens

## Class LevelModel

```
java.lang.Object
  com.example.project.models.gameScreens.GameScreenModel
    com.example.project.models.gameScreens.LevelModel
```

```
public class LevelModel
extends GameScreenModel
```

Represents the level model.

### Property Summary

#### Properties

Type

Property

Description

```
javafx.beans.property.ReadOnlyIntegerProperty
```

**wordMulti**

word multiplier.

```
javafx.beans.property.ReadOnlyIntegerProperty
```

**wordPoints**

### Field Summary

#### Fields inherited from class com.example.project.models.gameScreens.GameScreenModel

```
logger, session
```

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

```
LevelModel(Session session)
```

Default constructor.

protected

```
LevelModel(Session session, Logger logger)
```

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

```
protected void
addTileToRack(LetterTileModel tile)

protected void
addTileToRedrawWindow(LetterTileModel tile)

protected void
addTileToWordWindow(LetterTileModel tile)

void
addToCombo(LetterTileModel tile)
add combo sum and multiCombo

int
calcTotalWordScore()

String
getCurrentWord()
Gets the current word formed by tiles in the word area

javafx.beans.property.ReadOnlyBooleanProperty
getIsRedrawActive()
gets the redraw property.

javafx.beans.property.ReadOnlyIntegerProperty
getPlayersCurrentPoints()

javafx.beans.property.ReadOnlyIntegerProperty
getPlayersTotalPoints()

javafx.beans.property.ReadOnlyListProperty<LetterTileModel>
getRedrawWindowTilesProperty()

javafx.beans.property.ReadOnlyListProperty<LetterTileModel>
getTileRackTilesProperty()

ScoreChimePlayer
getTileScoreSoundPlayer()
Gets the tile score sound effect player.

javafx.beans.property.ReadOnlyListProperty<LetterTileModel>
getWordWindowTilesProperty()

boolean
hasLost()
true if player has lost

boolean
hasWon()
returns true if player has won.

boolean
```

**isCurrentWordValid()**

returns true if word is in dictionary.

void

**onLostLevel()**

Called when the level has been lost.

void

**onWonLevel()**

Called when level has been won.

void

**playTiles()**

clears the word row tiles.

void

**redrawTiles()**

redraws tiles into the tile rack and removes from redraw window.

void

**resetCombo()**

resets counts for sum and multi in combo

void

**returnRedrawTilesToTheRack()**

sends the selected redraw tiles back to the rack

void

**setIsRedrawActive(boolean newValue)**

set redraw active.

protected void

**setPlayersScore(int newValue)**

void

**setTotalScore(int totalScore)**

void

**setupNewLevel()**

Initialise new level.

void

**setWordMulti(int newMulti)**

sets the current word multiplier

void

**setWordPoints(int newWordPoints)**

sets the current word points before multipliers

boolean

**tryMoveTile(LetterTileModel tile)**

determines where tile should go and moves it

javafx.beans.property.ReadOnlyIntegerProperty

**wordMultiProperty()**

word multiplier.

javafx.beans.property.ReadOnlyIntegerProperty

```
wordPointsProperty()
```

#### Methods inherited from class java.lang.Object

```
clone2, equals2, finalize2, getClass2, hashCode2, notify2, notifyAll2, toString2, wait2, wait2, wait2
```

### Property Details

#### wordPoints

```
public javafx.beans.property.ReadOnlyIntegerProperty wordPointsProperty
```

##### See Also:

```
setWordPoints(int), wordPointsProperty()
```

#### wordMulti

```
public javafx.beans.property.ReadOnlyIntegerProperty wordMultiProperty
```

word multiplier.

##### See Also:

```
setWordMulti(int), wordMultiProperty()
```

### Constructor Details

#### LevelModel

```
protected LevelModel(Session session,  
                    Logger logger)
```

#### LevelModel

```
public LevelModel(Session session)
```

Default constructor.

##### Parameters:

session - session

### Method Details

#### getTileScoreSoundPlayer

```
public ScoreChimePlayer getTileScoreSoundPlayer()
```

Gets the tile score sound effect player.

##### Returns:

LevelTileScoreSoundPlayer.

#### getWordWindowTilesProperty

```
public javafx.beans.property.ReadOnlyListProperty<LetterTileModel> getWordWindowTilesProperty()
```

##### Returns:

Read-only list of tiles currently in the word area

## getTileRackTilesProperty

```
public javafx.beans.property.ReadOnlyListProperty<LetterTileModel> getTileRackTilesProperty()
```

**Returns:**

Read-only list of tiles currently in the rack

## getRedrawWindowTilesProperty

```
public javafx.beans.property.ReadOnlyListProperty<LetterTileModel> getRedrawWindowTilesProperty()
```

**Returns:**

Read-only list of tiles currently in the redraw window.

## getPlayersTotalPoints

```
public javafx.beans.property.ReadOnlyIntegerProperty getPlayersTotalPoints()
```

**Returns:**

the total points property to observe.

## wordPointsProperty

```
public javafx.beans.property.ReadOnlyIntegerProperty wordPointsProperty()
```

**Returns:**

the sum combo points property to observe.

**See Also:**

[setWordPoints\(int\)](#)

## getPlayersCurrentPoints

```
public javafx.beans.property.ReadOnlyIntegerProperty getPlayersCurrentPoints()
```

**Returns:**

the players current level points property to observe.

## wordMultiProperty

```
public javafx.beans.property.ReadOnlyIntegerProperty wordMultiProperty()
```

word multiplier.

**Returns:**

multiplier.

**See Also:**

[setWordMulti\(int\)](#)

## getIsRedrawActive

```
public javafx.beans.property.ReadOnlyBooleanProperty getIsRedrawActive()
```

gets the redraw property.

**Returns:**

returns indication if redraw active.

## setIsRedrawActive

```
public void setIsRedrawActive(boolean newValue)
```

set redraw active.

**Parameters:**

`newValue` - get if redraw window is on screen.

## **setPlayersScore**

```
protected void setPlayersScore(int newValue)
```

## **addTileToWordWindow**

```
protected void addTileToWordWindow(LetterTileModel tile)
```

## **addTileToRack**

```
protected void addTileToRack(LetterTileModel tile)
```

## **addTileToRedrawWindow**

```
protected void addTileToRedrawWindow(LetterTileModel tile)
```

## **onLostLevel**

```
public void onLostLevel()
```

Called when the level has been lost. resets the players session info and logs back out to the login screen.

## **onWonLevel**

```
public void onWonLevel()
```

Called when level has been won. reset the per level info: redraws plays. Goes to shop window.

## **hasWon**

```
public boolean hasWon()
```

returns true if player has won.

**Returns:**

value indicating if player has won.

## **hasLost**

```
public boolean hasLost()
```

true if player has lost

**Returns:**

value indicating if player has lost.

## **tryMoveTile**

```
public boolean tryMoveTile(LetterTileModel tile)
```

determines where tile should go and moves it

**Parameters:**

`tile` - The tile to move

**Returns:**

true if move was successful, false otherwise

## **getCurrentWord**

```
public String2 getCurrentWord()
```

Gets the current word formed by tiles in the word area

**Returns:**

returns current word string.

### isCurrentWordValid

```
public boolean isCurrentWordValid()
```

returns true if word is in dictionary.

**Returns:**

value indicating if word is valid.

### redrawTiles

```
public void redrawTiles()
```

redraws tiles into the tile rack and removes from redraw window.

### addToCombo

```
public void addToCombo(LetterTileModel tile)
```

add combo sum and multiCombo

**Parameters:**

tile - tile.

### calcTotalWordScore

```
public int calcTotalWordScore()
```

**Returns:**

total score int

### setWordPoints

```
public void setWordPoints(int newWordPoints)
```

sets the current word points before multipliers

**Parameters:**

newWordPoints - the new word points value

### setWordMulti

```
public void setWordMulti(int newMulti)
```

sets the current word multiplier

**Parameters:**

newMulti - the new multiplier value

### setTotalScore

```
public void setTotalScore(int totalScore)
```

**Parameters:**

totalScore - from calcTotalScore sets Total Score

### playTiles

```
public void playTiles()
```

clears the word row tiles. and refills the tile rack. and decreases the plays left.

### **returnRedrawTilesToTheRack**

`public void returnRedrawTilesToTheRack()`

sends the selected redraw tiles back to the rack

### **setupNewLevel**

`public void setupNewLevel()`

Initialise new level. Clears word row, redraw rack. draws new tiles for the player's tile rack.

### **resetCombo**

`public void resetCombo()`

resets counts for sum and multi in combo

**Module** com.example.project

**Package** com.example.project.models.gameScreens

## Class LoginModel

java.lang.Object<sup>↳</sup>  
  com.example.project.models.gameScreens.GameScreenModel  
    com.example.project.models.gameScreens.LoginModel

```
public class LoginModel  
extends GameScreenModel
```

Login model class.

### Field Summary

#### Fields inherited from class com.example.project.models.gameScreens.GameScreenModel

logger, session

### Constructor Summary

#### Constructors

##### Constructor

##### Description

`LoginModel(Session session, UsersDAO dao, PasswordHasher passwordHasher)`

constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

`javafx.beans.property.ReadOnlyStringProperty`

`getWelcomeTextProperty()`

Gets the info text property.

`void`

`onLoginClicked(String✉ username, String✉ password)`

`void`

`onSignUpClicked(String✉ username, String✉ password)`

Adds user to database.

#### Methods inherited from class java.lang.Object<sup>↳</sup>

`clone✉, equals✉, finalize✉, getClass✉, hashCode✉, notify✉, notifyAll✉, toString✉, wait✉, wait✉, wait✉`

### Constructor Details

#### LoginModel

```
public LoginModel(Session session,  
                  UsersDAO dao,  
                  PasswordHasher passwordHasher)
```

constructor.

**Parameters:**

session - session to use for the game.  
dao - users database implementation.  
passwordHasher - password hasher for this login

## Method Details

### getWelcomeTextProperty

```
public javafx.beans.property.ReadOnlyStringProperty getWelcomeTextProperty()
```

Gets the info text property.

**Returns:**

ReadonlyStringProperty.

### onLoginClicked

```
public void onLoginClicked(String username,  
                           String password)
```

**Parameters:**

username - username.

password - password.

### onSignUpClicked

```
public void onSignUpClicked(String username,  
                           String password)
```

Adds user to database.

**Parameters:**

username - username

password - password

**Module** com.example.project  
**Package** com.example.project.models.gameScreens

## MainMenuModel

java.lang.Object  
  com.example.project.models.gameScreens.GameScreenModel  
    com.example.project.models.gameScreens.MainMenuModel

```
public class MainMenuModel  
extends GameScreenModel
```

main menu model class

### Field Summary

#### Fields inherited from class com.example.project.models.gameScreens.GameScreenModel

logger, session

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**MainMenuModel(Session session)**

constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

void

**onLogoutClicked()**

switches to the login screen

void

**onStartClicked()**

switches to the level screen

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### MainMenuModel

```
public MainMenuModel(Session session)
```

constructor.

##### Parameters:

session - session to use for the game.

## **Method Details**

### **onStartClicked**

```
public void onStartClicked()
```

switches to the level screen

### **onLogoutClicked**

```
public void onLogoutClicked()
```

switches to the login screen

**Module** com.example.project  
**Package** com.example.project.models.gameScreens

## Class ScoreChimePlayer

java.lang.Object  
    com.example.project.models.gameScreens.ScoreChimePlayer

```
public class ScoreChimePlayer  
extends Object
```

plays a note from the F major scale ascending in pitch each time for a nice score combo sound.

### Constructor Summary

#### Constructors

**Constructor**

**Description**

`ScoreChimePlayer()`

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

`void`

`playNextNote()`

plays the note then increments the int index.

`void`

`reset()`

resets back to 0.

#### Methods inherited from class java.lang.Object

`clone()`, `equals()`, `finalize()`, `getClass()`, `hashCode()`, `notify()`, `notifyAll()`, `toString()`, `wait()`, `wait()`, `wait()`

### Constructor Details

#### ScoreChimePlayer

```
public ScoreChimePlayer()
```

### Method Details

#### reset

```
public void reset()
```

resets back to 0.

#### playNextNote

```
public void playNextNote()
```

```
plays the note then increments the int index.
```

---

Copyright © 2025. All rights reserved.

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Record Class GameScreenController.SidebarLoaded

java.lang.Object<sup>✉</sup>  
  java.lang.Record<sup>✉</sup>  
    com.example.project.controllers.gameScreens.GameScreenController.SidebarLoaded

**Enclosing class:**

GameScreenController

---

```
protected static record GameScreenController.SidebarLoaded(javafx.scene.Parent node,  
SidebarController controller)  
extends Record✉
```

### Constructor Summary

#### Constructors

**Modifier**

**Constructor**

**Description**

protected

**SidebarLoaded**(javafx.scene.Parent node, SidebarController controller)

Creates an instance of a SidebarLoaded record class.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

**Modifier and Type**

**Method**

**Description**

SidebarController

**controller()**

Returns the value of the controller record component.

final boolean

**equals(Object<sup>✉</sup> o)**

Indicates whether some other object is "equal to" this one.

final int

**hashCode()**

Returns a hash code value for this object.

javafx.scene.Parent

**node()**

Returns the value of the node record component.

final String<sup>✉</sup>

**toString()**

Returns a string representation of this record class.

#### Methods inherited from class java.lang.Object<sup>✉</sup>

clone<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

## Constructor Details

### SidebarLoaded

```
protected SidebarLoaded(javafx.scene.Parent node,  
                      SidebarController controller)
```

Creates an instance of a SidebarLoaded record class.

**Parameters:**

node - the value for the node record component

controller - the value for the controller record component

## Method Details

### toString

```
public final String toString()
```

Returns a string representation of this record class. The representation contains the name of the class, followed by the name and value of each of the record components.

**Specified by:**

[toString](#) in class [Record](#)

**Returns:**

a string representation of this object

### hashCode

```
public final int hashCode()
```

Returns a hash code value for this object. The value is derived from the hash code of each of the record components.

**Specified by:**

[hashCode](#) in class [Record](#)

**Returns:**

a hash code value for this object

### equals

```
public final boolean equals(Object o)
```

Indicates whether some other object is "equal to" this one. The objects are equal if the other object is of the same class and if all the record components are equal. All components in this record class are compared with [Objects::equals\(Object, Object\)](#).

**Specified by:**

[equals](#) in class [Record](#)

**Parameters:**

o - the object with which to compare

**Returns:**

true if this object is the same as the o argument; false otherwise.

### node

```
public javafx.scene.Parent node()
```

Returns the value of the node record component.

**Returns:**

the value of the node record component

## controller

```
public SidebarController controller()
```

Returns the value of the controller record component.

**Returns:**

the value of the controller record component

**Module** com.example.project

**Package** com.example.project.models.gameScreens

## Class ShopModel

java.lang.Object  
  com.example.project.models.gameScreens.GameScreenModel  
    com.example.project.models.gameScreens.ShopModel

public class **ShopModel**  
extends [GameScreenModel](#)

Shop Model.

### Property Summary

#### Properties

Type

Property

Description

`javafx.beans.property.ListProperty<UpgradeTileModel>`

**currentShopItems**

Current items in the shop row that you can buy.

`javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel>`

**playersUpgrades**

get what upgrades the player currently has.

### Field Summary

#### Fields inherited from class com.example.project.models.gameScreens.GameScreenModel

`logger, session`

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

**ShopModel(Session session)**

Constructor

protected

**ShopModel(Session session, Logger logger)**

Constructor for tests.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

```
boolean  
canPurchase(UpgradeTileModel tile)
```

Determines whether the player can afford to purchase the specified upgrade tile.

```
javafx.beans.property.ListProperty<UpgradeTileModel>  
currentShopItemsProperty()
```

Current items in the shop row that you can buy.

```
void  
onNextLevelPressed()
```

exists shop and increments level requirement for the next level.

```
javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel>  
playersUpgradesProperty()
```

get what upgrades the player currently has.

```
void  
purchase(UpgradeTileModel tileClickedOn)
```

This should attempt to purchase an upgrade tile from the shop.

```
void  
regenerateShopItems()  
create new shop items.
```

#### Methods inherited from class java.lang.Object

```
clone(), equals(), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()
```

## Property Details

### currentShopItems

```
public javafx.beans.property.ListProperty<UpgradeTileModel> currentShopItemsProperty
```

Current items in the shop row that you can buy.

#### See Also:

[currentShopItemsProperty\(\)](#)

### playersUpgrades

```
public javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> playersUpgradesProperty
```

get what upgrades the player currently has.

#### See Also:

[playersUpgradesProperty\(\)](#)

## Constructor Details

### ShopModel

```
protected ShopModel(Session session,  
                   Logger logger)
```

Constructor for tests.

#### Parameters:

session - session.

logger - logger to use.

## ShopModel

```
public ShopModel(Session session)
```

Constructor

**Parameters:**

session - game session.

## Method Details

### currentShopItemsProperty

```
public javafx.beans.property.ListProperty<UpgradeTileModel> currentShopItemsProperty()
```

Current items in the shop row that you can buy.

**Returns:**

shop items.

### regenerateShopItems

```
public void regenerateShopItems()
```

create new shop items.

### playersUpgradesProperty

```
public javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> playersUpgradesProperty()
```

get what upgrades the player currently has.

**Returns:**

returns the list of upgrades.

### purchase

```
public void purchase(UpgradeTileModel tileClickedOn)
```

This should attempt to purchase an upgrade tile from the shop. Verifies the player has sufficient funds, deducts the cost, removes the item from the shop, and logs the transaction.

**Parameters:**

tileClickedOn - the upgrade tile the player is trying to purchase

**Throws:**

`IllegalArgumentException` - if tileClickedOn is empty (null)

**See Also:**

`to check affordability before calling this method,  
for the payment mechanism,  
for adding to player's collection`

### canPurchase

```
public boolean canPurchase(UpgradeTileModel tile)
```

Determines whether the player can afford to purchase the specified upgrade tile. Compares the player's current money against the tile's cost.

**Parameters:**

tile - the upgrade tile the player wants to buy thus check affordability for

**Returns:**

true if the player has enough money to purchase the tile, false if the player is broke and can't

**Throws:**

`IllegalArgumentException` - if tile is empty

**See Also:**

[for tile pricing](#)

## **onNextLevelPressed**

`public void onNextLevelPressed()`

exists shop and increments level requirement for the next level.

## Uses of Class

### **com.example.project.models.gameScreens.DailyRewardModel**

No usage of com.example.project.models.gameScreens.DailyRewardModel

---

Copyright © 2025. All rights reserved.

## Uses of Enum Class

### com.example.project.models.gameScreens.DailyRewardType

#### Packages that use DailyRewardType

##### Package

##### Description

com.example.project.models.gameScreens

#### Uses of DailyRewardType in com.example.project.models.gameScreens

#### Methods in com.example.project.models.gameScreens that return DailyRewardType

##### Modifier and Type

##### Method

##### Description

static DailyRewardType

DailyRewardModel.rollReward()

Randomly selects a daily reward.

static DailyRewardType

DailyRewardType.valueOf(String<sup>?</sup> name)

Returns the enum constant of this class with the specified name.

static DailyRewardType[]

DailyRewardType.values()

Returns an array containing the constants of this enum class, in the order they are declared.

#### Methods in com.example.project.models.gameScreens with parameters of type DailyRewardType

##### Modifier and Type

##### Method

##### Description

static void

DailyRewardModel.applyReward(DailyRewardType reward)

Applies the effect of the given reward to the session and marks it as claimed.

## Uses of Class com.example.project.models.gameScreens.GameScreenModel

### Packages that use GameScreenModel

#### Package

#### Description

com.example.project.models.gameScreens

### Uses of GameScreenModel in com.example.project.models.gameScreens

#### Subclasses of GameScreenModel in com.example.project.models.gameScreens

##### Modifier and Type

##### Class

##### Description

class

##### LevelModel

Represents the level model.

class

##### LoginModel

Login model class.

class

##### MainMenuModel

main menu model class

class

##### ShopModel

Shop Model.

## Uses of Class com.example.project.models.gameScreens.LevelModel

### Packages that use LevelModel

#### Package

#### Description

com.example.project.controllers.gameScreens

com.example.project.controllers.gameScreens.animations

com.example.project.services

### Uses of LevelModel in com.example.project.controllers.gameScreens

#### Methods in com.example.project.controllers.gameScreens with parameters of type LevelModel

##### Modifier and Type

##### Method

##### Description

void

`SidebarController.setupProperties(LevelModel levelModel)`

Setup sync able properties.

#### Constructors in com.example.project.controllers.gameScreens with parameters of type LevelModel

##### Modifier

##### Constructor

##### Description

protected

`LevelController(LevelModel model)`

### Uses of LevelModel in com.example.project.controllers.gameScreens.animations

#### Constructors in com.example.project.controllers.gameScreens.animations with parameters of type LevelModel

##### Modifier

##### Constructor

##### Description

`LevelScoreSequence(List<LetterTileController> wordTileControllers, LevelModel levelModel, javafx.scene.control.Label comboCountLabel, javafx.scene.control.Label multiplierLabel)`

Constructor.

### Uses of LevelModel in com.example.project.services

#### Methods in com.example.project.services that return LevelModel

##### Modifier and Type

##### Method

##### Description

## LevelModel

`Session.getLevelModel()`

### Methods in `com.example.project.services` with parameters of type `LevelModel`

#### Modifier and Type

#### Method

#### Description

`void`

`Session.setLevelModel(LevelModel levelModel)`

sets the session's level model

## **Uses of Class com.example.project.models.gameScreens.LoginModel**

No usage of com.example.project.models.gameScreens.LoginModel

---

Copyright © 2025. All rights reserved.

## **Uses of Class com.example.project.models.gameScreens.MainMenuModel**

No usage of com.example.project.models.gameScreens.MainMenuModel

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.models.gameScreens.ScoreChimePlayer

### Packages that use ScoreChimePlayer

#### Package

#### Description

com.example.project.models.gameScreens

### Uses of ScoreChimePlayer in com.example.project.models.gameScreens

#### Methods in com.example.project.models.gameScreens that return ScoreChimePlayer

##### Modifier and Type

##### Method

##### Description

ScoreChimePlayer

**LevelModel.getTileScoreSoundPlayer()**

Gets the tile score sound effect player.

## Uses of Class com.example.project.models.gameScreens.ShopModel

### Packages that use ShopModel

#### Package

#### Description

com.example.project.controllers.gameScreens

## Uses of ShopModel in com.example.project.controllers.gameScreens

### Constructors in com.example.project.controllers.gameScreens with parameters of type ShopModel

#### Modifier

#### Constructor

#### Description

protected

`ShopController(ShopModel model, UpgradeTileGroup players, UpgradeTileGroup shopItems, Logger logger)`

protected constructor for unit testing with mock model injection.

## Package com.example.project.models.gameScreens

package com.example.project.models.gameScreens

### Related Packages

#### Package

#### Description

[com.example.project.models](#)

[com.example.project.models.popups](#)

[com.example.project.models.tileGroups](#)

[com.example.project.models.tiles](#)

### All Classes and Interfaces

### Classes

### Enum Classes

#### Class

#### Description

**DailyRewardModel**

Model for handling daily reward logic and state.

**DailyRewardType**

Enum representing the possible daily reward outcomes.

**GameScreenModel**

Game Screen Model.

**LevelModel**

Represents the level model.

**LoginModel**

Login model class.

**MainMenuModel**

main menu model class

**ScoreChimePlayer**

plays a note from the F major scale ascending in pitch each time for a nice score combo sound.

**ShopModel**

Shop Model.

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Class GameScreenController

java.lang.Object<sup>2</sup>  
com.example.project.controllers.gameScreens.GameScreenController

### Direct Known Subclasses:

DailyRewardController, LevelController, LoginController, MainMenuController, ShopController

```
public abstract class GameScreenController  
extends Object2
```

Game screen controller that has some startup on screen thing todo. Level, shop controllers.

### Nested Class Summary

#### Nested Classes

##### Modifier and Type

##### Class

##### Description

protected static final record

GameScreenController.SidebarLoaded

### Field Summary

#### Fields

##### Modifier and Type

##### Field

##### Description

protected Logger

logger

### Constructor Summary

#### Constructors

##### Modifier

##### Constructor

##### Description

protected

GameScreenController()

### Method Summary

#### All Methods

#### Instance Methods

#### Abstract Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

protected GameScreenController.SidebarLoaded

```
loadSidebar()
```

```
abstract void
```

```
onSceneChangedToThis()
```

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

#### **Methods inherited from class java.lang.Object**

```
clone✉, equals✉, finalize✉, getClass✉, hashCode✉, notify✉, notifyAll✉, toString✉, wait✉, wait✉, wait✉
```

### **Field Details**

#### **logger**

```
protected Logger logger
```

### **Constructor Details**

#### **GameScreenController**

```
protected GameScreenController()
```

### **Method Details**

#### **onSceneChangedToThis**

```
public abstract void onSceneChangedToThis()
```

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

#### **loadSidebar**

```
protected GameScreenController.SidebarLoaded loadSidebar()
```

## Hierarchy For Package com.example.project.models.gameScreens

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
  - [com.example.project.models.gameScreens.DailyRewardModel](#)
  - [com.example.project.models.gameScreens.GameScreenModel](#)
    - [com.example.project.models.gameScreens.LevelModel](#)
    - [com.example.project.models.gameScreens.LoginModel](#)
    - [com.example.project.models.gameScreens.MainMenuModel](#)
    - [com.example.project.models.gameScreens.ShopModel](#)
  - [com.example.project.models.gameScreens.ScoreChimePlayer](#)

### Enum Class Hierarchy

- [java.lang.Object](#)
  - [java.lang.Enum](#) (implements [java.lang.Comparable](#), [java.lang.constant.Constable](#), [java.io.Serializable](#))
    - [com.example.project.models.gameScreens.DailyRewardType](#)

---

Copyright © 2025. All rights reserved.

## Uses of Package com.example.project.models.gameScreens

### Packages that use com.example.project.models.gameScreens

#### Package

#### Description

com.example.project.controllers.gameScreens

com.example.project.controllers.gameScreens.animations

com.example.project.models.gameScreens

com.example.project.services

### Classes in com.example.project.models.gameScreens used by com.example.project.controllers.gameScreens

#### Class

#### Description

**LevelModel**

Represents the level model.

**ShopModel**

Shop Model.

### Classes in com.example.project.models.gameScreens used by com.example.project.controllers.gameScreens.animation

#### Class

#### Description

**LevelModel**

Represents the level model.

### Classes in com.example.project.models.gameScreens used by com.example.project.models.gameScreens

#### Class

#### Description

**DailyRewardType**

Enum representing the possible daily reward outcomes.

**GameScreenModel**

Game Screen Model.

**ScoreChimePlayer**

plays a note from the F major scale ascending in pitch each time for a nice score combo sound.

### Classes in com.example.project.models.gameScreens used by com.example.project.services

#### Class

#### Description

**LevelModel**

Represents the level model.

Module com.example.project

## Package com.example.project.models

package com.example.project.models

### Related Packages

#### Package

#### Description

[com.example.project](#)

[com.example.project.models.gameScreens](#)

[com.example.project.models.popups](#)

[com.example.project.models.tileGroups](#)

[com.example.project.models.tiles](#)

[com.example.project.controllers](#)

[com.example.project.services](#)

### Classes

#### Class

#### Description

#### User

Represents a user in the application that can log in.

## Hierarchy For Package com.example.project.models

### Package Hierarchies:

All Packages

### Class Hierarchy

- [java.lang.Object](#) ↗
  - [com.example.project.models.User](#)

---

Copyright © 2025. All rights reserved.

## Uses of Package com.example.project.models

### Packages that use com.example.project.models

#### Package

#### Description

com.example.project.services

com.example.project.services.sqlite.dAOs

### Classes in com.example.project.models used by com.example.project.services

#### Class

#### Description

User

Represents a user in the application that can log in.

### Classes in com.example.project.models used by com.example.project.services.sqlite.dAOs

#### Class

#### Description

User

Represents a user in the application that can log in.

**Module** com.example.project  
**Package** com.example.project.models.popups

## Class DefinitionPopup

java.lang.Object ↗  
  com.example.project.models.popups.PopupModel  
    com.example.project.models.popups.DefinitionPopup

```
public class DefinitionPopup  
extends PopupModel
```

Represents Definition window called when a word is played

### Property Summary

#### Properties

Type

Property

Description

javafx.beans.property.ReadOnlyStringProperty

definition

javafx.beans.property.ReadOnlyStringProperty

word

### Field Summary

Fields inherited from class com.example.project.models.popups.PopupModel

paperSoundPlayer, reversePaperSoundPlayer

### Constructor Summary

#### Constructors

Constructor

Description

DefinitionPopup()

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

javafx.beans.property.ReadOnlyStringProperty

definitionProperty()

String ↗

getFXMLPath()

gets FXML path

```
javafx.beans.property.ReadOnlyBooleanProperty
```

```
getIsDefinitionActive()
```

```
void
```

```
setIsDefinitionActive(boolean newState)
```

```
toggle isDefinitionActive
```

```
void
```

```
setPopup(String currentWord)
```

```
javafx.beans.property.ReadOnlyStringProperty
```

```
wordProperty()
```

## Methods inherited from class com.example.project.models.popups.PopupModel

```
getPaperSoundPlayer, getReversePaperSoundPlayer
```

## Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

## Property Details

### definition

```
public javafx.beans.property.ReadOnlyStringProperty definitionProperty
```

**See Also:**

```
definitionProperty()
```

### word

```
public javafx.beans.property.ReadOnlyStringProperty wordProperty
```

**See Also:**

```
wordProperty()
```

## Constructor Details

### DefinitionPopup

```
public DefinitionPopup()
```

## Method Details

### definitionProperty

```
public javafx.beans.property.ReadOnlyStringProperty definitionProperty()
```

**Returns:**

```
returns property of definition
```

### wordProperty

```
public javafx.beans.property.ReadOnlyStringProperty wordProperty()
```

**Returns:**

returns property of word

**getIsDefinitionActive**

```
public javafx.beans.property.ReadOnlyBooleanProperty getIsDefinitionActive()
```

**Returns:**

returns boolean

**setIsDefinitionActive**

```
public void setIsDefinitionActive(boolean newState)
```

toggle isDefinitionActive

**Parameters:**

newState - current state of definition window

**setPopup**

```
public void setPopup(String2 currentWord)
```

**Parameters:**

currentWord - Word, when OnPlayButton is passed

**getFXMLPath**

```
public String2 getFXMLPath()
```

**Description copied from class:** [PopupModel](#)

gets FXML path

**Specified by:**

[getFXMLPath](#) in class [PopupModel](#)

**Returns:**

string path

**Module** com.example.project

**Package** com.example.project.models.popups

## Class PopupModel

java.lang.Object<sup>✉</sup>  
com.example.project.models.popups.PopupModel

### Direct Known Subclasses:

DefinitionPopup

---

```
public abstract class PopupModel  
extends Object✉
```

Parent class for popup windows

## Field Summary

### Fields

#### Modifier and Type

#### Field

#### Description

```
protected final GameSoundPlayer
```

**paperSoundPlayer**

```
protected final GameSoundPlayer
```

**reversePaperSoundPlayer**

## Constructor Summary

### Constructors

#### Constructor

#### Description

**PopupModel()**

## Method Summary

### All Methods

### Instance Methods

### Abstract Methods

### Concrete Methods

#### Modifier and Type

#### Method

#### Description

```
abstract String✉
```

**getFXMLPath()**

gets FXML path

**GameSoundPlayer**

**getPaperSoundPlayer()**

gets paper sound player

**GameSoundPlayer**

**getReversePaperSoundPlayer()**

gets reversed paper sound player

## Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Details

### **paperSoundPlayer**

`protected final GameSoundPlayer paperSoundPlayer`

### **reversePaperSoundPlayer**

`protected final GameSoundPlayer reversePaperSoundPlayer`

## Constructor Details

### **PopupModel**

`public PopupModel()`

## Method Details

### **getFXMLPath**

`public abstract String getFXMLPath()`

gets FXML path

**Returns:**

string path

### **getPaperSoundPlayer**

`public GameSoundPlayer getPaperSoundPlayer()`

gets paper sound player

**Returns:**

sound player

### **getReversePaperSoundPlayer**

`public GameSoundPlayer getReversePaperSoundPlayer()`

gets reversed paper sound player

**Returns:**

sound player

## Uses of Class com.example.project.models.popups.DefinitionPopup

### Packages that use DefinitionPopup

#### Package

#### Description

com.example.project.controllers.popupControllers

### Uses of DefinitionPopup in com.example.project.controllers.popupControllers

#### Methods in com.example.project.controllers.popupControllers that return DefinitionPopup

##### Modifier and Type

##### Method

##### Description

DefinitionPopup

DefinitionController.getModel()

#### Methods in com.example.project.controllers.popupControllers with parameters of type DefinitionPopup

##### Modifier and Type

##### Method

##### Description

void

DefinitionController.initialize(DefinitionPopup model)

## Uses of Class com.example.project.models.popups.PopupModel

### Packages that use PopupModel

#### Package

#### Description

com.example.project.controllers.popupControllers

com.example.project.models.popups

com.example.project.services

## Uses of PopupModel in com.example.project.controllers.popupControllers

### Classes in com.example.project.controllers.popupControllers with type parameters of type PopupModel

#### Modifier and Type

#### Class

#### Description

class

PopupController<P extends PopupModel>

Parent class for PopupControllers

### Fields in com.example.project.controllers.popupControllers declared as PopupModel

#### Modifier and Type

#### Field

#### Description

protected P

PopupController.model

## Uses of PopupModel in com.example.project.models.popups

### Subclasses of PopupModel in com.example.project.models.popups

#### Modifier and Type

#### Class

#### Description

class

DefinitionPopup

Represents Definition window called when a word is played

## Uses of PopupModel in com.example.project.services

### Methods in com.example.project.services with type parameters of type PopupModel

#### Modifier and Type

#### Method

#### Description

<C extends PopupController<P>, P extends PopupModel>

C

**PopupLoader.createPopupController(P popupModel)**

create popup controller.

Module com.example.project

## Package com.example.project.models.popups

package com.example.project.models.popups

### Related Packages

#### Package

#### Description

[com.example.project.models](#)

[com.example.project.models.gameScreens](#)

[com.example.project.models.tileGroups](#)

[com.example.project.models.tiles](#)

### Classes

#### Class

#### Description

[DefinitionPopup](#)

Represents Definition window called when a word is played

[PopupModel](#)

Parent class for popup windows

**Module** com.example.project

**Package** com.example.project.controllers.gameScreens

## Class LevelController

```
java.lang.Object
  com.example.project.controllers.gameScreens.GameScreenController
    com.example.project.controllers.gameScreens.LevelController
```

```
public class LevelController
extends GameScreenController
```

Controller for the level screen.

### Nested Class Summary

**Nested classes/interfaces inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
GameScreenController.SidebarLoaded
```

### Field Summary

**Fields inherited from class com.example.project.controllers.gameScreens.GameScreenController**

```
logger
```

### Constructor Summary

#### Constructors

Modifier

Constructor

Description

```
LevelController()
```

Constructor only called once each time application opened.

protected

```
LevelController(LevelModel model)
```

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

Modifier and Type

Method

Description

```
void
```

```
initialize()
```

```
protected void
```

```
onConfirmRedrawButton()
```

Handle redraw confirm button.

```
protected void
```

### **onPlayButton()**

Handle play button

### **protected void onRedrawButton()**

redraw button opens or cancels the redraw.

void

### **onSceneChangedToThis()**

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

### **protected void onSkipButton()**

## **Methods inherited from class com.example.project.controllers.gameScreens.GameScreenController**

loadSidebar

## **Methods inherited from class java.lang.Object**

clone<sup>✉</sup>, equals<sup>✉</sup>, finalize<sup>✉</sup>, getClass<sup>✉</sup>, hashCode<sup>✉</sup>, notify<sup>✉</sup>, notifyAll<sup>✉</sup>, toString<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>, wait<sup>✉</sup>

## **Constructor Details**

### **LevelController**

public LevelController()

Constructor only called once each time application opened.

### **LevelController**

protected LevelController(LevelModel model)

## **Method Details**

### **initialize**

public void initialize()

#### **See Also:**

for the money binding

### **onSceneChangedToThis**

public void onSceneChangedToThis()

#### **Description copied from class: GameScreenController**

Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.

#### **Specified by:**

onSceneChangedToThis in class GameScreenController

### **onPlayButton**

protected void onPlayButton()

Handle play button

## **onSkipButton**

```
protected void onSkipButton()
```

## **onRedrawButton**

```
protected void onRedrawButton()
```

redraw button opens or cancels the redraw.

## **onConfirmRedrawButton**

```
protected void onConfirmRedrawButton()
```

Handle redraw confirm button.

## Hierarchy For Package com.example.project.models.popups

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#) ↗
  - [com.example.project.models.popups.PopupModel](#)
    - [com.example.project.models.popups.DefinitionPopup](#)

## Uses of Package com.example.project.models.popups

### Packages that use com.example.project.models.popups

#### Package

#### Description

[com.example.project.controllers.popupControllers](#)

[com.example.project.models.popups](#)

[com.example.project.services](#)

### Classes in com.example.project.models.popups used by com.example.project.controllers.popupControllers

#### Class

#### Description

[DefinitionPopup](#)

Represents Definition window called when a word is played

[PopupModel](#)

Parent class for popup windows

### Classes in com.example.project.models.popups used by com.example.project.models.popups

#### Class

#### Description

[PopupModel](#)

Parent class for popup windows

### Classes in com.example.project.models.popups used by com.example.project.services

#### Class

#### Description

[PopupModel](#)

Parent class for popup windows

**Module** com.example.project

**Package** com.example.project.models.tileGroups

## Class LetterTileGroup

```
java.lang.Object
  com.example.project.models.tileGroups.TileGroup<LetterTileModel,LetterTileController>
    com.example.project.models.tileGroups.LetterTileGroup
```

```
public class LetterTileGroup
extends TileGroup<LetterTileModel,LetterTileController>
```

tile group that observes an observable list and updates the ui Tile Controller's nodes into their EmptyTileSlotController nodes.

### Field Summary

#### Fields inherited from class com.example.project.models.tileGroups.TileGroup

container, onClickAction, tileControllerFactory, tileControllers

### Constructor Summary

#### Constructors

##### Modifier

##### Constructor

##### Description

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<?
<LetterTileController> onClickAction)
```

Constructor

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<?
<LetterTileController> onClickHandler, List<?><Runnable> afterSyncActions)
```

Constructor

protected

```
LetterTileGroup(int numberOfEmptyTileSlots, javafx.scene.layout.Pane container,
javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList, Consumer<?
<LetterTileController> onClickHandler, List<?><Runnable> afterSyncActions, TileControllerFactory factory)
```

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

protected void

`updateVisuals()`

#### Methods inherited from class com.example.project.models.tileGroups.TileGroup

`getControllers`, `syncTiles`

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Details

### LetterTileGroup

```
protected LetterTileGroup(int numberEmptyTileSlots,
                         javafx.scene.layout.Pane container,
                         javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList,
                         Consumer<LetterTileController> onClickHandler,
                         List<Runnable> afterSyncActions,
                         TileControllerFactory factory)
```

### LetterTileGroup

```
public LetterTileGroup(int numberEmptyTileSlots,
                      javafx.scene.layout.Pane container,
                      javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList,
                      Consumer<LetterTileController> onClickHandler,
                      List<Runnable> afterSyncActions)
```

Constructor

#### Parameters:

numberEmptyTileSlots - number of max tiles in group (empty slots)

container - container to place all in.

observedList - the observed list.

onClickHandler - On tile click action.

afterSyncActions - additional synchronisation actions that need to happen when this observed list changes.

### LetterTileGroup

```
public LetterTileGroup(int numberEmptyTileSlots,
                      javafx.scene.layout.Pane container,
                      javafx.beans.property.ReadOnlyListProperty<LetterTileModel> observedList,
                      Consumer<LetterTileController> onClickAction)
```

Constructor

#### Parameters:

numberEmptyTileSlots - number of max tiles in group (empty slots)

container - container to place all in.

observedList - the observed list.

onClickAction - On tile click action.

## Method Details

### updateVisuals

```
protected void updateVisuals()
```

#### Specified by:

[updateVisuals](#) in class [TileGroup<LetterTileModel,LetterTileController>](#)

**Module** com.example.project

**Package** com.example.project.models.tileGroups

## Class TileGroup<modelType extends TileModel, controllerType extends TileController<modelType>>

java.lang.Object<sup>2</sup>  
com.example.project.models.tileGroups.TileGroup<modelType,controllerType>

### Type Parameters:

modelType - tile model type.

controllerType - controller type.

### Direct Known Subclasses:

LetterTileGroup, UpgradeTileGroup

---

```
public abstract class TileGroup<modelType extends TileModel, controllerType extends TileController<modelType>>
extends Object2
```

Tile Group abstract class holds a row or column of tiles.

## Field Summary

### Fields

#### Modifier and Type

#### Field

#### Description

protected javafx.scene.layout.Pane  
**container**

protected Consumer<sup>2</sup><controllerType>  
**onClickAction**

protected TileControllerFactory  
**tileControllerFactory**

protected final List<sup>2</sup><controllerType>  
**tileControllers**

## Method Summary

### All Methods

### Instance Methods

### Abstract Methods

### Concrete Methods

#### Modifier and Type

#### Method

#### Description

List<sup>2</sup><controllerType>

**getControllers()**

Gets the groups tile controllers.

void

**syncTiles()**

Sync tiles in this tile groups controller.

protected abstract void

```
updateVisuals()
```

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

### Field Details

#### container

```
protected javafx.scene.layout.Pane container
```

#### tileControllers

```
protected final List<controllerType extends TileController<modelType>> tileControllers
```

#### tileControllerFactory

```
protected TileControllerFactory tileControllerFactory
```

#### onClickAction

```
protected Consumer<controllerType extends TileController<modelType>> onClickAction
```

### Method Details

#### updateVisuals

```
protected abstract void updateVisuals()
```

#### syncTiles

```
public void syncTiles()
```

Sync tiles in this tile groups controller.

#### getControllers

```
public List<controllerType> getControllers()
```

Gets the groups tile controllers.

**Returns:**

the letter tile groups controllers

**Module** com.example.project  
**Package** com.example.project.models.tileGroups

## Class UpgradeTileGroup

java.lang.Object  
  com.example.project.models.tileGroups.TileGroup<UpgradeTileModel,UpgradeTileController>  
    com.example.project.models.tileGroups.UpgradeTileGroup

```
public class UpgradeTileGroup  
extends TileGroup<UpgradeTileModel,UpgradeTileController>
```

tile group that observes an observable list and updates the ui nodes for some game tiles.

### Field Summary

#### Fields inherited from class com.example.project.models.tileGroups.TileGroup

container, onClickAction, tileControllerFactory, tileControllers

### Constructor Summary

#### Constructors

##### Constructor

##### Description

**UpgradeTileGroup**(javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> observedList)

Constructor with no additional sync action parameter.

**UpgradeTileGroup**(javafx.scene.layout.Pane container,  
javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> observedList, Consumer<UpgradeTileController> onClickAction)

Constructor.

### Method Summary

#### All Methods

#### Instance Methods

#### Concrete Methods

##### Modifier and Type

##### Method

##### Description

protected void

**updateVisuals()**

#### Methods inherited from class com.example.project.models.tileGroups.TileGroup

getControllers, syncTiles

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Details

#### UpgradeTileGroup

```
public UpgradeTileGroup(javafx.scene.layout.Pane container,
                        javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> observedList,
                        Consumer<UpgradeTileController> onClickAction)
```

Constructor.

**Parameters:**

container - container all tiles are in.

observedList - the observed list that when changes this tile group will sync.

onClickAction - additional synchronisation actions that need to be called.

## UpgradeTileGroup

```
public UpgradeTileGroup(javafx.scene.layout.Pane container,
                        javafx.beans.property.ReadOnlyListProperty<UpgradeTileModel> observedList)
```

Constructor with no additional sync action parameter.

**Parameters:**

container - container tiles are in.

observedList - the observed list that when changes this tile group will sync.

## Method Details

### updateVisuals

```
protected void updateVisuals()
```

**Specified by:**

`updateVisuals` in class `TileGroup<UpgradeTileModel, UpgradeTileController>`

## Uses of Class **com.example.project.models.tileGroups.LetterTileGroup**

No usage of com.example.project.models.tileGroups.LetterTileGroup

---

Copyright © 2025. All rights reserved.

## Uses of Class com.example.project.models.tileGroups.TileGroup

### Packages that use TileGroup

#### Package

#### Description

com.example.project.models.tileGroups

## Uses of TileGroup in com.example.project.models.tileGroups

### Subclasses of TileGroup in com.example.project.models.tileGroups

#### Modifier and Type

#### Class

#### Description

class

#### LetterTileGroup

tile group that observes an observable list and updates the ui Tile Controller's nodes into their EmptyTileSlotController nodes.

class

#### UpgradeTileGroup

tile group that observes an observable list and updates the ui nodes for some game tiles.

## Uses of Class com.example.project.models.tileGroups.UpgradeTileGroup

### Packages that use UpgradeTileGroup

#### Package

#### Description

com.example.project.controllers.gameScreens

### Uses of UpgradeTileGroup in com.example.project.controllers.gameScreens

#### Constructors in com.example.project.controllers.gameScreens with parameters of type UpgradeTileGroup

##### Modifier

##### Constructor

##### Description

protected

`ShopController(ShopModel model, UpgradeTileGroup players, UpgradeTileGroup shopItems, Logger logger)`

protected constructor for unit testing with mock model injection.

Module com.example.project

## Package com.example.project.models.tileGroups

package com.example.project.models.tileGroups

### Related Packages

#### Package

#### Description

[com.example.project.models](#)

[com.example.project.models.gameScreens](#)

[com.example.project.models.popups](#)

[com.example.project.models.tiles](#)

### Classes

#### Class

#### Description

**LetterTileGroup**

tile group that observes an observable list and updates the ui Tile Controller's nodes into their EmptyTileSlotController nodes.

**TileGroup<modelType extends TileModel, controllerType extends TileController<modelType>>**

Tile Group abstract class holds a row or column of tiles.

**UpgradeTileGroup**

tile group that observes an observable list and updates the ui nodes for some game tiles.

---

Copyright © 2025. All rights reserved.

## Hierarchy For Package com.example.project.models.tileGroups

### Package Hierarchies:

[All Packages](#)

### Class Hierarchy

- [java.lang.Object](#)
- [com.example.project.models.tileGroups.TileGroup<modelType,controllerType>](#)
  - [com.example.project.models.tileGroups.LetterTileGroup](#)
  - [com.example.project.models.tileGroups.UpgradeTileGroup](#)