

AnimationWrapper

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| AnimationWrapper() | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| setOnFinished(EventHandler) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| play() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| getChildren() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 21 of 21 | 0% | 0 of 0 | n/a | 4 | 4 | 8 | 8 | 4 | 4 |

Created with JaCoCo 0.8.12.202403310830

AnimationWrapper.java

```
1. package com.example.project.controllers.gameScreens.animations;
2.
3. import javafx.animation.Animation;
4. import javafx.animation.SequentialTransition;
5. import javafx.collections.ObservableList;
6. import javafx.event.ActionEvent;
7. import javafx.event.EventHandler;
8.
9. /**
10. * Animation wrapper class for our projects animations that get used.
11. */
12. public abstract class AnimationWrapper
13. {
14.     protected final SequentialTransition sequentialAnimation;
15.
16.     protected AnimationWrapper() {
17.         this.sequentialAnimation = new SequentialTransition();
18.     }
19.
20.     /**
21.      * play.
22.     */
23.     public void play() {
24.         sequentialAnimation.play();
25.     }
26.
27.     /**
28.      * Set on finished.
29.      * @param handler actions to do.
30.     */
31.     public void setOnFinished(EventHandler<ActionEvent> handler) {
32.         sequentialAnimation.setOnFinished(handler);
33.     }
34.
35.     /**
36.      * returns animations children.
37.      * @return list of animations.
38.     */
39.     public ObservableList<Animation> getChildren(){
40.         return this.sequentialAnimation.getChildren();
41.     }
42. }
```

TextEmphasisAnimation

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| TextEmphasisAnimation(Label, Paint, Paint, Duration) | | 0% | n/a | 1 | 1 | 16 | 16 | 1 | 1 | 1 |
| lambda\$new\$1(Label, Paint, ActionEvent) | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$new\$0(Label, Paint, ActionEvent) | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 82 of 82 | 0% | 0 of 0 | n/a | 3 | 3 | 16 | 16 | 3 | 3 |

Created with JaCoCo 0.8.12.202403310830

TileModel.java

```
1. package com.example.project.models.tiles;
2.
3. import com.example.project.services.sound.GameSoundPlayer;
4.
5. /**
6. * Parent tile class for objects in the scene that are tiles.
7. */
8. public abstract class TileModel
9.
10. {
11.     /**
12.      * gets the fxml path.
13.      * @return string path.
14.      */
15.     public abstract String getFXMLPath();
16.
17.     protected final GameSoundPlayer hoverSoundPlayer = new GameSoundPlayer("/com/example/project/Sounds/hover.wav", -80f);
18.     protected final GameSoundPlayer ClackSoundPlayer = new GameSoundPlayer("/com/example/project/Sounds/clack3.wav", 6f);
19.
20.     /**
21.      * gets the hover sound player
22.      * @return sound player
23.      */
24.     public GameSoundPlayer getHoverSoundPlayer() { return hoverSoundPlayer; }
25.
26.     /**
27.      * gets the sound player that plays the click sound.
28.      * @return sound player.
29.      */
30.     public GameSoundPlayer getClackSoundPlayer() {return ClackSoundPlayer;}
31.
32.     /**
33.      * corner radius (rounded corners).
34.      */
35.     public static final int CORNER_RADIUS = 20;
}
```

UpgradeEffects

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|----------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| braceletEffect(LevelModel) | | 0% | | 0% | 4 | 4 | 10 | 10 | 1 | 1 |
| glassesEffect(LevelModel) | | 0% | | 0% | 3 | 3 | 7 | 7 | 1 | 1 |
| buttonEffect(LevelModel) | | 0% | | 0% | 3 | 3 | 6 | 6 | 1 | 1 |
| diceEffect(LevelModel) | | 0% | | n/a | 1 | 1 | 5 | 5 | 1 | 1 |
| mirrorEffect(LevelModel) | | 0% | | 0% | 2 | 2 | 6 | 6 | 1 | 1 |
| coinEffect(LevelModel) | | 0% | | 0% | 2 | 2 | 6 | 6 | 1 | 1 |
| UpgradeEffects() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 200 of 200 | 0% | 18 of 18 | 0% | 16 | 16 | 41 | 41 | 7 | 7 |

Created with JaCoCo 0.8.12.202403310830

UpgradeEffects.java

```
1. package com.example.project.models.tiles;
2.
3. import com.example.project.models.gameScreens.LevelModel;
4. import com.example.project.services.Session;
5.
6. import java.util.Random;
7.
8. /**
9. * upgrade effect methods
10.*/
11. public class UpgradeEffects {
12. {
13. /**
14. * Adds +2 to the score multiplier for every identical pair of letters next to each other
15. * @param levelModel model to use.
16. */
17. public static void glassesEffect(LevelModel levelModel) {
18.     // for each letter in word, if the current letter matches the previous letter add 2 to the modifier
19.     ◆◆ for (int i = 1; i < levelModel.getWordWindowTilesProperty().size(); i++) {
20.         Character previousLetter = levelModel.getWordWindowTilesProperty().get(i-1).getLetter();
21.         Character currentLetter = levelModel.getWordWindowTilesProperty().get(i).getLetter();
22.         ◆◆ if (previousLetter.equals(currentLetter)) {
23.             int newMulti = levelModel.wordMultiProperty().get() + 2;
24.             levelModel.setWordMulti(newMulti);
25.         }
26.     }
27. }
28.
29. /**
30. * Value is doubled for a random letter in the played word
31. * @param levelModel model to use.
32. */
33. public static void diceEffect(LevelModel levelModel) {
34.     Random random = new Random();
35.     int randomNum = random.nextInt(levelModel.getWordWindowTilesProperty().size());
36.     int newScore = levelModel.wordPointsProperty().get() + levelModel.getWordWindowTilesProperty().get(randomNum).getValue();
37.     levelModel.setWordPoints(newScore);
38. }
39.
40. /**
41. * 20% Chance the total word score is doubled
42. * @param levelModel model to use.
43. */
44. public static void coinEffect(LevelModel levelModel) {
45.     Random random = new Random();
46.     double chance = 0.2;
47.     ◆◆ if (random.nextDouble() < chance) {
48.         int newScore = levelModel.wordPointsProperty().get() * 2;
49.         levelModel.setWordPoints(newScore);
50.     }
51. }
52.
53. /**
54. * If played word is a palindrome, doubles word score.
55. * @param levelModel model to use.
56. */
57. public static void mirrorEffect(LevelModel levelModel) {
58.     String word = levelModel.getCurrentWord();
59.     String reversedWord = new StringBuilder(word).reverse().toString();
60.
61.     ◆◆ if (word.equals(reversedWord)) {
62.         int newScore = levelModel.wordPointsProperty().get() * 2;
63.         levelModel.setWordPoints(newScore);
64.     }
65. }
66.
67. /**
68. * Add +1 to score multiplier for every consecutive letter alphabetical order.
69. * @param levelModel model to use.
70. */
71. public static void braceletEffect(LevelModel levelModel) {
72.
73.     String alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
74.
75.     // for each letter in the word, if the next letter matches the next letter in the alphabet, add 1 to modifier
76.     ◆◆ for (int i = 0; i < levelModel.getWordWindowTilesProperty().size()-1; i++) {
77.
78.         Character currentLetter = levelModel.getWordWindowTilesProperty().get(i).getLetter();
79.         Character nextLetter = levelModel.getWordWindowTilesProperty().get(i+1).getLetter();
80.
81.         ◆◆ if (currentLetter.equals('Z')) { continue; }
82.
83.         Character nextAlphabeticalLetter = alphabet.charAt(alphabet.indexOf(currentLetter)+1);
```

```
84.
85.         if (nextLetter.equals(nextAlphabeticalLetter)) {
86.             int newMulti = levelModel.wordMultiProperty().get() + 1;
87.             levelModel.setWordMulti(newMulti);
88.         }
89.     }
90.
91.
92. /**
93. * adds 5 to the word score for every x in the tile rack.
94. * @param levelModel model to use effect on.
95. */
96. public static void buttonEffect(LevelModel levelModel) {
97.     for (LetterTileModel tile : levelModel.getTileRackTilesProperty()) {
98.         if (tile.getLetter().equals('X')) {
99.             int newScore = levelModel.wordPointsProperty().get() + 5;
100.            levelModel.setWordPoints(newScore);
101.        }
102.    }
103. }
104. }
```

Created with JaCoCo 0.8.12.202403310830

UpgradeTileModel.UpgradeBuilder

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| description(String) | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| name(String) | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| imagePath(String) | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| cost(double) | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| upgradeEffect(Consumer) | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| build() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| UpgradeTileModel.UpgradeBuilder() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 33 | 100% | 0 of 0 | n/a | 0 | 7 | 0 | 7 | 0 | 7 |

Created with JaCoCo 0.8.12.202403310830

UpgradeTileModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| UpgradeTileModel(UpgradeTileModel.UpgradeBuilder) | 100% | n/a | 0 | 1 | 0 | 7 | 0 | 1 | 0 | 1 |
| runUpgradeEffect(LevelModel) | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 1 |
| getName() | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| getDescription() | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| getCost() | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| getAbilityImagePath() | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| getFXMLPath() | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 42 | 100% | 0 of 0 | n/a | 0 | 7 | 0 | 14 | 0 | 7 |

Created with JaCoCo 0.8.12.202403310830

UpgradeTileModel.java

```
1. package com.example.project.models.tiles;
2.
3. import com.example.project.models.gameScreens.LevelModel;
4.
5. import java.util.function.Consumer;
6.
7. /**
8. * Represents the upgrade tiles sold at the shop.
9. */
10. public class UpgradeTileModel extends TileModel
11. {
12.     private final String description;
13.     private final String name;
14.     private final String imagePath;
15.     private final double cost;
16.     private final Consumer<LevelModel> upgradeEffect;
17.
18. /**
19. * public upgrade constructor.
20. * @param builder upgrade builder
21. */
22. public UpgradeTileModel(UpgradeBuilder builder)
23. {
24.     this.description = builder.description;
25.     this.name = builder.name;
26.     this.imagePath = builder.imagePath;
27.     this.cost = builder.cost;
28.     this.upgradeEffect = builder.upgradeEffect;
29. }
30.
31. /**
32. * builder class for the upgrades
33. */
34. public static class UpgradeBuilder {
35.     private String description;
36.     private String name;
37.     private String imagePath;
38.     private double cost;
39.     private Consumer<LevelModel> upgradeEffect;
40.
41. /**
42. * upgrade description builder
43. * @param description new upgrade description
44. * @return description
45. */
46. public UpgradeBuilder description(String description) {this.description = description; return this; }
47.
48. /**
49. * upgrade name builder
50. * @param name new upgrade name
51. * @return name
52. */
53. public UpgradeBuilder name(String name) {this.name = name; return this; }
54.
55. /**
56. * upgrade image path builder
57. * @param imagePath new upgrade image path
58. * @return image path
59. */
60. public UpgradeBuilder imagePath(String imagePath) {this.imagePath = imagePath; return this; }
61.
62. /**
63. * upgrade cost builder
64. * @param cost new upgrade cost
65. * @return cost
66. */
67. public UpgradeBuilder cost(double cost) {this.cost = cost; return this; }
68.
69. /**
70. * upgrade effect builder
71. * @param upgradeEffect new upgrade effect
72. * @return upgrade effect
73. */
74. public UpgradeBuilder upgradeEffect(Consumer<LevelModel> upgradeEffect) { this.upgradeEffect = upgradeEffect; return this; }
75.
76. /**
77. * build upgrade
78. * @return new upgrade
79. */
80. public UpgradeTileModel build() { return new UpgradeTileModel(this); }
81. }
82.
83. /**
84. * gets the name.
85. * @return return name.
```

```
86.      */
87.     public String getName(){
88.         return this.name;
89.     }
90.
91.    /**
92.     * gets the description.
93.     * @return description.
94.     */
95.    public String getDescription(){
96.        return this.description;
97.    }
98.
99.    /**
100.     * gets the cost
101.     * @return return cost.
102.     */
103.    public double getCost()
104.    {
105.        return this.cost;
106.    }
107.
108.    /**
109.     * gets upgrade tile image path.
110.     * @return returns loaded Image for this tile.
111.     */
112.    public String getAbilityImagePath() { return imagePath; }
113.
114.    /**
115.     * gets the effect for this tile.
116.     * @param model level model.
117.     */
118.    public void runUpgradeEffect(LevelModel model)
119.    {
120.        upgradeEffect.accept(model);
121.    }
122.
123.    @Override
124.    public String getFXMLPath() {
125.        return "/com/example/project/SingleTiles/upgradeTileView.fxml";
126.    }
127.}
```

com.example.project.models.tiles

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C UpgradeEffects | | 0% | | 0% | 16 | 16 | 41 | 41 | 7 | 7 | 1 | 1 |
| C ScrabbleTileProvider | | 100% | | 100% | 0 | 7 | 0 | 45 | 0 | 5 | 0 | 1 |
| C UpgradeTileModel | | 100% | | n/a | 0 | 7 | 0 | 14 | 0 | 7 | 0 | 1 |
| C UpgradeTileModel.UpgradeBuilder | | 100% | | n/a | 0 | 7 | 0 | 7 | 0 | 7 | 0 | 1 |
| C LetterTileModel | | 100% | | n/a | 0 | 4 | 0 | 8 | 0 | 4 | 0 | 1 |
| C TileModel | | 100% | | n/a | 0 | 3 | 0 | 5 | 0 | 3 | 0 | 1 |
| C EmptyTileSlotModel | | 100% | | n/a | 0 | 4 | 0 | 6 | 0 | 4 | 0 | 1 |
| C ScrabbleTileProvider.ScrabbleTileData | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 200 of 736 | 72% | 18 of 22 | 18% | 16 | 49 | 41 | 127 | 7 | 38 | 1 | 8 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.models.tiles

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| UpgradeEffects.java | | 0% | | 0% | 16 | 16 | 41 | 41 | 7 | 7 | 1 | 1 |
| ScrabbleTileProvider.java | | 100% | | 100% | 0 | 8 | 0 | 46 | 0 | 6 | 0 | 2 |
| UpgradeTileModel.java | | 100% | | n/a | 0 | 14 | 0 | 21 | 0 | 14 | 0 | 2 |
| LetterTileModel.java | | 100% | | n/a | 0 | 4 | 0 | 8 | 0 | 4 | 0 | 1 |
| TileModel.java | | 100% | | n/a | 0 | 3 | 0 | 5 | 0 | 3 | 0 | 1 |
| EmptyTileSlotModel.java | | 100% | | n/a | 0 | 4 | 0 | 6 | 0 | 4 | 0 | 1 |
| Total | 200 of 736 | 72% | 18 of 22 | 18% | 16 | 49 | 41 | 127 | 7 | 38 | 1 | 8 |

Created with [JaCoCo](#) 0.8.12.202403310830

User

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| User(String, String, int) | 100% | | n/a | | 0 | 1 | 0 | 5 | 0 | 1 |
| getHighscore() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| getUsername() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| getPassword() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 23 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 8 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

User.java

```
1. package com.example.project.models;
2.
3. /**
4.  * Represents a user in the application that can log in.
5. */
6. public class User {
7.
8.     private final String username;
9.
10.    private final String password;
11.
12.    private final Integer highscore;
13.
14.    /**
15.     * Constructor.
16.     * @param username username.
17.     * @param password password.
18.     * @param highscore highscore.
19.     */
20.    public User(String username, String password, int highscore)
21.    {
22.        this.hightscore = highscore;
23.        this.username = username;
24.        this.password = password;
25.    }
26.
27.    /**
28.     * get username.
29.     * @return username.
30.     */
31.    public String getUsername(){
32.        return username;
33.    }
34.
35.    /**
36.     * get password.
37.     * @return password.
38.     */
39.    public String getPassword(){
40.        return password;
41.    }
42.
43.    /**
44.     * get highscore.
45.     * @return returns highscore.
46.     */
47.    public int getHighscore(){
48.        return this.hightscore;
49.    }
50.
51. }
```

TextEmphasisAnimation.java

```
1. package com.example.project.controllers.gameScreens.animations;
2.
3. import javafx.animation.PauseTransition;
4. import javafx.animation.ScaleTransition;
5. import javafx.scene.control.Label;
6. import javafx.scene.paint.Paint;
7. import javafx.util.Duration;
8.
9. /**
10. * An animation to quickly enlarge text and then return it to its original scale.
11. */
12. public class TextEmphasisAnimation extends AnimationWrapper
13. {
14.     /**
15.      * create the animations for text growing then shrinking
16.      * @param textToAnimate Label.
17.      * @param colorAtStart initial colour change.
18.      * @param changeColourAtEnd colour change after finished.
19.      * @param timePauseAfterAnim time delay after animation until animation finished.
20.      */
21.     public TextEmphasisAnimation(Label textToAnimate, Paint colorAtStart, Paint changeColourAtEnd, Duration timePauseAfterAnim)
22.     {
23.         super();
24.
25.         var colourTextAtStart = new PauseTransition(Duration.seconds(0));
26.         colourTextAtStart.setOnFinished(e -> textToAnimate.setTextFill(colorAtStart));
27.         this.sequentialAnimation.getChildren().add(colourTextAtStart);
28.
29.         var revertSizeTransition = new ScaleTransition(Duration.seconds(0.2), textToAnimate);
30.         revertSizeTransition.setToY(1);
31.         revertSizeTransition.setToX(1);
32.
33.         var doubleSizeTransition = new ScaleTransition(Duration.seconds(0.2), textToAnimate);
34.         doubleSizeTransition.setToY(2);
35.         doubleSizeTransition.setToX(2);
36.
37.         var pauseAfter = new PauseTransition(timePauseAfterAnim);
38.         pauseAfter.setOnFinished(e -> textToAnimate.setTextFill(changeColourAtEnd));
39.
40.         this.sequentialAnimation.getChildren().add(doubleSizeTransition);
41.         this.sequentialAnimation.getChildren().add(revertSizeTransition);
42.         this.sequentialAnimation.getChildren().add(pauseAfter);
43.     }
44. }
```

com.example.project.models

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed Cxty | Missed Lines | Missed Methods | Missed Classes |
|---------|--------------------------------------|------|-----------------|------|-------------|--------------|----------------|----------------|
| G User | <div style="width: 100%;">100%</div> | 100% | n/a | 0 | 4 | 0 | 8 | 0 |
| Total | 0 of 23 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.models

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|-----------|--------------------------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| User.java | <div style="width: 100%;">100%</div> | | n/a | | 0 | 4 | 0 | 8 | 0 | 4 | 0 | 1 |
| Total | 0 of 23 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 8 | 0 | 4 | 0 | 1 |

Created with JaCoCo 0.8.12.202403310830

UpgradeTiles

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| getUpgradeByName(String) | | 0% | | 0% | 3 | 3 | 5 | 5 | 1 | 1 |
| UpgradeTiles() | | 0% | n/a | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| static {...} | | 100% | n/a | n/a | 0 | 1 | 0 | 38 | 0 | 1 |
| getRandomUpgradeTile() | | 100% | n/a | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| Total | 23 of 145 | 84% | 4 of 4 | 0% | 4 | 6 | 6 | 46 | 2 | 4 |

Created with JaCoCo 0.8.12.202403310830

UpgradeTiles.java

```

1. package com.example.project.services.shopItems;
2.
3. import com.example.project.models.tiles.UpgradeEffects;
4. import com.example.project.models.tiles.UpgradeTileModel;
5.
6. import java.util.Arrays;
7. import java.util.List;
8. import java.util.Objects;
9. import java.util.Random;
10.
11. /**
12.  * Static class to store all the possible upgrade tiles available in the game.
13. */
14. public class UpgradeTiles {
15.
16.     private static final List<UpgradeTileModel> upgradeTiles = Arrays.asList(
17.         new UpgradeTileModel.UpgradeBuilder()
18.             .name("Grandma's Glasses")
19.             .description("Add +2 to the score multiplier for every identical pair of letters next to each other.")
20.             .imagePath("/com/example/project/upgradeTileImages/GrandmasGlasses_small.png")
21.             .cost(2)
22.             .upgradeEffect(UpgradeEffects::glassesEffect)
23.             .build(),
24.
25.         new UpgradeTileModel.UpgradeBuilder()
26.             .name("Loaded Dice")
27.             .description("Value is doubled for a random letter in your word.")
28.             .imagePath("/com/example/project/upgradeTileImages/LoadedDice_small.png")
29.             .cost(2)
30.             .upgradeEffect(UpgradeEffects::diceEffect)
31.             .build(),
32.
33.         new UpgradeTileModel.UpgradeBuilder()
34.             .name("Lucky Coin")
35.             .description("20% Chance your total word score is doubled.")
36.             .imagePath("/com/example/project/upgradeTileImages/LuckyCoin_small.png")
37.             .cost(2)
38.             .upgradeEffect(UpgradeEffects::coinEffect)
39.             .build(),
40.
41.         new UpgradeTileModel.UpgradeBuilder()
42.             .name("Compact Mirror")
43.             .description("Whenever a palindrome is played, double the word's base score. (A palindrome is a word that is spelt the same backwards as forwards, e.g. LEVEL.)")
44.             .imagePath("/com/example/project/upgradeTileImages/CompactMirror.png")
45.             .cost(2)
46.             .upgradeEffect(UpgradeEffects::mirrorEffect)
47.             .build(),
48.
49.         new UpgradeTileModel.UpgradeBuilder()
50.             .name("Friendship Bracelet")
51.             .description("Add +1 to the score multiplier for every consecutive letter in alphabetical order.")
52.             .imagePath("/com/example/project/upgradeTileImages/FriendshipBracelet.png")
53.             .cost(2)
54.             .upgradeEffect(UpgradeEffects::braceletEffect)
55.             .build(),
56.
57.         new UpgradeTileModel.UpgradeBuilder()
58.             .name("Lost Button")
59.             .description("Add +5 to the base word score for every X on the tile rack.")
60.             .imagePath("/com/example/project/upgradeTileImages/LostButton.png")
61.             .cost(2)
62.             .upgradeEffect(UpgradeEffects::buttonEffect)
63.             .build()
64.
65.     );
66.
67.     private static final Random random = new Random();
68.
69.     /**
70.      * Gets random upgrade tile.
71.      * @return returns upgrade tile.
72.     */
73.     public static UpgradeTileModel getRandomUpgradeTile() {
74.         var randomNum = random.nextInt(0, upgradeTiles.size());
75.         return upgradeTiles.get(randomNum);
76.     }
77.
78.     /**
79.      * Gets upgrade by name.
80.      * @param name name.
81.      * @return tile.
82.     */
83.     public static UpgradeTileModel getUpgradeByName(String name) {
84.         {
85.             for (UpgradeTileModel item : upgradeTiles)
86.             {
87.                 if (Objects.equals(item.getName(), name)){
88.                     return item;
89.                 }
90.             }
91.
92.             return null;
93.         }
94.     }
95.
96.

```

com.example.project.services.shopItems

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--------------------------------|---|------|--|------|--------|------|--------|-------|--------|---------|--------|---------|
| G UpgradeTiles |  84% | |  0% | | 4 | 6 | 6 | 46 | 2 | 4 | 0 | 1 |
| Total | 23 of 145 | 84% | 4 of 4 | 0% | 4 | 6 | 6 | 46 | 2 | 4 | 0 | 1 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.services.shopItems

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|-----------------------------------|---|------|--|------|--------|------|--------|-------|--------|---------|--------|---------|
| UpgradeTiles.java |  84% | 84% |  0% | 0% | 4 | 6 | 6 | 46 | 2 | 4 | 0 | 1 |
| Total | 23 of 145 | 84% | 4 of 4 | 0% | 4 | 6 | 6 | 46 | 2 | 4 | 0 | 1 |

Created with JaCoCo 0.8.12.202403310830

GameMusicPlayer

| Element | Missed Instructions Cov. | Missed Branches Cov. | Missed Cxtys | Missed Lines | Missed Methods |
|-------------------------------------|-----------------------------|-------------------------|-----------------|-----------------|-------------------|
| GameMusicPlayer() | 100% | n/a | 0 1 | 0 2 | 0 1 |
| playGameMusicLoop() | 100% | n/a | 0 1 | 0 2 | 0 1 |
| Total | 0 of 10 | 100% | 0 of 0 | n/a | 0 2 |

Created with JaCoCo 0.8.12.202403310830

GameMusicPlayer.java

```
1. package com.example.project.services.sound;
2.
3. import javax.sound.sampled.*;
4.
5. /**
6.  * Plays the game music track on repeat.
7. */
8. public class GameMusicPlayer extends GameSoundPlayer
9. {
10.     private static final String clipPath = "/com/example/project/Sounds/puzzleMusic_converted.wav";
11.
12.     /**
13.      * Create new instance of game music player.
14.      */
15.     public GameMusicPlayer()
16.     {
17.         super(clipPath, -3);
18.     }
19.
20.     /**
21.      * play game music loop continuously.
22.      */
23.     public void playGameMusicLoop()
24.     {
25.         clip.loop(Clip.LOOP_CONTINUOUSLY);
26.     }
27. }
```

GameSoundPlayer

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| convertFile(String) | | 100% | | 100% | 0 | 2 | 0 | 21 | 0 | 1 |
| GameSoundPlayer(String, Logger, float) | | 100% | | n/a | 0 | 1 | 0 | 8 | 0 | 1 |
| GameSoundPlayer(String, Logger) | | 100% | | n/a | 0 | 1 | 0 | 6 | 0 | 1 |
| GameSoundPlayer(String) | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| GameSoundPlayer(String, float) | | 100% | | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| changeVolume(float) | | 100% | | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| replay() | | 100% | | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| unMute() | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| mute() | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| Total | 0 of 174 | 100% | 0 of 2 | 100% | 0 | 10 | 0 | 50 | 0 | 9 |

Created with JaCoCo 0.8.12.202403310830

GameSoundPlayer.java

```
1. package com.example.project.services.sound;
2.
3. import com.example.project.services.Logger;
4.
5. import javax.sound.sampled.*;
6. import java.io.BufferedInputStream;
7. import java.io.IOException;
8. import java.io.InputStream;
9.
10.
11. /**
12.  * plays an audio clip as a sound effect for the game.
13. */
14. public class GameSoundPlayer
15. {
16.     private Logger logger = new Logger();
17.     protected final Clip clip;
18.     private float defaultGain = 0f;
19.
20.     /**
21.      * creates a new instance of GameSoundPlayer
22.      * @param filePath filepath to sound.
23.     */
24.     public GameSoundPlayer(String filePath)
25.     {
26.         clip = convertFile(filePath);
27.     }
28.
29.     protected GameSoundPlayer(String filePath, Logger logger)
30.     {
31.         this.logger = logger;
32.         clip = convertFile(filePath);
33.     }
34.
35.     protected GameSoundPlayer(String filePath, Logger logger, float gainAmount)
36.     {
37.         this.logger = logger;
38.         clip = convertFile(filePath);
39.         defaultGain = gainAmount;
40.         changeVolume(gainAmount);
41.     }
42.
43.     /**
44.      * creates a new instance of GameSoundPlayer
45.      * @param filePath filepath to sound.
46.      * @param gainAmount volume to set the clip at
47.     */
48.     public GameSoundPlayer(String filePath, float gainAmount)
49.     {
50.         this(filePath);
51.         defaultGain = gainAmount;
52.         changeVolume(gainAmount);
53.     }
54.
55.     /**
56.      * Reset clip and then play.
57.     */
58.     public void replay() {
59.         clip.setFramePosition(0);
60.         clip.start();
61.     }
62.
63.     /**
64.      * Change the current clips volume. -80f muted 0f is normal volume.
65.      * @param vol volume.
66.     */
67.     private void changeVolume(float vol)
68.     {
69.         FloatControl volumeControl = (FloatControl) clip.getControl(FloatControl.Type.MASTER_GAIN);
70.         volumeControl.setValue(vol);
71.     }
72.
73.     /**
74.      * set clip volume to mute.
75.     */
76.     public void mute()
77.     {
78.         changeVolume(-80f);
```

```

79. }
80.
81. /**
82. * Set clip to its default volume.
83. */
84. public void unMute(){
85.     changeVolume(defaultGain);
86. }
87.
88. /**
89. * File must be in a specific format for javax.sound.samples library. :(
90. * @param filePath file location.
91. * @return returns clip in correct format.
92. */
93. private Clip convertFile(String filePath) {
94.     Clip convertedClip;
95.
96.     InputStream resourceStream = GameMusicPlayer.class.getResourceAsStream(filePath);
97.     if (resourceStream == null)
98.     {
99.         var errorMessage = String.format("Sound file not found. No sound file at %s", filePath);
100.        this.logger.logError(errorMessage);
101.        throw new IllegalArgumentException(errorMessage);
102.    }
103.
104.    try {
105.        BufferedInputStream bufferedStream = new BufferedInputStream(resourceStream);
106.        AudioInputStream originalStream = AudioSystem.getAudioInputStream(bufferedStream);
107.
108.        AudioFormat targetFormat = new AudioFormat(
109.            AudioFormat.Encoding.PCM_SIGNED,
110.            44100.0f, 16, 2, 4, 44100.0f, false
111.        );
112.
113.        AudioInputStream convertedStream = AudioSystem.getAudioInputStream(targetFormat, originalStream);
114.
115.        convertedClip = AudioSystem.getClip();
116.        convertedClip.open(convertedStream);
117.
118.        convertedStream.close();
119.        originalStream.close();
120.        bufferedStream.close();
121.        resourceStream.close();
122.    }
123.    catch (UnsupportedAudioFileException | IOException | LineUnavailableException e)
124.    {
125.        var message = "Error loading sound file: " + e.getMessage();
126.        this.logger.logError(message);
127.        throw new IllegalArgumentException(message);
128.    }
129.
130.    return convertedClip;
131. }
132. }

```

com.example.project.controllers.gameScreens.animations

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C LevelScoreSequence | | 0% | | 0% | 3 | 3 | 19 | 19 | 2 | 2 | 1 | 1 |
| C TextEmphasisAnimation | | 0% | | n/a | 3 | 3 | 16 | 16 | 3 | 3 | 1 | 1 |
| C ConfettiAnimation | | 0% | | 0% | 3 | 3 | 15 | 15 | 2 | 2 | 1 | 1 |
| C ScoreTimeline | | 0% | | 0% | 5 | 5 | 17 | 17 | 4 | 4 | 1 | 1 |
| C InfiniteFloatingAnimation | | 0% | | n/a | 4 | 4 | 14 | 14 | 4 | 4 | 1 | 1 |
| C AnimationWrapper | | 0% | | n/a | 4 | 4 | 8 | 8 | 4 | 4 | 1 | 1 |
| Total | 416 of 416 | 0% | 6 of 6 | 0% | 22 | 22 | 89 | 89 | 19 | 19 | 6 | 6 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.services.sound

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|-----------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C GameSoundPlayer | 100% | 100% | 100% | n/a | 0 | 10 | 0 | 50 | 0 | 9 | 0 | 1 |
| C GameMusicPlayer | 100% | 100% | | n/a | 0 | 2 | 0 | 4 | 0 | 2 | 0 | 1 |
| Total | 0 of 184 | 100% | 0 of 2 | 100% | 0 | 12 | 0 | 54 | 0 | 11 | 0 | 2 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.services.sound

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| GameSoundPlayer.java | 100% | 100% | 100% | n/a | 0 | 10 | 0 | 50 | 0 | 9 | 0 | 1 |
| GameMusicPlayer.java | 100% | 100% | n/a | n/a | 0 | 2 | 0 | 4 | 0 | 2 | 0 | 1 |
| Total | 0 of 184 | 100% | 0 of 2 | 100% | 0 | 12 | 0 | 54 | 0 | 11 | 0 | 2 |

Created with JaCoCo 0.8.12.202403310830

DictionaryDAO

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|-----------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| getWordDefinition(String) | 100% | 100% | 100% | 100% | 0 | 3 | 0 | 20 | 0 | 1 |
| isWordInDictionary(String) | 100% | 100% | n/a | n/a | 0 | 1 | 0 | 9 | 0 | 1 |
| DictionaryDAO() | 100% | 100% | n/a | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| DictionaryDAO(Connection, Logger) | 100% | 100% | n/a | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| Total | 0 of 137 | 100% | 0 of 4 | 100% | 0 | 6 | 0 | 37 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

DictionaryDAO.java

```
1. package com.example.project.services.sqlite.dAOs;
2.
3. import com.example.project.services.Logger;
4. import com.example.project.services.SQLiteDictionaryConnection;
5.
6. import java.sql.*;
7. import java.util.ArrayList;
8. import java.util.List;
9.
10. /**
11. * The SQLite Dictionary. The connection returned from SQLiteDictionaryConnection().getInstance() is always the same.
12. */
13. public class DictionaryDAO
14. {
15.     private final Connection connection;
16.     private final Logger logger;
17.
18.     /**
19.     * Constructor for this class SQLLiteDictionary.
20.     */
21.     public DictionaryDAO()
22.     {
23.         this.connection = new SQLiteDictionaryConnection().getInstance();
24.         this.logger = new Logger();
25.     }
26.
27.     /**
28.     * constructor with injection for tests.
29.     * @param connection mock connection.
30.     * @param logger a mock logger.
31.     */
32.     public DictionaryDAO(Connection connection, Logger logger) {
33.         this.connection = connection;
34.         this.logger = logger;
35.     }
36.
37.     /**
38.     * gets the words definition.
39.     * @param wordToFind Word to get definition of.
40.     * @return Returns the definition.
41.     */
42.     public String getWordDefinition(String wordToFind)
43.     {
44.         List<String> definitions = new ArrayList<>();
45.         try
46.         {
47.             wordToFind = wordToFind.trim();
48.             PreparedStatement query = connection.prepareStatement("SELECT wordtype, definition FROM entries WHERE word = ? COLLATE NOCASE");
49.             query.setString(1, wordToFind);
50.             ResultSet result = query.executeQuery();
51.
52.             while (result.next())
53.             {
54.                 definitions.add("Wordtype: " +
55.                     result.getString("wordtype") +
56.                     System.lineSeparator() +
57.                     "Definition: " +
58.                     result.getString("definition"));
59.             }
60.             if (definitions.isEmpty())
61.             {
62.                 this.logger.LogError(String.format("No rows in database for word: %s", wordToFind));
63.                 definitions.add("No Definition");
64.             }
65.         }
66.         catch (SQLException e)
67.         {
68.             this.logger.LogError(String.format("Definition not found for word: %s", wordToFind));
69.             this.logger.LogError("Error message: " + e.getMessage());
70.             definitions.add("No Definition");
71.         }
72.         String doubleLineSeparator = System.lineSeparator() + System.lineSeparator();
73.         return String.join(doubleLineSeparator, definitions);
74.     }
75.
76.     /**
77.     * gets if word is in dictionary.
78.     * @param wordToCheck Word to check.
79.     * @return returns the boolean value indicating whether the word exists in our database.
80.     */
81.     public boolean isWordInDictionary(String wordToCheck)
82.     {
83.         wordToCheck = wordToCheck.toLowerCase();
84.
85.         try
86.         {
87.             PreparedStatement doesWordExistQuery = connection.prepareStatement("SELECT 1 FROM entries WHERE word = ? LIMIT 1");
88.             doesWordExistQuery.setString(1, wordToCheck);
89.             ResultSet doesWordExistResult = doesWordExistQuery.executeQuery();
90.             return doesWordExistResult.next();
```

```
91.     }
92.     catch (SQLException e)
93.     {
94.         this.logger.logError("word not found due to exception");
95.         this.logger.logError("Error message:" + e.getMessage());
96.     }
97.
98.     return false;
99. }
100.
```

UsersDAO

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| hasSaveData(User) | | 0% | | 0% | 5 | 5 | 17 | 17 | 1 | 1 |
| saveSessionData(String, String) | | 0% | | 0% | 2 | 2 | 15 | 15 | 1 | 1 |
| getSessionDataJson(String) | | 0% | | 0% | 2 | 2 | 14 | 14 | 1 | 1 |
| addSessionDataColumn() | | 0% | | 0% | 2 | 2 | 11 | 11 | 1 | 1 |
| doesUserExist(String) | | 87% | | 50% | 1 | 2 | 0 | 1 | 0 | 1 |
| addUser(User) | | 100% | | n/a | 0 | 1 | 0 | 12 | 0 | 1 |
| queryByUsername(String) | | 100% | | 100% | 0 | 2 | 0 | 13 | 0 | 1 |
| UsersDAO() | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| UsersDAO(Connection, Logger) | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| UsersDAO>PasswordHasher, Connection, Logger) | | 100% | | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| getUser(String) | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 291 of 475 | 38% | 15 of 18 | 16% | 12 | 20 | 57 | 96 | 4 | 11 |

Created with JaCoCo 0.8.12.202403310830

UsersDAO.java

```
1. package com.example.project.services.sqlite.dAOs;
2.
3. import com.example.project.services.Logger;
4. import com.example.project.services.PasswordHasher;
5. import com.example.project.models.User;
6. import com.example.project.services.sqlite.SQLiteUsersConnection;
7.
8. import java.sql.*;
9.
10. /**
11. * SQLite Users database. with a table `users` 2 columns. 'username', 'password'. Which are both defined as unique
12. * not null Strings in sqlite.
13. */
14. public class UsersDAO
15. {
16.     private final Logger logger;
17.     private final Connection connection;
18.     PasswordHasher passwordHasher = new PasswordHasher();
19.
20.     /**
21.      * Constructor for this class SQLiteDictionary.
22.      */
23.     public UsersDAO()
24.     {
25.         this.logger = new Logger();
26.         this.connection = new SQLiteUsersConnection().getInstance();
27.     }
28.
29.     protected UsersDAO(PasswordHasher hasher, Connection connection, Logger logger)
30.     {
31.         this(connection, logger);
32.         this.passwordHasher = hasher;
33.     }
34.
35.     /**
36.      * Constructor with injection for unit tests.
37.      * @param connection Connection
38.      * @param logger logger.
39.      */
40.     public UsersDAO(Connection connection, Logger logger)
41.     {
42.         this.logger = logger;
43.         this.connection = connection;
44.     }
45.
46.     /**
47.      * Adds user to the user.db.
48.      * password will be hashed before storing to ensure greater security (no plain text passwords)
49.      * @param user user to add.
50.      */
51.     public void addUser(User user)
52.     {
53.         String sql = "INSERT INTO users (username, password, highscore) VALUES (?, ?, ?)";
54.
55.         try
56.         {
57.             PreparedStatement query = this.connection.prepareStatement(sql);
58.             query.setString(1, user.getUsername());
59.
60.             query.setString(2, this.passwordHasher.hashPassword(user.getPassword())); //hashes password before storing
61.
62.             query.setInt(3, user.getHighscore());
63.             query.executeUpdate();
64.             this.logger.logMessage(String.format("added user: %s to the database", user.getUsername()));
65.         }
66.         catch (SQLException e)
67.         {
68.             this.logger.LogError(String.format("Failed to add user: %s to the database", user.getUsername()));
69.             throw new RuntimeException(String.format("Failed to add user to the database. SQL Error details: %s", e.getMessage()));
70.         }
71.     }
72.
73.     /**
74.      * @param username username
75.      * @return returns bool indicating whether use is in database already.
76.      */
77.     public boolean doesUserExist(String username)
78.     {
79.         return queryByUsername(username) != null;
80.     }
81.
82.     private User queryByUsername(String username)
83.     {
84.         String sql = "SELECT username, password, highscore FROM users WHERE username = ?";
```

```

85.
86.     try {
87.         PreparedStatement statement = connection.prepareStatement(sql);
88.         statement.setString(1, username);
89.         ResultSet result = statement.executeQuery();
90.
91.         if (!result.next()) { // if user does not exist in database return null.
92.             return null;
93.         }
94.
95.         int highscore = result.getInt("highscore");
96.         String usersPassword = result.getString("password");
97.         return new User(username, usersPassword, highscore);
98.     }
99.     catch (SQLException e) {
100.         {
101.             this.logger.logError(String.format("SQL Exception. Failed to get user username: %s", username));
102.             this.logger.logError(String.format("%s", e.getMessage()));
103.             throw new RuntimeException("Failed to get user by username", e);
104.         }
105.     }
106.
107. /**
108. * @param username username.
109. * @return returns user with matching username.
110. */
111. public User getUser(String username)
112. {
113.     return queryByUsername(username);
114. }
115.
116. /**
117. * Saves the session data as JSON for a specific user.
118. * @param username the username to save data for
119. * @param sessionJson the JSON string containing session data
120. */
121. public void saveSessionData(String username, String sessionJson)
122. {
123.     this.addSessionDataColumn();
124.     String sql = "UPDATE users SET session_data = ? WHERE username = ?";
125.
126.     try {
127.         PreparedStatement statement = connection.prepareStatement(sql);
128.         statement.setString(1, sessionJson);
129.         statement.setString(2, username);
130.
131.         int rowsAffected = statement.executeUpdate();
132.
133.         if (rowsAffected > 0) {
134.             logger.logMessage(String.format("Saved session data for user: %s", username));
135.         } else {
136.             logger.logError(String.format("No user found with username: %s", username));
137.         }
138.
139.     } catch (SQLException e) {
140.         logger.logError(String.format("Failed to save session data for user: %s", username));
141.         logger.logError(String.format("SQL Error: %s", e.getMessage()));
142.         throw new RuntimeException("Failed to save session data", e);
143.     }
144. }
145.
146. /**
147. * Get session saved data.
148. * @param username user.
149. * @return returns string.
150. */
151. public String getSessionDataJson(String username) {
152.     String sql = "SELECT session_data FROM users WHERE username = ?";
153.
154.     try {
155.         PreparedStatement statement = connection.prepareStatement(sql);
156.         statement.setString(1, username);
157.         ResultSet result = statement.executeQuery();
158.
159.         if (result.next()) {
160.             String sessionData = result.getString("session_data");
161.             logger.logMessage(String.format("Loaded session data for user: %s", username));
162.             return sessionData;
163.         } else {
164.             logger.logMessage(String.format("No session data found for user: %s", username));
165.             return null;
166.         }
167.
168.     } catch (SQLException e) {
169.         logger.logError(String.format("Failed to load session data for user: %s", username));
170.         logger.logError(String.format("SQL Error: %s", e.getMessage()));
171.         throw new RuntimeException("Failed to load session data", e);
172.     }
173. }
174.
```

```

175.     private void addSessionDataColumn() {
176.         String sql = "ALTER TABLE users ADD COLUMN session_data TEXT";
177.
178.         try {
179.             Statement statement = connection.createStatement();
180.             statement.executeUpdate(sql);
181.             logger.logMessage("Added session_data column to users table");
182.         } catch (SQLException e) {
183.             // Column might already exist
184.             if (e.getMessage().contains("duplicate column")) {
185.                 logger.logMessage("session_data column already exists");
186.             } else {
187.                 logger.logError("Error adding session_data column: " + e.getMessage());
188.                 throw new RuntimeException("Failed to add session_data column", e);
189.             }
190.         }
191.     }
192.
193. /**
194. * Does user have save data.
195. * @param user user.
196. * @return returns true if save data exists.
197. */
198. public boolean hasSaveData(User user)
199. {
200.     addSessionDataColumn();
201.
202.     String sql = "SELECT session_data FROM users WHERE username = ?";
203.
204.     try {
205.         PreparedStatement statement = connection.prepareStatement(sql);
206.         statement.setString(1, user.getUsername());
207.         ResultSet result = statement.executeQuery();
208.
209.         if (result.next()) {
210.             String sessionData = result.getString("session_data");
211.             // Check if session_data is not null and not empty
212.             boolean hasSave = sessionData != null && !sessionData.trim().isEmpty();
213.             logger.logMessage(String.format("User %s %s save data",
214.                 user.getUsername(), hasSave ? "has" : "does not have"));
215.             return hasSave;
216.         } else {
217.             logger.logError(String.format("User not found: %s", user.getUsername()));
218.             return false;
219.         }
220.
221.     } catch (SQLException e) {
222.         logger.logError(String.format("Failed to check save data for user: %s", user.getUsername()));
223.         logger.logError(String.format("SQL Error: %s", e.getMessage()));
224.         return false; // Return false instead of throwing to handle missing column gracefully
225.     }
226. }
227. }

```

com.example.project.services.sqlite.dAOs

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed Cxty | Missed Lines | Missed Methods | Missed Classes |
|---------------------------------|--|--|-----------------|------|-------------|--------------|----------------|----------------|
| C UsersDAO |  38% |  16% | 12 | 20 | 57 | 96 | 4 | 11 |
| C DictionaryDAO |  100% |  100% | 0 | 6 | 0 | 37 | 0 | 4 |
| Total | 291 of 612 | 52% | 15 of 22 | 31% | 12 | 26 | 57 | 133 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.services.sqlite.dAOs

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| UsersDAO.java | 38% | 16% | 12 | 20 | 57 | 96 | 4 | 11 | 0 | 1 | 0 | 1 |
| DictionaryDAO.java | 100% | 100% | 0 | 6 | 0 | 37 | 0 | 4 | 0 | 4 | 0 | 1 |
| Total | 291 of 612 | 52% | 15 of 22 | 31% | 12 | 26 | 57 | 133 | 4 | 15 | 0 | 2 |

Created with JaCoCo 0.8.12.202403310830

SQLiteConnection

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| getInstance() | | 57% | | 75% | 1 | 3 | 2 | 7 | 0 | 1 |
| createConnection() | | 36% | | n/a | 0 | 1 | 3 | 5 | 0 | 1 |
| static {...} | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| SQLiteConnection() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 28 of 65 | 56% | 1 of 4 | 75% | 1 | 6 | 5 | 14 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

SQLiteConnection.java

```
1. package com.example.project.services.sqlite;
2.
3. import com.example.project.services.Logger;
4.
5. import java.nio.file.Files;
6. import java.nio.file.Paths;
7. import java.sql.Connection;
8. import java.sql.DriverManager;
9. import java.sql.SQLException;
10.
11. /**
12. * Abstract SQLiteConnection class.
13. */
14. public abstract class SQLiteConnection {
15.     private static final Logger logger = new Logger();
16.
17.     protected abstract Connection getSQLiteInstance();
18.
19.     protected abstract void setSQLiteInstance(Connection newInstance);
20.
21.     /**
22.      * Each subclass provides its database path.
23.      */
24.     protected abstract String getDatabasePath();
25.
26.     /**
27.      * Gets the database connection from an existing readonly resource .db
28.      * @return returns the connection is already made or creates new one.
29.      */
30.     public Connection getInstance()
31.     {
32.         if (getSQLiteInstance() != null){
33.             return getSQLiteInstance();
34.         }
35.
36.         if (!Files.exists(Paths.get(getDatabasePath())))
37.         {
38.             logger.logError(String.format("Database connection failed: Database file not found: %s", getDatabasePath()));
39.             throw new RuntimeException("Cannot initialize connection");
40.         }
41.
42.         setSQLiteInstance(createConnection());
43.         return getSQLiteInstance();
44.     }
45.
46.     private Connection createConnection()
47.     {
48.         String dbUrl = "jdbc:sqlite:" + getDatabasePath();
49.         try
50.         {
51.             return DriverManager.getConnection(dbUrl);
52.         } catch (SQLException sqlEx)
53.         {
54.             logger.logError("Database connection failed: " + sqlEx.getMessage());
55.             throw new RuntimeException("Cannot initialize connection", sqlEx);
56.         }
57.     }
58. }
59. }
```

com.example.project.controllers.gameScreens.animations

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| LevelScoreSequence.java | | 0% | | 0% | 3 | 3 | 19 | 19 | 2 | 2 | 1 | 1 |
| TextEmphasisAnimation.java | | 0% | | n/a | 3 | 3 | 16 | 16 | 3 | 3 | 1 | 1 |
| ConfettiAnimation.java | | 0% | | 0% | 3 | 3 | 15 | 15 | 2 | 2 | 1 | 1 |
| ScoreTimeline.java | | 0% | | 0% | 5 | 5 | 17 | 17 | 4 | 4 | 1 | 1 |
| InfiniteFloatingAnimation.java | | 0% | | n/a | 4 | 4 | 14 | 14 | 4 | 4 | 1 | 1 |
| AnimationWrapper.java | | 0% | | n/a | 4 | 4 | 8 | 8 | 4 | 4 | 1 | 1 |
| Total | 416 of 416 | 0% | 6 of 6 | 0% | 22 | 22 | 89 | 89 | 19 | 19 | 6 | 6 |

SQLiteDictionaryConnection

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| SQLiteDictionaryConnection() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| setSQLiteInstance(Connection) | 100% | | n/a | | 0 | 1 | 0 | 2 | 0 | 1 |
| getSQLiteInstance() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| getDatabasePath() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 10 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 5 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

SQLiteDictionaryConnection.java

```
1. package com.example.project.services.sqlite;
2.
3. import java.sql.Connection;
4.
5. /**
6.  * Connection to the dictionary.db
7. */
8. public class SQLiteDictionaryConnection extends SQLiteConnection
9. {
10.     private static Connection instance;
11.
12.     @Override
13.     protected Connection getSQLiteInstance() {
14.         return instance;
15.     }
16.
17.     @Override
18.     protected void setSQLiteInstance(Connection newInstance) {
19.         instance = newInstance;
20.     }
21.
22.     @Override
23.     protected String getDatabasePath()
24.     {
25.         return "databases/English-Dictionary-Open-Source-main/sqlite3/dictionary.db";
26.     }
27. }
```

SQLiteUsersConnection

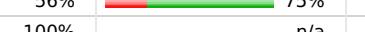
| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| SQLiteUsersConnection() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| setSQLiteInstance(Connection) | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| getSQLiteInstance() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getDatabasePath() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 10 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 5 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

SQLiteUsersConnection.java

```
1. package com.example.project.services.sqlite;
2.
3. import java.sql.Connection;
4.
5. /**
6. * Connection to the users.db
7. */
8. public class SQLiteUsersConnection extends SQLiteConnection
9. {
10.     private static Connection instance;
11.
12.     @Override
13.     protected Connection getSQLiteInstance() {
14.         return instance;
15.     }
16.
17.     @Override
18.     protected void setSQLiteInstance(Connection newInstance) {
19.         instance = newInstance;
20.     }
21.
22.     @Override
23.     protected String getDatabasePath()
24.     {
25.         return "databases/users.db";
26.     }
27. }
```

com.example.project.services.sqlite

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--|--|---|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C SQLiteConnection |  56% |  75% | 1 | 6 | 5 | 14 | 0 | 4 | 0 | 4 | 0 | 1 |
| C SQLiteUsersConnection |  100% | n/a | 0 | 4 | 0 | 5 | 0 | 4 | 0 | 4 | 0 | 1 |
| C SQLiteDictionaryConnection |  100% | n/a | 0 | 4 | 0 | 5 | 0 | 4 | 0 | 4 | 0 | 1 |
| Total | 28 of 85 | 67% | 1 of 4 | 75% | 1 | 14 | 5 | 24 | 0 | 12 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.services.sqlite

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| SQLiteConnection.java | | 56% | | 75% | 1 | 6 | 5 | 14 | 0 | 4 | 0 | 1 |
| SQLiteDictionaryConnection.java | | 100% | | n/a | 0 | 4 | 0 | 5 | 0 | 4 | 0 | 1 |
| SQLiteUsersConnection.java | | 100% | | n/a | 0 | 4 | 0 | 5 | 0 | 4 | 0 | 1 |
| Total | 28 of 85 | 67% | 1 of 4 | 75% | 1 | 14 | 5 | 24 | 0 | 12 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

FXMLPageLoader

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|----------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| load(String) | 100% | | n/a | | 0 | 1 | 0 | 2 | 0 | 1 |
| FXMLPageLoader() | 100% | | n/a | | 0 | 1 | 0 | 3 | 0 | 1 |
| getController() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 26 | 100% | 0 of 0 | n/a | 0 | 3 | 0 | 6 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

FXMLPageLoader.java

```
1. package com.example.project.services;
2.
3. import javafx.fxml.FXMLLoader;
4. import javafx.scene.Parent;
5.
6. import java.io.IOException;
7.
8. /**
9. * FXML Loader wrapper class for the project to load the game screen FXML pages.
10.*/
11. public class FXMLPageLoader implements PageLoader
12. {
13.     private FXMLLoader loader;
14.
15.     /**
16.      * constructor.
17.     */
18.     public FXMLPageLoader() {
19.         this.loader = new FXMLLoader();
20.     }
21.
22.     @Override
23.     public Parent load(String fxmlPath) throws IOException
24.     {
25.         this.loader = new FXMLLoader(this.getClass().getResource(fxmlPath));
26.         return loader.load();
27.     }
28.
29.     @Override
30.     public <T> T getController() {
31.         return loader.getController();
32.     }
33. }
```

GameScene

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| static {...} | 100% | | | n/a | 0 | 1 | 0 | 6 | 0 | 1 |
| Total | 0 of 33 | 100% | 0 of 0 | n/a | 0 | 1 | 0 | 6 | 0 | 1 |

Created with JaCoCo 0.8.12.202403310830

GameScene.java

```
1. package com.example.project.services;
2.
3. /**
4.  * Game Scenes enum.
5. */
6. public enum GameScene
7. {
8.     /**
9.      * Login Screen.
10.     */
11.    LOGIN,
12.
13.    /**
14.     * Main Menu Screen.
15.     */
16.    MAINMENU,
17.
18.    /**
19.     * Daily Reward Spinner Screen.
20.     */
21.    DAILY_REWARD,
22.
23.    /**
24.     * Level Screen
25.     */
26.    LEVEL,
27.
28.    /**
29.     * Shop screen.
30.     */
31.    SHOP,
32. }
```

DailyRewardController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| onSpinButtonClicked() | | 0% | | 0% | 2 | 2 | 11 | 11 | 1 | 1 |
| showReward(DailyRewardType) | | 0% | | 0% | 3 | 3 | 10 | 10 | 1 | 1 |
| onSceneChangedToThis() | | 0% | | 0% | 2 | 2 | 7 | 7 | 1 | 1 |
| setup(Session, SceneManager) | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| lambda\$onSpinButtonClicked\$0(ActionEvent) | | 0% | | n/a | 1 | 1 | 4 | 4 | 1 | 1 |
| DailyRewardController() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| lambda\$showReward\$1(ActionEvent) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 142 of 142 | 0% | 7 of 7 | 0% | 11 | 11 | 37 | 37 | 7 | 7 |

Created with JaCoCo 0.8.12.202403310830

Logger

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| Logger(ByteArrayOutputStream, ByteArrayOutputStream) | | 100% | n/a | 0 | 1 | 0 | 12 | 0 | 1 | |
| Logger() | | 100% | n/a | 0 | 1 | 0 | 9 | 0 | 1 | |
| logWithCapture(String, PrintStream, PrintStream) | | 100% | | 100% | 0 | 2 | 0 | 4 | 0 | 1 |
| Logger(ByteArrayOutputStream, ByteArrayOutputStream, PrintStream, PrintStream) | | 100% | n/a | 0 | 1 | 0 | 4 | 0 | 1 | |
| logMessage(String) | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| logError(String) | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| setPrintToConsole(boolean) | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| getErrorLogs() | | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 1 | |
| getLogs() | | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 1 | |
| clearLogs() | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| Total | 0 of 145 | 100% | 0 of 2 | 100% | 0 | 11 | 0 | 36 | 0 | 10 |

Created with JaCoCo 0.8.12.202403310830

Logger.java

```
1. package com.example.project.services;
2.
3. import java.io.ByteArrayOutputStream;
4. import java.io.PrintStream;
5.
6. /**
7.  * com.example.project.services.Logger for Application and to show up in test and build CI workflow.
8. */
9. public class Logger
10. {
11.     private PrintStream consoleErrorStream;
12.     private final PrintStream captureErrorStream;
13.     private ByteArrayOutputStream capturedErrorLogs = new ByteArrayOutputStream();
14.
15.     private PrintStream consoleStdOutStream;
16.     private final PrintStream captureStdOutStream;
17.     private ByteArrayOutputStream capturedStdOutLogs = new ByteArrayOutputStream();
18.
19.     private boolean printToConsole = true;
20.
21.     /**
22.      * Default constructor writes to console does not capture. For the project files. Use below for unit tests.
23.      */
24.     public Logger()
25.     {
26.         this.consoleErrorStream = System.err;
27.         this.captureErrorStream = new PrintStream(capturedErrorLogs);
28.
29.         this.consoleStdOutStream = System.out;
30.         this.captureStdOutStream = new PrintStream(capturedStdOutLogs);
31.     }
32.
33.     /**
34.      * 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.)
35.      * @param capturedErrStream byte array to store error logs.
36.      * @param capturedOutStream byte array to store standard output logs.
37.      */
38.     public Logger(ByteArrayOutputStream capturedErrStream, ByteArrayOutputStream capturedOutStream) {
39.         this.setPrintToConsole(false);
40.         this.consoleErrorStream = System.err;
41.         this.capturedErrorLogs = capturedErrStream;
42.         this.captureErrorStream = new PrintStream(this.capturedErrorLogs);
43.
44.         this.consoleStdOutStream = System.out;
45.         this.capturedStdOutLogs = capturedOutStream;
46.         this.captureStdOutStream = new PrintStream(this.capturedStdOutLogs);
47.     }
48.
49.     protected Logger(ByteArrayOutputStream capturedErrStream, ByteArrayOutputStream capturedOutStream, PrintStream mockRealErrOutput, PrintStream mockRealStdOutput)
50.     {
51.         this(capturedErrStream, capturedOutStream);
52.         this.consoleErrorStream = mockRealErrOutput;
53.         this.consoleStdOutStream = mockRealStdOutput;
54.     }
55.
56.     /**
57.      * @param value if this logger will also print to the console.
58.      */
59.     public void setPrintToConsole(boolean value){
60.         this.printToConsole = value;
61.     }
62.
63.     /**
64.      * @return Gets the error log messages.
65.      */
66.     public String getErrorLogs()
67.     {
68.         return this.capturedErrorLogs.toString();
69.     }
70.
71.
72.     /**
73.      * @return returns the standard log messages.
74.      */
75.     public String getLogs(){ return this.capturedStdOutLogs.toString(); }
76.
77.     /**
78.      * Logs an error message to standard error. And adds a newline.
79.      * <p>
80.      * The message can include format specifiers like in {@link String#format(String, Object...)}.
81.      *
82.      * @param message the error message format string (e.g., "Failed to connect to %s")
83.      */
84.     public void logError(String message)
85.     {
86.         this.logWithCapture(message, captureErrorStream, consoleErrorStream);
87.     }
88.
89.     /**
90.      * Clears captured logs use in tests teardown.
91.      */
92.     public void clearLogs() {
93.         capturedErrorLogs.reset();
94.     }
95.
96.     /**
97.      * Log message to System.out.
98.      * @param message message.
99.      */
100.    public void logMessage(String message)
101.    {
102.        this.logWithCapture(String.format(message), captureStdOutStream, consoleStdOutStream);
103.    }
104.
105.    private synchronized void logWithCapture(String formattedMessage, PrintStream captureStream, PrintStream consoleStream)
106.    {
```

```
107.     captureStream.printf(formattedMessage + "%n"); // capture in memory for tests to look at.
108.     ◆ if (printToConsole)
109.     {
110.         consoleStream.printf(formattedMessage + "%n"); // console output
111.     }
112. }
113. }
```

PasswordHasher

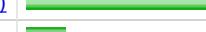
| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| doesPasswordMatch(String, String) | 100% | | n/a | | 0 | 1 | 0 | 5 | 0 | 1 |
| PasswordHasher() | 100% | | n/a | | 0 | 1 | 0 | 3 | 0 | 1 |
| PasswordHasher(Logger) | 100% | | n/a | | 0 | 1 | 0 | 3 | 0 | 1 |
| hashPassword(String) | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 49 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 12 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

PasswordHasher.java

```
1. package com.example.project.services;
2. import org.mindrot.jbcrypt.BCrypt;
3.
4.
5. /**
6.  * Hashes passwords.
7. */
8. public class PasswordHasher
9. {
10.     private final Logger logger;
11.
12.     /**
13.      * public constructor.
14.     */
15.     public PasswordHasher(){
16.         this.logger = new Logger();
17.     }
18.
19.     protected PasswordHasher(Logger logger)
20.     {
21.         this.logger = logger;
22.     }
23.
24.     /**
25.      * Hashes a raw password using BCrypt.
26.      *
27.      * @param rawPassword the plain text password
28.      * @return the hashed password string
29.      */
30.     public String hashPassword(String rawPassword) {
31.         return BCrypt.hashpw(rawPassword, BCrypt.gensalt());
32.     }
33.
34.     /**
35.      * Verifies a candidate password against a hashed password.
36.      * @param candidate the plain text password entered by the user
37.      * @param hashed    the hashed password stored in the database
38.      * @return true if the password matches, false otherwise
39.      */
40.     public boolean doesPasswordMatch(String candidate, String hashed)
41.     {
42.         try {
43.             return BCrypt.checkpw(candidate, hashed);
44.         }
45.         catch (IllegalArgumentException e)
46.         {
47.             this.logger.logMessage(String.format("Password check failed, caught exception: %s", e.getMessage()));
48.             this.logger.logMessage(String.format("Cause: %s", e.getCause()));
49.             return false;
50.         }
51.     }
52. }
```

PopupLoader

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|-----------------------------------|---|------|-----------------|------|--------|------|--------|-------|--------|---------|
| PopupLoader() |  | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| createPopupController(PopupModel) |  | 100% | | n/a | 0 | 1 | 0 | 7 | 0 | 1 |
| PopupLoader(FXMLLoaderPageLoader) |  | 100% | | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| Total | 8 of 41 | 80% | 0 of 0 | n/a | 1 | 3 | 3 | 13 | 1 | 3 |

Created with JaCoCo 0.8.12.202403310830

PopupLoader.java

```
1. package com.example.project.services;
2.
3. import com.example.project.controllers.popupControllers.PopupController;
4. import com.example.project.models.popups.PopupModel;
5. import javafx.scene.Parent;
6.
7. import java.io.IOException;
8.
9. /**
10. * class to load the popup and initialize it with a controller
11. */
12. public class PopupLoader
13. {
14.     private final FXMLPageLoader loader;
15.
16.     protected PopupLoader(FXMLPageLoader loader)
17.     {
18.         this.loader = loader;
19.     }
20.
21.     /**
22.      * Constructor.
23.     */
24.     public PopupLoader()
25.     {
26.         this.loader = new FXMLPageLoader();
27.     }
28.
29.     /**
30.      * create popup controller.
31.      * @param popupModel popup object.
32.      * @param <C> class of popup controller.
33.      * @param <P> class of popup.
34.      * @return returns controller of the tile.
35.     */
36.     public <C extends PopupController<P>, P extends PopupModel> C createPopupController(P popupModel)
37.     {
38.         try
39.         {
40.             // In this case not using javafx Popup because of its limitations.
41.             // Might refactor in case we use actual Popups.
42.             String fxmlPath = popupModel.getFXMLPath();
43.             Parent root = this.loader.load(fxmlPath);
44.             C controller = loader.getController();
45.             controller.initialize(popupModel);
46.             return controller;
47.         } catch (IOException e) {
48.             throw new RuntimeException("Failed to create popup controller: " + popupModel.getFXMLPath(), e);
49.         }
50.     }
51. }
```

SVGIcons

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| getCogIcon() | 0% | | n/a | | 1 | 1 | 4 | 4 | 1 | 1 |
| SVGIcons() | 0% | | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 15 of 15 | 0% | 0 of 0 | n/a | 2 | 2 | 5 | 5 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

SVGIcons.java

```
1. package com.example.project.services;
2.
3. import javafx.scene.paint.Color;
4. import javafx.scene.shape.SVGPath;
5.
6. /**
7. * a static class to hold some icons for the application.
8. */
9. public class SVGIcons
10. {
11.     private static final String cogIcon = "M19.43 12.98c.04-.32.07-.66.07-1s-.03-.68-.07-1l2.11-1.65c.19-.15.24-.42.12-.64l-2-3.46c-.12-.21-.37-.3-.6-.22l-2.49 1a7.027 7.027 0 00-1.73-1l-.38-2.65A.4";
12.
13.     /**
14.      * gets a cog icon.
15.      * @return icon SVG.
16.     */
17.     public static SVGPath getCogIcon()
18.     {
19.         SVGPath cog = new SVGPath();
20.         cog.setContent(cogIcon);
21.         cog.setFill(Color.GRAY);
22.         return cog;
23.     }
24. }
```

SceneManager

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| getScene() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| SceneManager(RootLayoutController, Map, Map) | | 100% | | n/a | 0 | 1 | 0 | 7 | 0 | 1 |
| switchScene(GameScene) | | 100% | | n/a | 0 | 1 | 0 | 6 | 0 | 1 |
| Total | 3 of 47 | 93% | 0 of 0 | n/a | 1 | 3 | 1 | 14 | 1 | 3 |

Created with JaCoCo 0.8.12.202403310830

SceneManager.java

```
1. package com.example.project.services;
2.
3. import com.example.project.controllers.RootLayoutController;
4. import com.example.project.services.sound.GameSoundPlayer;
5. import javafx.scene.Parent;
6.
7. import java.util.HashMap;
8. import java.util.Map;
9. import com.example.project.controllers.gameScreens.GameScreenController;
10.
11. /**
12.  * Represents the scene manager class.
13. */
14. public class SceneManager
15. {
16.     private final RootLayoutController rootController;
17.     private Map<GameScene, Parent> pages = new HashMap<>();
18.     private Map<GameScene, GameScreenController> controllers = new HashMap<>();
19.     private GameScene currentScene;
20.
21. /**
22.  * Constructor with injection for tests
23.  * @param newRootController root controller
24.  * @param controllersToUse controllers
25.  * @param pagesToUse pages
26. */
27. public SceneManager(RootLayoutController newRootController, Map<GameScene,
28.                     GameScreenController> controllersToUse, Map<GameScene, Parent> pagesToUse)
29. {
30.     rootController = newRootController;
31.     controllers = controllersToUse;
32.     pages = pagesToUse;
33. }
34.
35. /**
36.  * Switch between scenes specify scene type of type from {@link GameScene}.
37.  * @param type scene.
38. */
39. public void switchScene(GameScene type)
40. {
41.     currentScene = type;
42.     Parent page = pages.get(type);
43.     rootController.setContent(page);
44.     GameScreenController controller = controllers.get(type);
45.     controller.onSceneChangedToThis();
46. }
47.
48. /**
49.  * get current scene.
50.  * @return scene.
51. */
52. public GameScene getCurrentScene()
53. {
54.     return currentScene;
55. }
56. }
```

DailyRewardController.java

```
1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.models.gameScreens.DailyRewardModel;
4. import com.example.project.models.gameScreens.DailyRewardType;
5. import com.example.project.services.GameScene;
6. import com.example.project.services.SceneManager;
7. import com.example.project.services.Session;
8. import javafx.animation.PauseTransition;
9. import javafx.animation.RotateTransition;
10. import javafx.fxml.FXML;
11. import javafx.scene.control.Button;
12. import javafx.scene.control.Label;
13. import javafx.scene.image.ImageView;
14. import javafx.scene.layout.Pane;
15. import javafx.scene.layout.StackPane;
16. import javafx.util.Duration;
17.
18. import java.time.LocalDate;
19. import java.util.Random;
20.
21. /**
22. * Controller for the daily reward popup screen.
23. */
24. public class DailyRewardController extends GameScreenController
25. {
26.     @FXML private StackPane root;
27.     @FXML private ImageView wheelImage;
28.     @FXML private ImageView stickImage;
29.     @FXML private Button spinButton;
30.     @FXML private Label rewardResultLabel;
31.     @FXML private Pane confettiLayer;
32.
33.     private final Random random = new Random();
34.     private DailyRewardModel model;
35.
36.     /**
37.      * Initialises the daily reward view by resetting the reward label.
38.      */
39.     @Override
40.     public void setup(Session session, SceneManager sceneManager)
41.     {
42.         model = new DailyRewardModel(session, sceneManager);
43.         rewardResultLabel.setText("");
44.     }
45.
46.     @Override
47.     public void onSceneChangedToThis() {
48.         rewardResultLabel.setText("");
49.         confettiLayer.getChildren().clear();
50.
51.         if (model.getSession().hasClaimedRewardToday()) {
52.             spinButton.setDisable(true);
53.             rewardResultLabel.setText("You've already claimed today's reward.");
54.         } else {
55.             spinButton.setDisable(false);
56.         }
57.     }
58.
59.     @FXML
60.     private void onSpinButtonClicked() {
61.         // Block spin if already claimed
62.         if (model.getSession().hasClaimedRewardToday())
63.         {
64.             rewardResultLabel.setText("Already claimed.");
65.             spinButton.setDisable(true);
66.             return;
67.         }
68.
69.         spinButton.setDisable(true);
70.
71.         int spinDegrees = 720 + random.nextInt(360); // random stop angle
72.         RotateTransition rotate = new RotateTransition(Duration.seconds(2), stickImage);
73.         rotate.setByAngle(spinDegrees);
74.         rotate.setOnFinished(event ->
75.         {
76.             DailyRewardType reward = model.rollReward();
77.             model.applyReward(reward);
78.             showReward(reward);
```

```
79.     });
80.     rotate.play();
81. }
82.
83. private void showReward(DailyRewardType reward)
84. {
85.     var timeDelay = new PauseTransition(Duration.millis(3000));
86.
87.     String message = switch (reward) {
88.         case Daily_Reward_Won_1Dollar -> "You won $1!";
89.         case Daily_Reward_Won_5Dollars -> "You won $5!";
90.         case Daily_Reward_Won_NOTHING -> "You won... nothing! Try again tomorrow for a better Prize!";
91.     };
92.
93.     rewardResultLabel.setText(message);
94.     model.getSession().setLastRewardDate(LocalDate.now());
95.     timeDelay.setOnFinished(e -> model.getSceneManager().switchScene(GameScene.MAINMENU));
96.     timeDelay.play();
97. }
98. }
```

Created with JaCoCo 0.8.12.202403310830

Session

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| load() | | 0% | | 0% | 7 | 7 | 25 | 25 | 1 | 1 |
| Save() | | 0% | | 0% | 3 | 3 | 15 | 15 | 1 | 1 |
| Session() | | 0% | n/a | n/a | 1 | 1 | 18 | 18 | 1 | 1 |
| modifyMoney(double) | | 0% | n/a | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| addUpgrade(UpgradeTileModel) | | 0% | n/a | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| hasClaimedRewardToday() | | 0% | n/a | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| resetMoney() | | 0% | n/a | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| setUser(User) | | 0% | n/a | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| getHandSize() | | 0% | n/a | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getWordWindowSize() | | 0% | n/a | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| setLastRewardDate(LocalDate) | | 0% | n/a | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| getUser() | | 0% | n/a | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getRedrawWindowSize() | | 0% | n/a | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Session(int, int, int, ObservableList, User, int, int, int, int, int) | | 100% | n/a | n/a | 0 | 1 | 0 | 24 | 0 | 1 |
| updateLevelInfo() | | 100% | n/a | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| resetGame() | | 100% | n/a | n/a | 0 | 1 | 0 | 6 | 0 | 1 |
| resetPlaysRedraws() | | 100% | n/a | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| getPlayersUpgradesProperty() | | 100% | n/a | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getLevelRequirement() | | 100% | n/a | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getLevelsBeaten() | | 100% | n/a | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getMoneyProperty() | | 100% | n/a | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getCurrentPlays() | | 100% | n/a | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getCurrentRedraws() | | 100% | n/a | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 318 of 488 | 34% | 16 of 16 | 0% | 21 | 31 | 61 | 104 | 13 | 23 |

Created with JaCoCo 0.8.12.202403310830

Session.java

```

1. package com.example.project.services;
2.
3. import com.example.project.models.User;
4. import com.example.project.models.tiles.UpgradeTileModel;
5. import com.example.project.services.shopItems.UpgradeTiles;
6. import com.example.project.services.sqlite.dAOs.UsersDAO;
7. import com.google.gson.Gson;
8. import com.google.gson.GsonBuilder;
9. import javafx.beans.property.*;
10. import javafx.collections.FXCollections;
11. import javafx.collections.ObservableList;
12. import java.time.LocalDate;
13.
14.
15. /**
16. * Game Session. holds info of the current session.
17. */
18. public class Session
19. {
20.     private Integer handSize = 9;
21.
22.     private Integer wordWindowSize = 9;
23.
24.     private Integer redrawWindowSize = 9;
25.
26.     private final ObservableList<UpgradeTileModel> upgrades = FXCollections.observableArrayList();
27.
28.     private final DoubleProperty money;
29.
30.     private final int initialMoney;
31.
32.     private User loggedInUser;
33.
34.     private int levelsBeaten = 0;
35.
36.     private final int initialLevelRequirement;
37.
38.     private final Logger logger = new Logger();
39.
40.     // initial plays and redraws are used to reset the redraws / plays at start of level. As upgrade effects can change how many plays/redraws you start with.
41.     private int initialRedraws = 4;
42.     private final ReadOnlyIntegerWrapper currentRedraws = new ReadOnlyIntegerWrapper(initialRedraws);
43.     private int initialPlays = 4;
44.     private final ReadOnlyIntegerWrapper currentPlays = new ReadOnlyIntegerWrapper(initialPlays);
45.     private final UsersDAO usersDB = new UsersDAO();
46.
47. /**
48. * points required for the player to score at least to beat the current level.
49. */
50. private final ReadOnlyIntegerWrapper levelRequirement;
51.
52. /**
53. * @return points required for the play to score at least to beat the level.
54. */
55. public ReadOnlyIntegerWrapper getLevelRequirement() {
56.     return levelRequirement;
57. }
58.
59. /**
60. * Constructor for injecting values in for unit test.
61. * @param newHandSize hand size.
62. * @param newWordViewSize word length allowed.
63. * @param newRedrawWindowSize redraw window size.
64. * @param newUpgrades upgrade tiles.
65. * @param newUser user.
66. * @param newMoney money.
67. * @param newLevelsBeaten levels beaten.
68. * @param currentLevelRequirement current level requirement.
69. * @param newFirstLevelsRequirement first level requirement.
70. */
71. protected Session(int newHandSize, int newWordViewSize, int newRedrawWindowSize,
72.                 ObservableList<UpgradeTileModel> newUpgrades, User newUser,
73.                 int newMoney, int newLevelsBeaten,
74.                 int currentLevelRequirement, int newFirstLevelsRequirement, int newInitialMoney)
75. {
76.     initialMoney = newInitialMoney;
77.     money = new ReadOnlyDoubleWrapper(newMoney);
78.     initialLevelRequirement = newFirstLevelsRequirement;
79.     handSize = newHandSize;
80.     wordWindowSize = newWordViewSize;
81.     redrawWindowSize = newRedrawWindowSize;
82.     loggedInUser = newUser;
83.     upgrades.setAll(newUpgrades);
84.     levelsBeaten = newLevelsBeaten;
85.     levelRequirement = new ReadOnlyIntegerWrapper(currentLevelRequirement);
86. }
87.
88. protected int getLevelsBeaten(){
89.     return levelsBeaten;
90. }
91.
92. /**
93. * Default constructor.
94. */
95. public Session()
96. {
97.     initialLevelRequirement = 4;
98.     levelRequirement = new ReadOnlyIntegerWrapper(initialLevelRequirement);
99.     initialMoney = 0;
100.    money = new ReadOnlyDoubleWrapper(initialMoney);
101. }
102.
```

```

103.
104.    /**
105.     * Returns the read-only money property for binding to UI components.
106.     * This allows UI elements to automatically update when the players money changes.
107.     * @return ReadOnlyIntegerProperty representing the player's current money amount
108.    */
109.   public DoubleProperty getMoneyProperty()
110.   {
111.     return money;
112.   }
113.
114.   /**
115.    * Adds an upgrade tile to the player's collection.
116.    * This will automatically update all UI displays bound to the upgrades property.
117.    *
118.    * @param upgrade this upgrades the tile to add it to the players collection
119.    */
120.   public void addUpgrade(UpgradeTileModel upgrade)
121.   {
122.     upgrades.add(upgrade);
123.   }
124.
125.   /**
126.    * Checks if the player already claimed today's reward.
127.    * @return true if already claimed today
128.    */
129.   public boolean hasClaimedRewardToday() {
130.     return LocalDate.now().equals(lastRewardDate);
131.   }
132.
133.   /**
134.    * set new user.
135.    * @param newUser user that logged in.
136.    */
137.   public void setUser(User newUser) {
138.     loggedInUser = newUser;
139.   }
140.
141.   /**
142.    * Returns logged in user.
143.    * @return user.
144.    */
145.   public User getUser(){
146.     return loggedInUser;
147.   }
148.
149.   /**
150.    * gets hand size.
151.    * @return returns number of tiles allowed in hand.
152.    */
153.   public int getHandSize() {
154.     return handSize;
155.   }
156.
157.   /**
158.    * gets word size.
159.    * @return return int word size.
160.    */
161.   public int getWordWindowSize() {
162.     return wordWindowSize;
163.   }
164.
165.   /**
166.    * gets upgrade tile property
167.    * @return upgrade tiles model list
168.    */
169.   public ReadOnlyListProperty<UpgradeTileModel> getPlayersUpgradesProperty() {
170.     return new ReadOnlyListWrapper<>(upgrades);
171.   }
172.
173.   /**
174.    * Increments how many points are required to beat the level.
175.    */
176.   public void updateLevelInfo() {
177.     this.levelsBeaten++;
178.     var current = this.levelRequirement.get();
179.     this.levelRequirement.set(current + (int) Math.pow(2, this.levelsBeaten));
180.   }
181.
182.   /**
183.    * Resets the current session when you lose
184.    */
185.   public void resetGame() {
186.     money.set(initialMoney); // resets the players account to their starting amount
187.     levelsBeaten = 0;
188.     levelRequirement.set(initialLevelRequirement);
189.     upgrades.clear();
190.     resetPlaysRedraws();
191.   }
192.
193.   /**
194.    * gets redraw window size (number of slots)
195.    * @return return int redraw window size.
196.    */
197.   public Integer getRedrawWindowSize() {
198.     return redrawWindowSize;
199.   }
200.
201.   /**
202.    * gets the current plays.
203.    * @return current plays remaining.
204.    */
205.   public ReadOnlyIntegerWrapper getCurrentPlays() {
206.     return currentPlays;
207.   }
208.
209.   /**
210.    * gets the redraws property.
211.    * @return the current redraws.
212.   */

```

```

211. /*
212.  * @return ReadOnlyIntegerWrapper getCurrentRedraws()
213.  */
214. 
215. /**
216.  * Reset the plays and redraws.
217. */
218. 
219. public void resetPlaysRedraws()
220. {
221.     getCurrentRedraws().set(initialRedraws);
222.     getCurrentPlays().set(initialPlays);
223. }
224. 
225. private LocalDate lastRewardDate = null;
226. 
227. /**
228.  * Adds or remove money to the player's balance.
229.  * @param amount amount to add
230. */
231. public void modifyMoney(double amount) {
232.     this.money.set(this.money.get() + amount);
233. }
234. 
235. /**
236.  * Sets the date the daily reward was last claimed.
237.  * @param date LocalDate of the reward claim
238. */
239. public void setLastRewardDate(LocalDate date) {
240.     this.lastRewardDate = date;
241. }
242. 
243. /**
244.  * Resets the player's money to the initial state (e.g. 0).
245. */
246. public void resetMoney() {
247.     this.money.set(0);
248. }
249. 
250. /**
251.  * will save a copy of this session data to local drive.
252. */
253. public void Save()
254. {
255.     SessionData data = new SessionData();
256.     data.money = this.money.get();
257.     data.levelsBeaten = this.levelsBeaten;
258.     data.levelRequirement = this.levelRequirement.get();
259.     data.currentInitialPlays = this.initialPlays;
260.     data.currentInitialRedraws = this.initialRedraws;
261.     data.lastRewardDate = this.lastRewardDate != null ? this.lastRewardDate.toString() : null;
262.     data.username = this.loggedInUser.getUsername();
263. 
264.     for (UpgradeTileModel tile : this.upgrades)
265.     {
266.         data.upgradeNames.add(tile.getName());
267.     }
268. 
269.     Gson gson = new GsonBuilder().setPrettyPrinting().create();
270.     String json = gson.toJson(data);
271.     this.usersDB.saveSessionData(loggedInUser.getUsername(), json);
272. }
273. 
274. /**
275.  * Load logged in users data.
276. */
277. 
278. public void load()
279. {
280.     try {
281.         // Get session data from database
282.         String json = this.usersDB.getSessionDataJson(this.loggedInUser.getUsername());
283. 
284.         // No saved data exists
285.         if (json == null || json.isEmpty()) {
286.             this.logger.logError("No save data found for user: " + this.loggedInUser.getUsername());
287.             return;
288.         }
289. 
290.         // Parse JSON
291.         Gson gson = new Gson();
292.         SessionData data = gson.fromJson(json, SessionData.class);
293. 
294.         // Restore session state
295.         this.money.set(data.money);
296.         this.levelsBeaten = data.levelsBeaten;
297.         this.levelRequirement.set(data.levelRequirement);
298.         this.initialPlays = data.currentInitialPlays;
299.         this.initialRedraws = data.currentInitialRedraws;
300.         this.currentPlays.set(data.currentInitialPlays);
301.         this.currentRedraws.set(data.currentInitialRedraws);
302.         this.lastRewardDate = data.lastRewardDate != null ? LocalDate.parse(data.lastRewardDate) : null;
303. 
304.         // Restore upgrades
305.         this.upgrades.clear();
306.         if (data.upgradeNames != null) {
307.             for (String name : data.upgradeNames)
308.             {
309.                 UpgradeTileModel upgrade = UpgradeTiles.getUpgradeByName(name);
310.                 if (upgrade != null) this.upgrades.add(upgrade);
311.             }
312. 
313.             this.logger.logMessage(String.format("Successfully loaded session for user: " + this.loggedInUser.getUsername()));
314. 
315.         } catch (Exception e)
316.         {
317.             this.logger.logError("Failed to load session data: " + e.getMessage());
318.         }

```


SessionData

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxts | Missed | Lines | Missed | Methods |
|---------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| SessionData() | 0% | | n/a | | 1 | 1 | 2 | 2 | 1 | 1 |
| Total | 8 of 8 | 0% | 0 of 0 | n/a | 1 | 1 | 2 | 2 | 1 | 1 |

Created with JaCoCo 0.8.12.202403310830

SessionData.java

```
1. package com.example.project.services;
2.
3. import java.util.ArrayList;
4. import java.util.List;
5.
6. /**
7. * Session data.
8. */
9. public class SessionData
10. {
11.     /**
12.      * current money.
13.      */
14.     public double money;
15.
16.     /**
17.      * levels beaten.
18.      */
19.     public int levelsBeaten;
20.
21.     /**
22.      * level requirement.
23.      */
24.     public int levelRequirement;
25.
26.     /**
27.      * current max plays.
28.      */
29.     public int currentInitialPlays;
30.
31.     /**
32.      * current max redraws.
33.      */
34.     public int currentInitialRedraws;
35.
36.     /**
37.      * last reward date.
38.      */
39.     public String lastRewardDate;
40.
41.     /**
42.      * username.
43.      */
44.     public String username;
45.
46.     /**
47.      * upgrade names.
48.      */
49.     public List<String> upgradeNames = new ArrayList<>();
50. }
```

com.example.project.services

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| Session | | 34% | | 0% | 21 | 31 | 61 | 104 | 13 | 23 | 0 | 1 |
| SVGIcons | | 0% | | n/a | 2 | 2 | 5 | 5 | 2 | 2 | 1 | 1 |
| PopupLoader | | 80% | | n/a | 1 | 3 | 3 | 13 | 1 | 3 | 0 | 1 |
| SessionData | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 |
| SceneManager | | 93% | | n/a | 1 | 3 | 1 | 14 | 1 | 3 | 0 | 1 |
| Logger | | 100% | | 100% | 0 | 11 | 0 | 36 | 0 | 10 | 0 | 1 |
| PasswordHasher | | 100% | | n/a | 0 | 4 | 0 | 12 | 0 | 4 | 0 | 1 |
| GameScene | | 100% | | n/a | 0 | 1 | 0 | 6 | 0 | 1 | 0 | 1 |
| FXMLPageLoader | | 100% | | n/a | 0 | 3 | 0 | 6 | 0 | 3 | 0 | 1 |
| Total | 352 of 852 | 58% | 16 of 18 | 11% | 26 | 59 | 72 | 198 | 18 | 50 | 2 | 9 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.services

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|-------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| Session.java | 34% | 34% | 0% | 0% | 21 | 31 | 61 | 104 | 13 | 23 | 0 | 1 |
| SVGIcons.java | 0% | 0% | n/a | n/a | 2 | 2 | 5 | 5 | 2 | 2 | 1 | 1 |
| PopupLoader.java | 80% | 80% | n/a | n/a | 1 | 3 | 3 | 13 | 1 | 3 | 0 | 1 |
| SessionData.java | 0% | 0% | n/a | n/a | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 |
| SceneManager.java | 93% | 93% | n/a | n/a | 1 | 3 | 1 | 14 | 1 | 3 | 0 | 1 |
| Logger.java | 100% | 100% | 100% | 100% | 0 | 11 | 0 | 36 | 0 | 10 | 0 | 1 |
| PasswordHasher.java | 100% | 100% | n/a | n/a | 0 | 4 | 0 | 12 | 0 | 4 | 0 | 1 |
| GameScene.java | 100% | 100% | n/a | n/a | 0 | 1 | 0 | 6 | 0 | 1 | 0 | 1 |
| FXMLPageLoader.java | 100% | 100% | n/a | n/a | 0 | 3 | 0 | 6 | 0 | 3 | 0 | 1 |
| Total | 352 of 852 | 58% | 16 of 18 | 11% | 26 | 59 | 72 | 198 | 18 | 50 | 2 | 9 |

Created with JaCoCo 0.8.12.202403310830

Application

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| start(Stage) | | 0% | | n/a | 1 | 1 | 14 | 14 | 1 | 1 |
| main(String[]) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| Application() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| static {...} | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 65 of 65 | 0% | 0 of 0 | n/a | 4 | 4 | 18 | 18 | 4 | 4 |

Created with JaCoCo 0.8.12.202403310830

Application.java

```
1. package com.example.project;
2.
3. import com.example.project.controllers.RootLayoutController;
4. import com.example.project.controllers.gameScreens.GameScreenFactory;
5. import com.example.project.services.Session;
6. import javafx.fxml.FXMLLoader;
7. import javafx.scene.Parent;
8. import javafx.scene.Scene;
9. import javafx.stage.Stage;
10. import com.example.project.services.FXMLPageLoader;
11.
12. import java.io.IOException;
13.
14. /**
15. * The entry point of the application.
16. */
17. public class Application extends javafx.application.Application
18. {
19.     // Constants defining the window title and size
20.     public static final String TITLE = "WordPlay";
21.
22.     // defining the window width.
23.     public static final int WIDTH = 1300;
24.
25.     // Defining the window height.
26.     public static final int HEIGHT = 700;
27.
28.     /**
29.      * dev buttons should be on or not.
30.     */
31.     public static boolean DEV_BUTTONS_ON = false;
32.
33.     @Override
34.     public void start(Stage stage) throws IOException
35.     {
36.         String rootScenePath = "/com/example/project/rootLayout-view.fxml";
37.         FXMLLoader rootLoader = new FXMLLoader(this.getClass().getResource(rootScenePath));
38.         Parent root = rootLoader.load();
39.         Scene scene = new Scene(root, WIDTH, HEIGHT);
40.         stage.setTitle(TITLE);
41.         stage.setScene(scene);
42.         stage.show();
43.         stage.setResizable(true);
44.         stage.setFullScreen(true);
45.
46.         Session session = new Session();
47.         GameScreenFactory factory = new GameScreenFactory(session);
48.         RootLayoutController rootController = rootLoader.getController();
49.         factory.loadGameScreens(rootController, new FXMLPageLoader());
50.     }
51.
52.     /**
53.      * @param args the command-line arguments passed to your program when it starts.
54.     */
55.     public static void main(String[] args) {
56.         launch();
57.     }
58. }
```

com.example.project

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---------------|---|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| G Application | <div style="width: 0%; background-color: red;"></div> | 0% | n/a | | 4 | 4 | 18 | 18 | 4 | 4 | 1 | 1 |
| Total | 65 of 65 | 0% | 0 of 0 | n/a | 4 | 4 | 18 | 18 | 4 | 4 | 1 | 1 |

Created with JaCoCo 0.8.12.202403310830

com.example.project

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|----------------------------------|---|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| Application.java | <div style="width: 0%; background-color: red;"></div> | 0% | n/a | | 4 | 4 | 18 | 18 | 4 | 4 | 1 | 1 |
| Total | 65 of 65 | 0% | 0 of 0 | n/a | 4 | 4 | 18 | 18 | 4 | 4 | 1 | 1 |

Created with JaCoCo 0.8.12.202403310830

GameScreenController SidebarLoaded

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| GameScreenController SidebarLoaded(Parent, SidebarController) | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 9 of 9 | 0% | 0 of 0 | n/a | 1 | 1 | 1 | 1 | 1 | 1 |

Created with JaCoCo 0.8.12.202403310830

Project

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| com.example.project.controllers.gameScreens | | 5% | | 2% | 141 | 151 | 435 | 468 | 92 | 102 | 8 | 11 |
| com.example.project.controllers.gameScreens.animations | | 0% | | 0% | 22 | 22 | 89 | 89 | 19 | 19 | 6 | 6 |
| com.example.project.services | | 58% | | 11% | 26 | 59 | 72 | 198 | 18 | 50 | 2 | 9 |
| com.example.project.services.sqlite.dAOs | | 52% | | 31% | 12 | 26 | 57 | 133 | 4 | 15 | 0 | 2 |
| com.example.project.models.tiles | | 72% | | 18% | 16 | 49 | 41 | 127 | 7 | 38 | 1 | 8 |
| com.example.project.models.gameScreens | | 82% | | 78% | 30 | 101 | 43 | 244 | 18 | 65 | 2 | 8 |
| com.example.project.controllers.popupControllers | | 0% | | 0% | 12 | 12 | 27 | 27 | 10 | 10 | 2 | 2 |
| com.example.project.controllers.tiles | | 62% | | 78% | 9 | 36 | 34 | 93 | 7 | 28 | 1 | 5 |
| com.example.project | | 0% | | n/a | 4 | 4 | 18 | 18 | 4 | 4 | 1 | 1 |
| com.example.project.controllers | | 0% | | 0% | 9 | 9 | 21 | 21 | 7 | 7 | 2 | 2 |
| com.example.project.services.sqlite | | 67% | | 75% | 1 | 14 | 5 | 24 | 0 | 12 | 0 | 3 |
| com.example.project.services.shopItems | | 84% | | 0% | 4 | 6 | 6 | 46 | 2 | 4 | 0 | 1 |
| com.example.project.models.tileGroups | | 95% | | 76% | 8 | 33 | 3 | 84 | 2 | 20 | 0 | 3 |
| com.example.project.models.popups | | 96% | | 75% | 1 | 12 | 1 | 23 | 0 | 10 | 0 | 2 |
| com.example.project.services.sound | | 100% | | 100% | 0 | 12 | 0 | 54 | 0 | 11 | 0 | 2 |
| com.example.project.models | | 100% | | n/a | 0 | 4 | 0 | 8 | 0 | 4 | 0 | 1 |
| Total | 4,065 of 7,541 | 46% | 184 of 294 | 37% | 295 | 550 | 852 | 1,657 | 190 | 399 | 25 | 66 |

Created with JaCoCo 0.8.12.202403310830

Sessions

This coverage report is based on execution data from the following sessions:

| Session | Start Time | Dump Time |
|------------------------|--------------------------|--------------------------|
| runnervmwhb2z-39ed52ec | Oct 24, 2025, 6:03:10 AM | Oct 24, 2025, 6:03:15 AM |

Execution data for the following classes is considered in this report:

| Class | Id |
|---|-------------------|
| C IntegrationTests.SQLite.DAOs.DictionaryDAOTests | cbc478615c608970 |
| C IntegrationTests.SQLite.SQLiteDictionaryConnectionTests | 48cce90217d0662 |
| C com.example.project.controllers.gameScreens.GameScreenController | a5c416048453017a |
| C com.example.project.controllers.gameScreens.LevelController | f087ed4a3583e474 |
| C com.example.project.controllers.gameScreens.LevelControllerTests | f2bf8ed73d5094d4 |
| C com.example.project.controllers.gameScreens.ShopController | f5c459561de4e34e |
| C com.example.project.controllers.gameScreens.ShopControllerTests | f0d0a6d93714d7d8 |
| C com.example.project.controllers.popupControllers.DefinitionController | a27e9fd845b67d9e |
| C com.example.project.controllers.tiles.EmptyTileControllerTests | 6497475060dddb55 |
| C com.example.project.controllers.tiles.EmptyTileSlotController | 18b6e94a91e2870d |
| C com.example.project.controllers.tiles.LetterTileController | 3155123f322d1e95 |
| C com.example.project.controllers.tiles.TileController | 472170f1175ccb7a |
| C com.example.project.controllers.tiles.TileControllerFactory | c8eecc2f00b38344 |
| C com.example.project.controllers.tiles.TileControllerFactoryTests | f42d29fed32812f |
| C com.example.project.controllers.tiles.TileControllerFactoryTests.UnsupportedTileModel | f68f61f240dc72ed |
| C com.example.project.controllers.tiles.UpgradeTileController | e82b0a2b77e89c80 |
| C com.example.project.controllers.tiles.UpgradeTileControllerTests | 6dc5393ffc2a3798 |
| C com.example.project.models.User | 73f2d93c7dc8d563 |
| C com.example.project.models.gameScreens.GameScreenModel | a02ec0117e959460 |
| C com.example.project.models.gameScreens.LevelModel | db61a48eb96f9331 |
| C com.example.project.models.gameScreens.LevelModelTests | ad854a0c1e4ebe7f |
| C com.example.project.models.gameScreens.LoginModel | cb6088a017ad9855 |
| C com.example.project.models.gameScreens.LoginModelTest | 8d85db1af9f00930 |
| C com.example.project.models.gameScreens.MainMenuModel | 3c45ba970243aac6 |
| C com.example.project.models.gameScreens.MainMenuModelTest | 5e19381269f9c9e0 |
| C com.example.project.models.gameScreens.ScoreChimePlayer | 305eee8b1be24dfe |
| C com.example.project.models.gameScreens.ShopModel | c38bbe926dda7010 |
| C com.example.project.models.gameScreens.ShopModelTests | 74bd0c9d304f9b3c |
| C com.example.project.models.popups.DefinitionPopup | c8f10388524de747 |
| C com.example.project.models.popups.DefinitionPopupTest | 8cbcfc00e98e10091 |
| C com.example.project.models.popups.PopupModel | d7d747ca54f3b887 |
| C com.example.project.models.popups.PopupModelTest | 04d5e36a76a2f67f |
| C com.example.project.models.popups.PopupModelTest.TestPopup | 3c2eea42713b0843 |
| C com.example.project.models.tileGroups.LetterTileGroup | 772523f0066a5e5e |
| C com.example.project.models.tileGroups.LetterTileGroupTests | ff80914c3c043aca |
| C com.example.project.models.tileGroups.TileGroup | 7e19ae733332145c |
| C com.example.project.models.tileGroups.UpgradeTileGroup | 929e544f2e2e1545 |
| C com.example.project.models.tileGroups.UpgradeTileGroupTests | 416d5d8f386f9b9e |
| C com.example.project.models.tiles.EmptyTileSlotModel | 560309348fcfd5aa7 |
| C com.example.project.models.tiles.EmptyTileSlotTests | eea0f1d61a927270 |
| C com.example.project.models.tiles.LetterTileModel | 9fd302f38c1f8cd1 |
| C com.example.project.models.tiles.LetterTileTest | c23418cec948c9d6 |
| C com.example.project.models.tiles.ScrabbleTileProvider | 27315692a76d665a |
| C com.example.project.models.tiles.ScrabbleTileProvider.ScrabbleTileData | a8fe0d118608ac4b |
| C com.example.project.models.tiles.ScrabbleTileProviderTests | 55e168852cf7ab64 |
| C com.example.project.models.tiles.TileModel | 7f89317b74b781f4 |
| C com.example.project.models.tiles.TileTests | f139067565a01480 |
| C com.example.project.models.tiles.TileTests.TestTile | 1ca92c58728a891 |
| C com.example.project.models.tiles.UpgradeTileModel | 122d054afe01fdf5 |
| C com.example.project.models.tiles.UpgradeTileModel.UpgradeBuilder | 8f7f4f25dfc1224b |
| C com.example.project.models.tiles.UpgradeTileModelTests | 3efadf905438cd91 |
| C com.example.project.services.FXMLPageLoader | fc465941ce8499fa |
| C com.example.project.services.GameScene | be9d54d3c1ad3e88 |
| C com.example.project.services.Logger | a9bb6c27fd3106e |
| C com.example.project.services.LoggerTests | c1ae23810f932124 |
| C com.example.project.services.PasswordHasher | 21259083042798f8 |
| C com.example.project.services.PasswordHasherTests | ab9b14223d126452 |
| C com.example.project.services.PopupLoader | 81a0bc8fbeb958e2 |
| C com.example.project.services.PopupLoaderTests | 4a1e5fed2011cccc |
| C com.example.project.services.SceneManager | 2c7e1a920d680f48 |
| C com.example.project.services.SceneManagerTests | 2fc31a4a2c7dbcfc8 |
| C com.example.project.services.Session | 08b52d5bbf8413f2 |
| C com.example.project.services.SessionTests | 92bf2a443c991b99 |
| C com.example.project.services.shopItems.UpgradeTiles | d21024558dc878b4 |

| | |
|--|-------------------|
| com.example.project.services.sound.GameMusicPlayer | aa5a7a44e9cf5149 |
| com.example.project.services.sound.GameMusicPlayerTests | 3ef4c70490b0845c |
| com.example.project.services.sound.GameSoundPlayer | d9fd4a247737058d |
| com.example.project.services.sound.GameSoundPlayerTests | 5ed8bc9023ccee40 |
| com.example.project.services.sqlite.ConnectionComparer | 3788cb41159b3e33 |
| com.example.project.services.sqlite.SQLiteConnection | 99d0f75cd42ab9bc |
| com.example.project.services.sqlite.SQLiteDictionaryConnection | dd0780a3988434c0 |
| com.example.project.services.sqlite.SQLiteUsersConnection | 75961846f7046111 |
| com.example.project.services.sqlite.dAOs.DictionaryDAO | 2e2be8bd42950ccb |
| com.example.project.services.sqlite.dAOs.DictionaryDAOTests | 59928558d04cdf1 |
| com.example.project.services.sqlite.dAOs.SQLiteDBCreator | 8d3f0f1224c95ccf |
| com.example.project.services.sqlite.dAOs.UsersDAO | d21c07f35b47d2c8 |
| com.example.project.services.sqlite.dAOs.UsersDAOTests | a414ca3d9e49520d |
| com.example.project.testHelpers.MockAudioSystemExtension | ecc3043b791c9ac |
| com.sun.glass.ui.Application | d2c633053effff922 |
| com.sun.glass.ui.View.Capability | 137b2444912a7638 |
| com.sun.glass.utils.NativeLibLoader | 2756c6e87ea3573a |
| com.sun.javafx.PlatformUtil | 180f2410faf7cd2c |
| com.sun.javafx.application.PlatformImpl | 91660b3e95e248d3 |
| com.sun.javafx.application.PlatformImpl.2 | b971d51ebea10887 |
| com.sun.javafx.application.PlatformImpl.HighContrastScheme | d2d4a159e49aa741 |
| com.sun.javafx.beans.event.AbstractNotifyListener | 2a527122b153d8f7 |
| com.sun.javafx.binding.ExpressionHelper | f926da9605240bf |
| com.sun.javafx.binding.ExpressionHelperBase | 9465a503879a5bb7 |
| com.sun.javafx.binding.ListExpressionHelper | 6be51d95359214b6 |
| com.sun.javafx.binding.ListExpressionHelper.Generic | ecc179e6031e116e |
| com.sun.javafx.binding.ListExpressionHelper.SingleChange | a0e45bc5db7b1d8f |
| com.sun.javafx.collections.ListListenerHelper | 77e86a2f08f0d7be |
| com.sun.javafx.collections.ListListenerHelper.Generic | 6ba162bb98d6936f |
| com.sun.javafx.collections.ListListenerHelper.SingleChange | a74f7e03718465a8 |
| com.sun.javafx.collections.MapListenerHelper | 83a6071d9d8e996c |
| com.sun.javafx.collections.ObservableListWrapper | 43e38702734066b8 |
| com.sun.javafx.collections.ObservableMapWrapper | 9beb57dce3694f06 |
| com.sun.javafx.collections.ObservableMapWrapper.SimpleChange | 6d2a9cb0f6001132 |
| com.sun.javafx.collections.SourceAdapterChange | 22fe67ed36a975b9 |
| com.sun.javafx.collections.TrackableObservableList | d30f4d2fe4b29e79 |
| com.sun.javafx.collections.VetoableListDecorator | 347d1dd01557fc01 |
| com.sun.javafx.css.BitSet | ab2966ccf18a1d6f |
| com.sun.javafx.css.PseudoClassImpl | 3f1829ba7270d233 |
| com.sun.javafx.css.PseudoClassState | 6f7e579d2277afb4 |
| com.sun.javafx.event.BasicEventDispatcher | aa45d7a164f70eb3 |
| com.sun.javafx.event.CompositeEventDispatcher | 1d685633a1724ba3 |
| com.sun.javafx.event.CompositeEventHandler | 1e76a577da3cf8c4 |
| com.sun.javafx.event.EventHandlerManager | fd757caa74231309 |
| com.sun.javafx.fxml.BeanAdapter | 2f491e945420e949 |
| com.sun.javafx.fxml.BeanAdapter.1 | 54763299be740b64 |
| com.sun.javafx.fxml.BeanAdapter.MethodCache | 7ca3780f0a0405b6 |
| com.sun.javafx.fxml.FXMLLoaderHelper | ef6e8fbe52a02992 |
| com.sun.javafx.fxml.MethodHelper | c772402c943b6f9f |
| com.sun.javafx.fxml.ModuleHelper | 9001ae8bc94c3759 |
| com.sun.javafx.geom.BaseBounds | ed0a354be28d924a |
| com.sun.javafx.geom.RectBounds | 8919b9f96338309c |
| com.sun.javafx.geom.RectangularShape | 7fd95f3413eb8787 |
| com.sun.javafx.geom.RoundRectangle2D | c717cc2e84756909 |
| com.sun.javafx.geom.Shape | c8bc730560d51518 |
| com.sun.javafx.geom.Vec2d | d828757c2463f252 |
| com.sun.javafx.geom.transform.BaseTransform | 1eb9022f26ea24ef |
| com.sun.javafx.geom.transform.Identity | 797e8fc16bcdcbdf |
| com.sun.javafx.perf.PerformanceTracker | c0073c4114dae170 |
| com.sun.javafx.reflect.ConstructorUtil | 61fe60a532acbe26 |
| com.sun.javafx.reflect.MethodUtil | 2df70e9f3f59b0d |
| com.sun.javafx.reflect.MethodUtil.1 | 411edf81fa5fe806 |
| com.sun.javafx.reflect.ReflectUtil | 955b6e20e3cf3988 |
| com.sun.javafx.runtime.VersionInfo | 097d2593b109124b |
| com.sun.javafx.scene.CssFlags | 7074f98c6728098c |
| com.sun.javafx.scene.DirtyBits | 6d9772ee39dddc4c |
| com.sun.javafx.scene.EnteredExitedHandler | b8c2963c3fde263f |
| com.sun.javafx.scene.EventHandlerProperties | 648a60aa74c8d9a6 |
| com.sun.javafx.scene.EventHandlerProperties.EventHandlerProperty | 666b1b8358e54996 |
| com.sun.javafx.scene.ImageViewHelper | c083c8f185c5da1d |
| com.sun.javafx.scene.LayoutFlags | 0df59ec15bbc4d05 |
| com.sun.javafx.scene.NodeEventDispatcher | 3b549faa95a5d88b |
| com.sun.javafx.scene.NodeHelper | a4b93b169753f510 |
| com.sun.javafx.scene.ParentHelper | c23f48847b5f5a46 |
| com.sun.javafx.scene.layout.PaneHelper | a289606e8fb303fb |

| | |
|--|-------------------|
| com.sun.javafx.scene.layout.RegionHelper | 97ad39fc0e16ade1 |
| com.sun.javafx.scene.shape.RectangleHelper | 912449853f782479 |
| com.sun.javafx.scene.shape.ShapeHelper | 9d6790c3d4b6e077 |
| com.sun.javafx.scene.transform.TransformHelper | 1f659cc46e4280c5 |
| com.sun.javafx.sg.prism.NGShape.Mode | 92ac86dee521cb6d |
| com.sun.javafx.tk.Toolkit | aa53129eb4a5d031 |
| com.sun.javafx.tk.quantum.GlassSystemMenu | de866cf40dbbee0eb |
| com.sun.javafx.tk.quantum.PaintCollector | 44df43dc614d8fbe |
| com.sun.javafx.tk.quantum.PerformanceTrackerHelper | 2f870f9613045e15 |
| com.sun.javafx.tk.quantum.PerformanceTrackerHelper.1 | d6105bcb01c52025 |
| com.sun.javafx.tk.quantum.PerformanceTrackerHelper.PerformanceTrackerDummyImpl | cd4b8ff71678a2253 |
| com.sun.javafx.tk.quantum.PerformanceTrackerImpl | 5a12f07b8477f48e |
| com.sun.javafx.tk.quantum.QuantumRenderer | d35e94179b5d12af |
| com.sun.javafx.tk.quantum.QuantumRenderer.PipelineRunnable | 9245b3aa82fb9b1 |
| com.sun.javafx.tk.quantum.QuantumRenderer.QuantumThreadFactory | 5247389d1c9eb181 |
| com.sun.javafx.tk.quantum.QuantumToolkit | 0ac1bcd0138cfbee |
| com.sun.javafx.tk.quantum.QuantumToolkit.1 | c1f65e7a1298638d |
| com.sun.javafx.tk.quantum.QuantumToolkit.3 | 1b3c034cb8122f46 |
| com.sun.javafx.tk.quantum.QuantumToolkit.PulseTask | 986af6e7eae3b82f |
| com.sun.javafx.util.Utils | d2f9b00a9a3e4b69 |
| com.sun.prism.BasicStroke | 3417623692b8498e |
| com.sun.prism.GraphicsPipeline | 1ea8581ec0cac4c2 |
| com.sun.prism.es2.ES2Pipeline | 12c69b05002c53d2 |
| com.sun.prism.es2.GLFactory | de00ec3076b6872 |
| com.sun.prism.es2.FactoryLoader | 71b6ea996e4d096 |
| com.sun.prism.es2.GLGPUInfo | 23a63eb0b2f3c8a4 |
| com.sun.prism.es2.GLPixelFormat | 5ca3832be6da3d2d |
| com.sun.prism.es2.GLPixelFormat.Attributes | f0a08c0d5bbfe95d |
| com.sun.prism.es2.X11GLFactory | 97cd118484915fa9 |
| com.sun.prism.impl.PrismSettings | 832493b2febdafc7 |
| com.sun.prism.impl.PrismSettings.RasterizerType | 7fe513df0e240ce3 |
| com.sun.prism.sw.SWPipeline | fc33f4f522d468ce |
| java.sql.DriverInfo | ea91df4c5e4e8e22 |
| java.sql.DriverManager | fe9c6ae8f9ed72e9 |
| java.sql.DriverManager.1 | 2ea175afe4482980 |
| java.sql.DriverManager.2 | 60e8a2c0231a1a2e |
| java.sql.SQLException | c2a7742cda8a83ff |
| java.sql.SQLPermission | 545a7e448515d9f6 |
| javafx.application.ConditionalFeature | 4de75e8169bbb408 |
| javafx.application.Platform | 7704090522af6319 |
| javafx.beans.WeakInvalidationListener | fc4d99766f197be1 |
| javafx.beans.binding.BooleanExpression | b1becc7549966757 |
| javafx.beans.binding.DoubleExpression | 10bb2e36be67250e |
| javafx.beans.binding.IntegerExpression | ac1a3e7f2ec46b03 |
| javafx.beans.binding.ListExpression | 525dc5226e624658 |
| javafx.beans.binding.NumberExpressionBase | 70a4960fb7c7a0ad |
| javafx.beans.binding.ObjectExpression | 71fce2987e1e3a7 |
| javafx.beans.binding.StringExpression | fad2310c05aa584d |
| javafx.beans.property.BooleanProperty | 1f157f7ffe1d66b2 |
| javafx.beans.property.BooleanPropertyBase | 18da3c04ea61d9a7 |
| javafx.beans.property.DoubleProperty | 22c72527e41c548a |
| javafx.beans.property.DoublePropertyBase | c8058b74ae1b4aec |
| javafx.beans.property.IntegerProperty | 860f9829eea70f98 |
| javafx.beans.property.IntegerPropertyBase | 3dd9f4069b6bdd90 |
| javafx.beans.property.ListProperty | 29alec9e1248f4fe |
| javafx.beans.property.ListPropertyBase | 7e21b5955a903554 |
| javafx.beans.property.ObjectProperty | 50a7a1c50070f772 |
| javafx.beans.property.ObjectPropertyBase | 1b77caf6ee06f0c5 |
| javafx.beans.property.ReadOnlyBooleanProperty | 66c2d9017baabc55 |
| javafx.beans.property.ReadOnlyBooleanPropertyBase | 0554ba38d8176bfe |
| javafx.beans.property.ReadOnlyBooleanWrapper | 05943b8893bc32ea |
| javafx.beans.property.ReadOnlyBooleanWrapper.ReadOnlyPropertyImpl | dfa0cd2532c362c7 |
| javafx.beans.property.ReadOnlyDoubleProperty | 81793df4b3a47322 |
| javafx.beans.property.ReadOnlyDoubleWrapper | 17a5e3dbe87d8c8 |
| javafx.beans.property.ReadOnlyIntegerProperty | b13c35e24e34c27e |
| javafx.beans.property.ReadOnlyIntegerPropertyBase | b3a87bcf2e9fa979 |
| javafx.beans.property.ReadOnlyIntegerWrapper | d147b7f5307c06bc |
| javafx.beans.property.ReadOnlyIntegerWrapper.ReadOnlyPropertyImpl | 9625fc21b7fbff3 |
| javafx.beans.property.ReadOnlyListProperty | 326c5d21b5e41bbe |
| javafx.beans.property.ReadOnlyListPropertyBase | 0402abd0036925b8 |
| javafx.beans.property.ReadOnlyListWrapper | 62924df6ddbfb859 |
| javafx.beans.property.ReadOnlyListWrapper.ReadOnlyPropertyImpl | 2d8a16c775c539b6 |
| javafx.beans.property.ReadOnlyObjectProperty | 339ea559d888c0ce |
| javafx.beans.property.ReadOnlyObjectPropertyBase | f87507710d6dbf30 |
| javafx.beans.property.ReadOnlyObjectWrapper | 7ef6a5d83ad3373c |

| | |
|--|-------------------|
| javafx.beans.property.ReadOnlyObjectWrapper.ReadOnlyPropertyImpl | 6e02d426b4689162 |
| javafx.beans.property.ReadOnlyStringProperty | 6b8ef23028fa46eb |
| javafx.beans.property.ReadOnlyStringPropertyBase | 8bc a58b879328b72 |
| javafx.beans.property.ReadOnlyStringWrapper | 635d11120cae20f8 |
| javafx.beans.property.ReadOnlyStringWrapper.ReadOnlyPropertyImpl | b7aabc847e1dc97 |
| javafx.beans.property.SimpleBooleanProperty | e2d31f507f6d1bb8 |
| javafx.beans.property.SimpleDoubleProperty | 509a5d3a3bcf81a3 |
| javafx.beans.property.SimpleIntegerProperty | aed4a9da35d48da7 |
| javafx.beans.property.SimpleListProperty | e83a8061470a66f2 |
| javafx.beans.property.SimpleObjectProperty | 8244731f22cda9a9 |
| javafx.beans.property.SimpleStringProperty | 86674865c31bf610 |
| javafx.beans.property.StringProperty | aee26901ce9c5769 |
| javafx.beans.property.StringPropertyBase | 60fce33df74167c0 |
| javafx.collections.FXCollections | 046e1fbff2414879 |
| javafx.collections.FXCollections.EmptyObservableList | a548f7960565182a |
| javafx.collections.FXCollections.EmptyObservableList.1 | fbca10af75c2fd23a |
| javafx.collections.FXCollections.EmptyObservableMap | 4180d2ac0b81de99 |
| javafx.collections.FXCollections.EmptyObservableSet | 35a313fd7cc99877 |
| javafx.collections.FXCollections.EmptyObservableSet.1 | 4394fa5b7c103428 |
| javafx.collections.FXCollections.UnmodifiableObservableListImpl | 81254e56283c2213 |
| javafx.collections.FXCollections.UnmodifiableObservableSet | 7d47d4a75de603fe |
| javafx.collections.ListChangeBuilder | 8958dfad8635242e |
| javafx.collections.ListChangeBuilder.SingleChange | 2472a7ae75c80a63 |
| javafx.collections.ListChangeBuilder.SubChange | 473eb48dab5e3f55 |
| javafx.collections.ListChangeListener.Change | ff56149b808c32b3 |
| javafx.collections.MapChangeListener.Change | 8f5e1c5decb27258 |
| javafx.collections.ModifiableObservableListBase | d4f104fd3a700623 |
| javafx.collections.ObservableList.MockitoMock.VKYUIYts | b7f535a084bc2ff7 |
| javafx.collections.ObservableListBase | 36500a719201d2f4 |
| javafx.collections.WeakListChangeListener | 2e10b4d0553c7631 |
| javafx.css.PseudoClass | e96d5c022f1a335e |
| javafx.css.StyleOrigin | ae6e54fc8b969978 |
| javafx.css.StyleableDoubleProperty | 56db11910ecfb4ef |
| javafx.css.StyleableObjectProperty | 02667de3152973b3 |
| javafx.event.Event | 780b729a667b2865 |
| javafx.event.EventType | ae7afab5bbc83a2 |
| javafx.fxml.FXMLLoader | eb3f0b84a4f8d82e |
| javafx.fxml.FXMLLoader.1 | c1c995e38d84fcfcf |
| javafx.fxml.FXMLLoader.2 | e143ce550df057b1 |
| javafx.fxml.FXMLLoader.3 | 03a848685f1dd34e |
| javafx.fxml.FXMLLoader.Attribute | b7134d3e7cf4838b |
| javafx.fxml.FXMLLoader.ControllerAccessor | a761524b005113e0 |
| javafx.fxml.FXMLLoader.ControllerAccessor.1 | 9381a5881692dda9 |
| javafx.fxml.FXMLLoader.Element | 41a073e82c0dac3b |
| javafx.fxml.FXMLLoader.InstanceDeclarationElement | 465b2ed425462588 |
| javafx.fxml.FXMLLoader.SupportedType | ae16cd64f693ee9d |
| javafx.fxml.FXMLLoader.SupportedType.1 | a223674fda652bbc |
| javafx.fxml.FXMLLoader.SupportedType.2 | 68e64353d2b59749 |
| javafx.fxml.FXMLLoader.SupportedType.3 | b628cc6f0f67ae5c |
| javafx.fxml.FXMLLoader.SupportedType.4 | d7a80cb2c36edbee |
| javafx.fxml.FXMLLoader.SupportedType.5 | ee4bc4f0d06898f0 |
| javafx.fxml.FXMLLoader.SupportedType.6 | 842aed37912d33b0 |
| javafx.fxml.FXMLLoader.ValueElement | 3e8c2ad2e1362731 |
| javafx.fxml.JavaFXBuilderFactory | 52aaef54c3ceb3d85 |
| javafx.geometry.Insets | 93efb0eb3777987d |
| javafx.geometry.NodeOrientation | 9676323e71e51d05 |
| javafx.geometry.Point3D | f3d94acee5cc82ea |
| javafx.scene.AccessibleAttribute | a34c5945ad0d13bd |
| javafx.scene.AccessibleRole | 1a7be54cc8b0fc51 |
| javafx.scene.CacheHint | 68f1cae20270bb2e |
| javafx.scene.DepthTest | d21b142391df7715 |
| javafx.scene.Node | 597ea66bee1cc58c |
| javafx.scene.Node.1 | 220ac68b2d872a92 |
| javafx.scene.Node.13 | 12c1a8193f5ed173 |
| javafx.scene.Node.17 | 82730bc4e3cf2a6e |
| javafx.scene.Node.18 | 7150e1cb3ac8d045 |
| javafx.scene.Node.2 | 7658d20ff953c4d0 |
| javafx.scene.Node.3 | 83ccf6a1d784fdac |
| javafx.scene.Node.4 | fd1746a2e52f5962 |
| javafx.scene.Node.9 | bfe57129ad855c7e |
| javafx.scene.Node.FocusPropertyBase | 6186be3b3ce4e613 |
| javafx.scene.Node.FocusWithinProperty | e1245529e8dc5a8 |
| javafx.scene.Node.LazyBoundsProperty | 448e1c6d09e38fa8 |
| javafx.scene.Node.NodeTransformation | 1adb43f3c335d2e2 |

| | |
|--|-------------------|
| javafx.scene.Node.NodeTransformation.5 | 9ed67bbba856b13b |
| javafx.scene.Node.NodeTransformation.6 | cf8e6e2840547a35 |
| javafx.scene.Node.ReadOnlyObjectWrapperManualFire | ca3ee85b7935758d |
| javafx.scene.Node.TreeVisibleProperty | 8971fcb2890fab06 |
| javafx.scene.Parent | 5c806511f80e5bb8 |
| javafx.scene.Parent.1 | 7db4e9f30e5f5563 |
| javafx.scene.Parent.2 | 851977cb78c8c14d |
| javafx.scene.Parent.3 | 19244e4c5182952e |
| javafx.scene.Parent.4 | 960c69df124b8173 |
| javafx.scene.PropertyHelper | 29eca032a498fd43 |
| javafx.scene.image.Image | 2a66305e140044f6 |
| javafx.scene.image.Image.1 | 1fb4f045d048b979 |
| javafx.scene.image.ImageView | 0c60924211773384 |
| javafx.scene.image.ImageView.1 | c67458f4f6e78c6e |
| javafx.scene.input.InputEvent | 811f29761efc0ef2 |
| javafx.scene.input.MouseEvent | e99e6768b643f90b |
| javafx.scene.layout.Pane | fb8f6ebc35f376e3 |
| javafx.scene.layout.Pane.1 | 358f55173e4f006b |
| javafx.scene.layout.Region | df888f21f609e4a2 |
| javafx.scene.layout.Region.1 | 0905e0fc1786e24c |
| javafx.scene.layout.Region.3 | 181a08ba2894eca2 |
| javafx.scene.layout.Region.4 | 5a2d84f77dde14ea |
| javafx.scene.layout.Region.5 | 0b43493b501702f4 |
| javafx.scene.layout.RegionInsetsProperty | f0d5329d865cf7f0 |
| javafx.scene.layout.StackPane | a17af0c7080247a3 |
| javafx.scene.paint.Color | 3862b8439e4f22ad |
| javafx.scene.paint.Paint | c27464b04839934b |
| javafx.scene.paint.Paint.1 | 6b3cefda6a177ae2 |
| javafx.scene.shape.Rectangle | 2692e054a7266682 |
| javafx.scene.shape.Rectangle.1 | 558a8b89681a4e7f |
| javafx.scene.shape.Rectangle.4 | 0a71374a9652199a |
| javafx.scene.shape.Rectangle.5 | 126c82028549fde0 |
| javafx.scene.shape.Rectangle.6 | da461ccdf9a0696b |
| javafx.scene.shape.Rectangle.7 | 2423b68fe48878d5 |
| javafx.scene.shape.Shape | 5c9474d93604c42f |
| javafx.scene.shape.Shape.1 | b519cc3a9b89933b |
| javafx.scene.shape.Shape.2 | 628a35ec7e11529e |
| javafx.scene.shape.Shape.3 | 6546ed06f482fb75 |
| javafx.scene.shape.StrokeLineCap | 593798e5a68ddda1 |
| javafx.scene.shape.StrokeLineJoin | d35288540e5a9e82 |
| javafx.scene.shape.StrokeType | deee5f164e2dac50 |
| javafx.scene.transform.Rotate | 50f1bc2c92d9276c |
| javafx.scene.transform.Transform | e1876168d76b1cc7 |
| javafx.scene.transform.Transform.1 | ae38419af1c35779 |
| javafx.util.FXPermission | 8327cec7275ee5cd |
| net.bytebuddy.ByteBuddy | 8a25cfee68883757 |
| net.bytebuddy.ClassFileVersion | da7c006b44346640 |
| net.bytebuddy.ClassFileVersion.VersionLocator.Resolved | a389024132c0cc61 |
| net.bytebuddy.ClassFileVersion.VersionLocator.Resolver | 6f0a8d6c5c95d22c |
| net.bytebuddy.NamingStrategy.AbstractBase | 47d6d27e15064a2b |
| net.bytebuddy.NamingStrategy.Suffixing | 83ffc7a50d03e29a |
| net.bytebuddy.NamingStrategy.Suffixing.BaseNameResolver.ForUnnamedType | dc4c226cbd0897a8 |
| net.bytebuddy.NamingStrategy.SuffixingRandom | c7e2e00a103cd13c |
| net.bytebuddy.TypeCache | 05243229e50ea1b0 |
| net.bytebuddy.TypeCache.LookupKey | 599592f0f74bbe07 |
| net.bytebuddy.TypeCache.SimpleKey | aab9b5395600f0fd |
| net.bytebuddy.TypeCache.Sort | 2863b0d48a0f3008 |
| net.bytebuddy.TypeCache.Sort.1 | ba4f2805581e1090 |
| net.bytebuddy.TypeCache.Sort.2 | db80a28c74867927 |
| net.bytebuddy.TypeCache.Sort.3 | 65abff93a29d75a2 |
| net.bytebuddy.TypeCache.StorageKey | b7a810d4119627f9 |
| net.bytebuddy.TypeCache.WithInlineExpunction | 131bdbf6b155e141 |
| net.bytebuddy.asm.Advice | 70b624c9f019e2d7 |
| net.bytebuddy.asm.Advice.AdviceVisitor | fabc880f6c0d5cd1 |
| net.bytebuddy.asm.Advice.AdviceVisitor.WithExitAdvice | e5e4c6806aaeee24 |
| net.bytebuddy.asm.Advice.AdviceVisitor.WithExitAdvice.WithoutExceptionHandling | ee7c0bbc4d058e48 |
| net.bytebuddy.asm.Advice.ArgumentHandler.Factory | 88f478d1b3b07ef4 |
| net.bytebuddy.asm.Advice.ArgumentHandler.Factory.1 | 5b27bf7bbbedbea54 |
| net.bytebuddy.asm.Advice.ArgumentHandler.Factory.2 | 57b0b2625db26866 |
| net.bytebuddy.asm.Advice.ArgumentHandler.ForAdvice.Default | 8f1d6e695fd7ed52 |
| net.bytebuddy.asm.Advice.ArgumentHandler.ForAdvice.Default.ForMethodEnter | 2298e62edd4a8f61 |
| net.bytebuddy.asm.Advice.ArgumentHandler.ForAdvice.Default.ForMethodExit | d03149f225f747e5 |
| net.bytebuddy.asm.Advice.ArgumentHandler.ForInstrumentedMethod.Default | 3ac42d24c686a98d |
| net.bytebuddy.asm.Advice.ArgumentHandler.ForInstrumentedMethod.Default.Copying | f3d840402494b1eb |
| net.bytebuddy.asm.Advice.Delegator.ForRegularInvocation.Factory | 5378b1b8ea4a3362 |

| | |
|--|-------------------|
| net.bytebuddy.asm.Advice.Dispatcher | 0e7fef258bff623e |
| net.bytebuddy.asm.Advice.Dispatcher.Inactive | 17d1a367a7c4f802 |
| net.bytebuddy.asm.Advice.Dispatcher.Inlining | c3ce45049068fa1f |
| net.bytebuddy.asm.Advice.DispatcherInlining.CodeTranslationVisitor | 1504514b9b3dac05 |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved | 6de58a2c31e6f05a |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.AdviceMethodInliner | 35df575204406b21 |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.AdviceMethodInliner.ExceptionTableCollector | 5c4bed9acb4fe33a |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.AdviceMethodInliner.ExceptionTableExtractor | fb86d13a4d8b8eb1 |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.AdviceMethodInliner.ExceptionTableSubstitutor | dec4d3e69163577d |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.ForMethodEnter | 34cb9d6bc3c7cc9a |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.ForMethodEnter.WithRetainedEnterType | aed2977cf6532c53 |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.ForMethodExit | 57a7e629b3866515 |
| net.bytebuddy.asm.Advice.DispatcherInlining.Resolved.ForMethodExit.WithoutExceptionHandler | 4677875bb641e3ed |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.Disabled | 85ecfbf093d6a3b1 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForType | fa2fbcc6481a65a5 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue | 5b7eff5293f1f424 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.1 | d679d59d5fd6c1f6 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.2 | 2e3770e70234e10e |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.3 | fe0c7bd84802fd9 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.4 | 1aabf94ae884226 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.5 | f45edbabe0e9c5f |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.6 | 67ef92acb77ae368 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.7 | b2a655f0f65d66c8 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.8 | e7434f88906a9c24 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.9 | b84a4f1e7e3ce186 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.Bound | 243cd25d86e4ab5 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.ForValue.OfNonDefault | fea862c952f8245 |
| net.bytebuddy.asm.Advice.Dispatcher.RelocationHandler.Relocation.ForLabel | 2ca8b3bcf6049a33 |
| net.bytebuddy.asm.Advice.Dispatcher.Resolved.AbstractBase | c11dfd520c9193f |
| net.bytebuddy.asm.Advice.Dispatcher.SuspensionHandler.NoOp | c509d4e15ee51348 |
| net.bytebuddy.asm.Advice.Dispatcher.SuspensionHandler.Suppressing | 09e072bc1ca4ce4c |
| net.bytebuddy.asm.Advice.ExceptionHandler.Default | 5150bcf8b7ca4eb0 |
| net.bytebuddy.asm.Advice.ExceptionHandler.Default.1 | b1848b8f9d54a2a8 |
| net.bytebuddy.asm.Advice.ExceptionHandler.Default.2 | ee9674baa95729a9 |
| net.bytebuddy.asm.Advice.ExceptionHandler.Default.3 | 2039586b2e878fb7 |
| net.bytebuddy.asm.Advice.MethodSizeHandler.Default | b7646d7a13f17583 |
| net.bytebuddy.asm.Advice.MethodSizeHandler.Default.ForAdvice | 42c9c05091c4a149 |
| net.bytebuddy.asm.Advice.MethodSizeHandler.Default.WithCopiedArguments | 613e58a9d40f4543 |
| net.bytebuddy.asm.Advice.NoExceptionHandler | 85ae8e6a09ca4f81 |
| net.bytebuddy.asm.Advice.OffsetMapping.Factory.AdviceType | dd14d1576caa2ad4 |
| net.bytebuddy.asm.Advice.OffsetMapping.Factory.Illegal | fa561d82583ef937 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForAllArguments | 872309721f07bc32 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForAllArguments.Factory | e34fac7274362266 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForArgument | ff73ee4c55cc6164 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForArgument.Unresolved | 9551e89b98949a4c |
| net.bytebuddy.asm.Advice.OffsetMapping.ForArgument.Unresolved.Factory | 278dd3908309ea1a |
| net.bytebuddy.asm.Advice.OffsetMapping.ForDynamicConstant.Factory | a14df00a10dbe7d9 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForEnterValue | 3ed67498436236ab |
| net.bytebuddy.asm.Advice.OffsetMapping.ForEnterValue.Factory | 290ac0d704269bd6 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForExitValue.Factory | 988128caa26d936c |
| net.bytebuddy.asm.Advice.OffsetMapping.ForField.Unresolved.Factory | 690f359d0e8a8eee |
| net.bytebuddy.asm.Advice.OffsetMapping.ForFieldHandle.Unresolved.ReaderFactory | 4af22e2a757bf31a |
| net.bytebuddy.asm.Advice.OffsetMapping.ForFieldHandle.Unresolved.WriterFactory | 8fb88f057a84ce01 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForHandle.Factory | fc4d8ad5640502b1 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForInstrumentedMethod | f9c766b7738a0693 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForInstrumentedMethod.1 | 7f8b1a3dde7a0c48 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForInstrumentedMethod.2 | 2440d1f955b26dbe |
| net.bytebuddy.asm.Advice.OffsetMapping.ForInstrumentedMethod.3 | 78e7d35d754c495a |
| net.bytebuddy.asm.Advice.OffsetMapping.ForInstrumentedMethod.4 | 6795cd6ec57c163 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForInstrumentedMethod.5 | 29f0de3fb8feaef4 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForInstrumentedType | 9cb199178b40c2e0 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForLocalValue.Factory | dee082d8fb8c9059 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForOrigin.Factory | cc27f773ada1ac11 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForReturnValue | df9ae25477abe21 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForReturnValue.Factory | 3f8ddbaca9c0ca46 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForSelfCallHandle.Factory | f4e803bdd3ec001f |
| net.bytebuddy.asm.Advice.OffsetMapping.ForStackManipulation | 42fae42b06e0ab7d |
| net.bytebuddy.asm.Advice.OffsetMapping.ForStackManipulation.Factory | 2b1ef8ab130df062 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForStubValue | f6e4b7bed2d4994e |
| net.bytebuddy.asm.Advice.OffsetMapping.ForThisReference | 61bdbcb141a58594 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForThisReference.Factory | e6438a2fa03556fa |
| net.bytebuddy.asm.Advice.OffsetMapping.ForThrowable.Factory | 8258647049b8eb56 |
| net.bytebuddy.asm.Advice.OffsetMapping.ForUnusedValue.Factory | bfd367cdcaa8cf459 |
| net.bytebuddy.asm.Advice.OffsetMapping.Sort | fb270c614c10d4b0 |

| | |
|---|-------------------|
| net.bytebuddy.asm.Advice.OffsetMapping.Sort.1 | 621f2e7f3b3b6565 |
| net.bytebuddy.asm.Advice.OffsetMapping.Sort.2 | ace13bc50cb664fa |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForArray | f0f6d54d957fc045 |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForArray.ReadOnly | 0f9962dd8bc7de98 |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForDefaultValue | 6d8821b5f8b246b1 |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForDefaultValue.ReadWrite | 6cb12c2494912a7b |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForStackManipulation | fccfc9f054f497ff |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForVariable | 24b0c4f2c614c48c |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForVariable.ReadOnly | b6c7fd082249ae77 |
| net.bytebuddy.asm.Advice.OffsetMapping.Target.ForVariable.ReadWrite | 0c416afdf59072e |
| net.bytebuddy.asm.Advice.PostProcessor.NoOp | 1512979491b2467e |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default | 33caa14d82b5421e |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.ForAdvice | d4b482c28c799b9f |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.Initialization | 76b8e9fc7058856 |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.Initialization.1 | f1e81aecea736054 |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.Initialization.2 | 5ee1fd4ae8179ac1 |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.TranslationMode | 2f8f1e9d91c57542 |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.TranslationMode.1 | c94ef73f34640c1b |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.TranslationMode.2 | 87cf86c20729f65e |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.TranslationMode.3 | e6194c7d74113b12 |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.WithPreservedArguments | 3a5b748d9afc1836 |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.Default.WithPreservedArguments.WithArgumentCopy | 894c7278034b34b6 |
| net.bytebuddy.asm.Advice.StackMapFrameHandler.NoOp | dfc8866815118963 |
| net.bytebuddy.asm.Advice.WithCustomMapping | b39f1cffed199307 |
| net.bytebuddy.asm.AsmVisitorWrapper.AbstractBase | 1e9138fb1b0b4185 |
| net.bytebuddy.asm.AsmVisitorWrapper.Compound | 0648db34b4a99a08 |
| net.bytebuddy.asm.AsmVisitorWrapper.ForDeclaredMethods | cf768d8f5b16d996 |
| net.bytebuddy.asm.AsmVisitorWrapper.ForDeclaredMethods.DispatchingVisitor | 6468c29bcf0a7088 |
| net.bytebuddy.asm.AsmVisitorWrapper.ForDeclaredMethods.Entry | ea3722f7f3e6a8e8 |
| net.bytebuddy.asm.AsmVisitorWrapper.NoOp | 6782431026ffffec0 |
| net.bytebuddy.description.ByteCodeElement.Token.TokenList | c378dffbc959a48 |
| net.bytebuddy.description.ModifierReviewable.AbstractBase | 2ea5306ed3260bac |
| net.bytebuddy.description.NamedElement.WithDescriptor | 79917fb7c151850b |
| net.bytebuddy.description.TypeVariableSource.AbstractBase | afa9cbe714a69ea6 |
| net.bytebuddy.description.annotation.AnnotationDescription | 7b5d7f348452a8fd |
| net.bytebuddy.description.annotation.AnnotationDescription.AbstractBase | 91fe2a0215db7a25 |
| net.bytebuddy.description.annotation.AnnotationDescription.ForLoadedAnnotation | 45b968ae6eb349e2 |
| net.bytebuddy.description.annotation.AnnotationList.AbstractBase | b2574970a6108617 |
| net.bytebuddy.description.annotation.AnnotationList.Empty | 85778debddc6a8e2 |
| net.bytebuddy.description.annotation.AnnotationList.Explicit | 45fb4e2b44a1ab64 |
| net.bytebuddy.description.annotation.AnnotationList.ForLoadedAnnotations | 309c064a5edad9f1 |
| net.bytebuddy.description.annotation.AnnotationSource.Empty | 3efda4f89915900c |
| net.bytebuddy.description.annotation.AnnotationValue | d7f3d011bdffb51e |
| net.bytebuddy.description.annotation.AnnotationValue.AbstractBase | 69e2bb699468284a |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant | 7427def2563f6d0 |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType | 42f42342648b8fa7 |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.1 | de1b42a4850c814e |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.2 | 2062c5f9436afbd9 |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.3 | 9f54ef11aadacf72 |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.4 | 994767655151955f |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.5 | 1745bb6a04e8993d |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.6 | e1f9c9a005abae22 |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.7 | 6ed2f2d151367cbb |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.8 | a27ee00384a36bef |
| net.bytebuddy.description.annotation.AnnotationValue.ForConstant.PropertyDelegate.ForNonArrayType.9 | 63c6c63b84353202 |
| net.bytebuddy.description.annotation.AnnotationValue.ForDescriptionArray | fcfee5f41864d050 |
| net.bytebuddy.description.annotation.AnnotationValue.ForEnumerationDescription | ae910611763d0f68 |
| net.bytebuddy.description.annotation.AnnotationValue.ForEnumerationDescription.Loaded | d5a7abe5386d61fe |
| net.bytebuddy.description.annotation.AnnotationValue.ForTypeDescription | 6ab0ae6789cbf5ef |
| net.bytebuddy.description.annotation.AnnotationValue.Loaded.AbstractBase | f3f7beb6650a5ac8 |
| net.bytebuddy.description.annotation.AnnotationValue.Sort | 0a66a94600ced3bc |
| net.bytebuddy.description.annotation.AnnotationValue.State | ec800f741ddcd502 |
| net.bytebuddy.description.enumeration.EnumerationDescription.AbstractBase | 99c01057a1777d0c |
| net.bytebuddy.description.enumeration.EnumerationDescription.ForLoadedEnumeration | affebde010d964e1 |
| net.bytebuddy.description.field.FieldDescription | beebcc8fb52f5092 |
| net.bytebuddy.description.field.FieldDescription.AbstractBase | 14186b080645f953 |
| net.bytebuddy.description.field.FieldDescription.ForLoadedField | 123156fd14617ba2 |
| net.bytebuddy.description.field.FieldDescription.InDefinedShape.AbstractBase | 0d8b55a89d1f5d33 |
| net.bytebuddy.description.field.FieldDescription.Latent | 3988698b9a06a78b |
| net.bytebuddy.description.field.FieldDescription.SignatureToken | 56fee817a36ae13e |
| net.bytebuddy.description.field.FieldDescription.Token | cb32bbcddd28b15f |
| net.bytebuddy.description.field.FieldList.AbstractBase | eaf7adc9c2f91154 |
| net.bytebuddy.description.field.FieldList.Explicit | 8f12c029b0e18281 |
| net.bytebuddy.description.field.FieldList.ForLoadedFields | 8b99118397373efd |

| | |
|---|-------------------|
| net.bytebuddy.description.field.FieldList.ForTokens | 212571f44e74f5ee |
| net.bytebuddy.description.method.MethodDescription | 29fc82bada408f50 |
| net.bytebuddy.description.method.MethodDescription.AbstractBase | cd48701a16ec6aae |
| net.bytebuddy.description.method.MethodDescription.ForLoadedConstructor | 6b82fc21752f5919 |
| net.bytebuddy.description.method.MethodDescription.ForLoadedMethod | 1c3e6a782b133159 |
| net.bytebuddy.description.method.MethodDescription.InDefinedShape.AbstractBase | a1150ad60ccc435a |
| net.bytebuddy.description.method.MethodDescription.InDefinedShape.AbstractBase.ForLoadedExecutable | 1009e43f35e34cbd |
| net.bytebuddy.description.method.MethodDescription.Latent | 906c68aeeaaf7c2c |
| net.bytebuddy.description.method.MethodDescription.Latent.TypeInitializer | d2d992bea0797b42 |
| net.bytebuddy.description.method.MethodDescription.SignatureToken | 5451f2ff109b0d00 |
| net.bytebuddy.description.method.MethodDescription.Token | a2181b4586337ee7 |
| net.bytebuddy.description.method.MethodDescription.TypeSubstituting | 0993d75a33526eb0 |
| net.bytebuddy.description.method.MethodDescription.TypeToken | 5e47ae2a4768bc1c |
| net.bytebuddy.description.method.MethodList.AbstractBase | 6e4bb8d5a09ca7f3 |
| net.bytebuddy.description.method.MethodList.Explicit | 9650ca53b2f6a6e2 |
| net.bytebuddy.description.method.MethodList.ForLoadedMethods | 7648140277a32974 |
| net.bytebuddy.description.method.MethodList.ForTokens | 42b106ad57e53b91 |
| net.bytebuddy.description.method.MethodList.TypeSubstituting | e3ff3658a2c85bc8 |
| net.bytebuddy.description.method.ParameterDescription.AbstractBase | f0e631571e74e319 |
| net.bytebuddy.description.method.ParameterDescription.ForLoadedParameter | 1f8303d30bd71a08 |
| net.bytebuddy.description.method.ParameterDescription.ForLoadedParameter.OfConstructor | c6cfab06736c11d7 |
| net.bytebuddy.description.method.ParameterDescription.ForLoadedParameter.OfMethod | 4bdfdb69b0fecfa2 |
| net.bytebuddy.description.method.ParameterDescription.InDefinedShape.AbstractBase | 607a2c5720c1d99a |
| net.bytebuddy.description.method.ParameterDescription.Latent | be72fd24f033a8d0 |
| net.bytebuddy.description.method.ParameterDescription.Token | 8b04ac8d91e9a55d |
| net.bytebuddy.description.method.ParameterDescription.Token.TypeList | c866bbd9b65b2971 |
| net.bytebuddy.description.method.ParameterDescription.TypeSubstituting | 0a2f4fe45f35ee10 |
| net.bytebuddy.description.method.ParameterList.AbstractBase | 713c270249b385e6 |
| net.bytebuddy.description.method.ParameterList.Empty | 8a77392f505a85b6 |
| net.bytebuddy.description.method.ParameterList.Explicit.ForTypes | e403d19044367493 |
| net.bytebuddy.description.method.ParameterList.ForLoadedExecutable | 6ddce2700deb6f43 |
| net.bytebuddy.description.method.ParameterList.ForLoadedExecutable.OfConstructor | 884be9c14d50eac9 |
| net.bytebuddy.description.method.ParameterList.ForLoadedExecutable.OfMethod | 863355ac9b305941 |
| net.bytebuddy.description.method.ParameterList.ForTokens | 9205617d32f4ed0f |
| net.bytebuddy.description.method.ParameterList.TypeSubstituting | 3ad933fe68b4ba20 |
| net.bytebuddy.description.modifier.FieldManifestation | 8c08d347304d6d18 |
| net.bytebuddy.description.modifier.ModifierContributor.Resolver | 0703516f4b7fd825 |
| net.bytebuddy.description.modifier.Ownership | 2e60a3d39ac5641b |
| net.bytebuddy.description.modifier.SynchronizationState | f5442374e6d2e05c |
| net.bytebuddy.description.modifier.SyntheticState | 32e09bf9a909c23b |
| net.bytebuddy.description.modifier.TypeManifestation | b090b1e7c7385c73 |
| net.bytebuddy.description.modifier.Visibility | 98008a87e5e30e3e |
| net.bytebuddy.description.modifier.Visibility.1 | b8d629c2c45ceb7b |
| net.bytebuddy.description.type.PackageDescription.AbstractBase | 0ec6f311394275e0 |
| net.bytebuddy.description.type.PackageDescription.ForLoadedPackage | 4c0675dabc7f0352 |
| net.bytebuddy.description.type.PackageDescription.Simple | f70620e1fa6685ee |
| net.bytebuddy.description.type.RecordComponentList.AbstractBase | 6ec7cebb7657a89e |
| net.bytebuddy.description.type.RecordComponentList.Empty | 8a33bfff3c9e86862 |
| net.bytebuddy.description.type.RecordComponentList.ForTokens | fc914dd18d7a6558 |
| net.bytebuddy.description.type.TypeDefinition.Sort | 6274c605f85caf0c |
| net.bytebuddy.description.type.TypeDefinition.SuperClassIterator | 5809749d8b426dc8 |
| net.bytebuddy.description.type.TypeDescription | c3eaeda62d963b8f |
| net.bytebuddy.description.type.TypeDescription.AbstractBase | 4c470694ace19ac8 |
| net.bytebuddy.description.type.TypeDescription.AbstractBase.OfSimpleType | 5818a940b298c7ab |
| net.bytebuddy.description.type.TypeDescription.ArrayProjection | db4200f41a9d25b8 |
| net.bytebuddy.description.type.TypeDescription.ForLoadedType | 445de55c15b3091b |
| net.bytebuddy.description.type.TypeDescription.Generic | 52e171d973daa9cb |
| net.bytebuddy.description.type.TypeDescription.Generic.AbstractBase | 70091e48533fe83a |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator | ba33a37252d9901e |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.Chained | a18769b831f5045 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.ForLoadedExecutableExceptionType | fc12460a1e315c5f |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.ForLoadedExecutableParameterType | 820fde02a9920d37 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.ForLoadedField | c983cfe01d0d5edb |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.ForLoadedInterface | a5b8cb5a2d7c21c3 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.ForLoadedMethodReturnType | 850a31c9319cfa02 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.ForLoadedSuperClass | c2f372c822fcfa3e8 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.ForLoadedTypeVariable | 3925b4cc15f4bd2f |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.Delegator.Simple | 242d40ab18115093 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.ForComponentType | 4748d1b5aeab5236 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.ForOwnerType | a2f75393f0f99bb4 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.ForTypeArgument | b6147d2e51640023 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.ForTypeVariableBoundType | fdfe10d1fb135210 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.ForTypeVariableBoundType.OfFormalTypeVariable | bb1fab9736bbbedf3 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.ForWildcardLowerBoundType | b97a1df71c1b4dc5 |
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.ForWildcardUpperBoundType | 98fbf4d7a37c0e61 |

| | |
|--|-------------------|
| net.bytebuddy.description.type.TypeDescription.Generic.AnnotationReader.NoOp | bfc9d9d8dca3065f5 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection | 07ac00385cc953c1 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.ForLoadedFieldType | 1b18d66c9610018d |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.ForLoadedReturnType | 2bf5be358b4dc655 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.ForLoadedSuperClass | b3ce70f79a11e22c |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.OfConstructorParameter | dc5b41f427a54066 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.OfMethodParameter | fb1afa36f177f218 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.WithEagerNavigation | 2c30af0f6fa58f88 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.WithEagerNavigation.OfAnnotatedElement | 6806a9bf7ba25171 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.WithLazyNavigation | 80a39599d01968d6 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.WithLazyNavigation.OfAnnotatedElement | d2786c8c9ae86232 |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProjection.WithResolvedErasure | 306963d1fc7a671c |
| net.bytebuddy.description.type.TypeDescription.Generic.LazyProxy | c5562144abf34544 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfGenericArray | 0660b19e071b04d9 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfGenericArray.ForLoadedType | 4a2923389eaef5c8b |
| net.bytebuddy.description.type.TypeDescription.Generic.OfGenericArray.Latent | 0980141661f7e1a7 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfNonGenericType | 9f6b0bd613a646fc |
| net.bytebuddy.description.type.TypeDescription.Generic.OfNonGenericType.ForErasure | 8c4eacb30cea265e |
| net.bytebuddy.description.type.TypeDescription.Generic.OfNonGenericType.ForLoadedType | 5274fb8c35225f5f |
| net.bytebuddy.description.type.TypeDescription.Generic.OfNonGenericType.ForReifiedErasure | aed938d46e95e44c |
| net.bytebuddy.description.type.TypeDescription.Generic.OfNonGenericType.Latent | 372c896f35267a96 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfParameterizedType | cf9990642c03405f |
| net.bytebuddy.description.type.TypeDescription.Generic.OfParameterizedType.ForGenerifiedErasure | 7ed56b64be3ef06d |
| net.bytebuddy.description.type.TypeDescription.Generic.OfParameterizedType.ForLoadedType | 554831d385360a6d |
| net.bytebuddy.description.type.TypeDescription.Generic.OfParameterizedType.ForLoadedType.ParameterArgumentTypeList | 4d7b4f0a54d46909 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfParameterizedType.ForReifiedType | 320948b9f385d4b4 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfParameterizedType.Latent | 65bad8524d24da78 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfTypeVariable | 86e4078bd5cd6a82 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfTypeVariable.ForLoadedType | ed5cd3cbbe779d8c |
| net.bytebuddy.description.type.TypeDescription.Generic.OfTypeVariable.ForLoadedType.TypeVariableBoundList | eb6620f5799fe8f5 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfTypeVariable.Symbolic | 58cf78bca4d2bd51 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfTypeVariable.WithAnnotationOverlay | ec478e7fa8287853 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfWildcardType | 9f36a440f0a04724 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfWildcardType.ForLoadedType | 2224f84636e0f952 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfWildcardType.ForLoadedType.WildcardLowerBoundTypeList | 615a97949ca84b45 |
| net.bytebuddy.description.type.TypeDescription.Generic.OfWildcardType.ForLoadedType.WildcardUpperBoundTypeList | 87a5af4372397ddaa |
| net.bytebuddy.description.type.TypeDescription.Generic.OfWildcardType.Latent | 0f49202883abd15b |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.AnnotationStripper | 3e19b0550a8be1bd |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.AnnotationStripper.NonAnnotatedTypeVariable | 617699ec0a3562b8 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.ForRawType | 2ab665659e488ee3 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.ForSignatureVisitor | c0eebf71c2c20203 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.ForSignatureVisitorOfTypeArgument | 0561fc5d35509089 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Reducing | 87d9b9c7cdb46685 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Reifying | 1f23e61efd8c040e |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Reifying.1 | 9992e854290cac45 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Reifying.2 | 3880aef22a6c4930 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Substitutor | 5bc3866175f926f2 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Substitutor.ForAttachment | 8cf9a45e6526538b |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Substitutor.ForDetachment | c1c177d019118d9b |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Substitutor.ForTypeVariableBinding | c8c364768506ed0b |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Substitutor.ForTypeVariableBinding.RetainedMethodTypeVariable | 96bac28c872b4281 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Substitutor.ForTypeVariableBinding.TypeVariableSubstitutor | cce8953b87ceb945 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Substitutor.WithoutTypeSubstitution | 039af874f4087139 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.TypeErasing | c81cedd98f160974 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Validator | 140b09b73b4b60c8 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Validator.1 | d799308b1611cd13 |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Validator.2 | 499de672436311ce |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.Validator.3 | 7923429fa88ea5cf |
| net.bytebuddy.description.type.TypeDescription.Generic.Visitor.ValidatorForTypeAnnotations | 6025d4a2423e87ac |
| net.bytebuddy.description.type.TypeDescription.Latent | aff7537b8e4a3b56 |
| net.bytebuddy.description.type.TypeDescription.LazyProxy | edc5abaabf7281a4 |
| net.bytebuddy.description.type.TypeList | ffa2fe3b5b233353 |
| net.bytebuddy.description.type.TypeList.AbstractBase | 03ace8fa7043bfce |
| net.bytebuddy.description.type.TypeList.Empty | d5bf5d4843d3b40b |
| net.bytebuddy.description.type.TypeList.Explicit | 23780d287a231e41 |
| net.bytebuddy.description.type.TypeList.ForLoadedTypes | 826dc7ad585dcbbc |
| net.bytebuddy.description.type.TypeList.Generic.AbstractBase | 82e09b4ce109a939 |
| net.bytebuddy.description.type.TypeList.Generic.Empty | 09b9aca5ee8c1475 |
| net.bytebuddy.description.type.TypeList.Generic.Explicit | d184b9433829cab |
| net.bytebuddy.description.type.TypeList.Generic.ForDetachedTypes | 7df55a5ca5fec13f |
| net.bytebuddy.description.type.TypeList.Generic.ForDetachedTypes.OfTypeVariables | 3a26e31857238724 |
| net.bytebuddy.description.type.TypeList.Generic.ForDetachedTypes.OfTypeVariables.AttachedTypeVariable | 75cc8314964e08b5 |
| net.bytebuddy.description.type.TypeList.Generic.ForDetachedTypes.WithResolvedErasure | 5dc3a74323127305 |
| net.bytebuddy.description.type.TypeList.Generic.ForLoadedTypes | 6d36b199d5e88b53 |

| | |
|---|-------------------|
| net.bytebuddy.description.type.TypeList.Generic.ForLoadedTypes.OfTypeVariables | 0ed9eb434cab9b70 |
| net.bytebuddy.description.type.TypeList.Generic.OfConstructorExceptionTypes | 1eb0243fd9192b8a |
| net.bytebuddy.description.type.TypeList.Generic.OfConstructorExceptionTypes.TypeProjection | e3069579f0c36ea7 |
| net.bytebuddy.description.type.TypeList.Generic.OfLoadedInterfaceTypes | 1d9424dd0d7bc1c0 |
| net.bytebuddy.description.type.TypeList.Generic.OfLoadedInterfaceTypes.TypeProjection | 808c89479cd00a5d |
| net.bytebuddy.description.type.TypeList.Generic.OfMethodExceptionTypes | f6b71c9483cf7675 |
| net.bytebuddy.description.type.TypeList.Generic.OfMethodExceptionTypes.TypeProjection | 30acd7d949c355bb |
| net.bytebuddy.description.type.TypeVariableToken | ba16c063046983a1 |
| net.bytebuddy.dynamic.ClassFileLocator.ForClassLoader | 27d77cb6b78fc547 |
| net.bytebuddy.dynamic.ClassFileLocator.ForClassLoader.BootLoaderProxyCreationAction | cd2788dbef627a42 |
| net.bytebuddy.dynamic.ClassFileLocator.Resolution.Explicit | 3b5906ac6a8e97c |
| net.bytebuddy.dynamic.ClassFileLocator.Simple | aa55299a7027b445 |
| net.bytebuddy.dynamic.DynamicType.AbstractBase | 690dc473a1fcc899 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase | aa1759a3194d7f14 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter | 1052045033e158fb |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter.FieldDefinitionAdapter | b0d316958f2c533a |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter.MethodDefinitionAdapter | 060b669748c6eb43 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter.MethodDefinitionAdapter.AnnotationAdapter | 734373a7d618763a |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter.MethodDefinitionAdapter.SimpleParameterAnnotationAdapter | 0e661f268cd9c08 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter.MethodMatchAdapter | 7a8cf3bcd2d805c6 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter.MethodMatchAdapter.AnnotationAdapter | 1505339515cd1c19 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Adapter.OptionalMethodMatchAdapter | d415d86eeb500e07 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.Delegator | 936b4705c2af1791 |
| net.bytebuddy.dynamic.DynamicType.Builder.AbstractBase.UsingTypeWriter | fd88ea62d3bec9e8 |
| net.bytebuddy.dynamic.DynamicType.Builder.FieldDefinition.Optional.AbstractBase | 305861cdfa6660be |
| net.bytebuddy.dynamic.DynamicType.Builder.FieldDefinition.Optional.Valuable.AbstractBase | 6ac9b330fc124370 |
| net.bytebuddy.dynamic.DynamicType.Builder.FieldDefinition.Optional.Valuable.AbstractBase.Adapter | 61c7c08f0fb4acb0 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.AbstractBase | e1b17723ee466981 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.AbstractBase.Adapter | fa398faff0645c86 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ExceptionDefinition.AbstractBase | 553558ccdd2ea7d30 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ImplementationDefinition.AbstractBase | 72bbf9be43329300 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ParameterDefinition.AbstractBase | c46ba31879e7f18 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ParameterDefinition.Initial.AbstractBase | bf9ad000af8dfbe7 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ParameterDefinition.Simple.AbstractBase | 3591bd96cf8ba88c |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ParameterDefinition.Simple.Annotatable.AbstractBase | dd4ec10b2ca5aed6 |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ParameterDefinition.Simple.Annotatable.AbstractBase.Adapter | 511cf7967b5892fe |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.ReceiverTypeDefinition.AbstractBase | 6abfb4279099921b |
| net.bytebuddy.dynamic.DynamicType.Builder.MethodDefinition.TypeVariableDefinition.AbstractBase | 9bb9648ea001ee26 |
| net.bytebuddy.dynamic.DynamicType.Default | f56db2a964930d9f |
| net.bytebuddy.dynamic.DynamicType.Default.Loaded | b3ca6c151890771d |
| net.bytebuddy.dynamic.DynamicType.Default.Unloaded | 60aab9827a2fe060 |
| net.bytebuddy.dynamic.TargetType | 92d2ae273e27a92e |
| net.bytebuddy.dynamic.Transformer.Compound | 31031064056b461f |
| net.bytebuddy.dynamic.Transformer.ForMethod | 9763e7fc8d3d0039 |
| net.bytebuddy.dynamic.Transformer.ForMethod.MethodModifierTransformer | 392d6ab6d2a08504 |
| net.bytebuddy.dynamic.Transformer.ForMethod.TransformedMethod | 8ee8fb29bba1f45 |
| net.bytebuddy.dynamic.Transformer.ForMethod.TransformedMethod.AttachmentVisitor | 41498d5684407e50 |
| net.bytebuddy.dynamic.Transformer.ForMethod.TransformedMethod.TransformedParameter | 2196a64af1ada8f5 |
| net.bytebuddy.dynamic.Transformer.ForMethod.TransformedMethod.TransformedParameterList | 68df6a158c64de3a |
| net.bytebuddy.dynamic.Transformer.NoOp | 8fa58b41007a0c6e |
| net.bytebuddy.dynamic.TypeResolutionStrategy.Passive | 560bec77a3a9e1cd |
| net.bytebuddy.dynamic.VisibilityBridgeStrategy.Default | 92dd499e424995ab |
| net.bytebuddy.dynamic.VisibilityBridgeStrategy.Default.1 | 88969522be51c47a |
| net.bytebuddy.dynamic.VisibilityBridgeStrategy.Default.2 | e90257accee75dbe |
| net.bytebuddy.dynamic.VisibilityBridgeStrategy.Default.3 | dbaleec57628b9b8 |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader | bc61ad56649f6a21 |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.ClassDefinitionAction | 7ddd05378a81b5ac |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.PackageLookupStrategy.CreationAction | 579d717b5fb1777c |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.PackageLookupStrategy.ForJava9CapableVm | 3116856af2d9e391 |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.PersistenceHandler | e4d3975d11965972 |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.PersistenceHandler.1 | c5bf4188a059f348 |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.PersistenceHandler.2 | d4d9d89c20c33c79 |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.SynchronizationStrategy.CreationAction | 85defb9db5b2b5cc |
| net.bytebuddy.dynamic.loading.ByteArrayClassLoader.SynchronizationStrategy.ForJava8CapableVm | fae027d8e16a32fa |
| net.bytebuddy.dynamic.loading.ClassFilePostProcessor.NoOp | 6146ad1c41c815ee |
| net.bytebuddy.dynamic.loading.ClassInjector.AbstractBase | 48ea808bb6ccbbf1 |
| net.bytebuddy.dynamic.loading.ClassInjector.UsingLookup | 0ac0b66e8bfc7f1d |
| net.bytebuddy.dynamic.loading.ClassLoadingStrategy | eee19f05d61ced57 |
| net.bytebuddy.dynamic.loading.ClassLoadingStrategy.UsingLookup | 3fe445da3a017ea7 |
| net.bytebuddy.dynamic.loading.InjectionClassLoader | 41bc10f352e48896 |
| net.bytebuddy.dynamic.loading.InjectionClassLoader.Strategy | 6ed217ae5fb57d05 |
| net.bytebuddy.dynamic.loading.MultipleParentClassLoader.Builder | 079ab85cc0317f7f |
| net.bytebuddy.dynamic.loading.PackageDefinitionStrategy.Definition.Trivial | 968c1d05fe34e6ba |
| net.bytebuddy.dynamic.loading.PackageDefinitionStrategy.Trivial | 1b8be35a135f3f27 |

| | |
|---|------------------|
| net.bytebuddy.dynamic.scaffold.FieldLocator.AbstractBase | cf87e1b20bb0b8da |
| net.bytebuddy.dynamic.scaffold.FieldLocator.ForClassHierarchy | 7a3b903f068f7597 |
| net.bytebuddy.dynamic.scaffold.FieldLocator.ForClassHierarchy.Factory | 03b1190c054626d7 |
| net.bytebuddy.dynamic.scaffold.FieldLocator.Resolution.Illegal | 2aa162cec2d09e4e |
| net.bytebuddy.dynamic.scaffold.FieldLocator.Resolution.Simple | 9c60c3ef37a8e07c |
| net.bytebuddy.dynamic.scaffold.FieldRegistry.Default | d252721578bc0fb5 |
| net.bytebuddy.dynamic.scaffold.FieldRegistry.Default.Compiled | 7dac3c03ab5df0af |
| net.bytebuddy.dynamic.scaffold.FieldRegistry.Default.Compiled.Entry | a356ed792e610abf |
| net.bytebuddy.dynamic.scaffold.FieldRegistry.Default.Entry | 21cbb71bbb74244a |
| net.bytebuddy.dynamic.scaffold.InstrumentedType.Default | e9bbd49fcda040b4 |
| net.bytebuddy.dynamic.scaffold.InstrumentedType.Factory.Default | c7f976e803f8d3a8 |
| net.bytebuddy.dynamic.scaffold.InstrumentedType.Factory.Default.1 | 01e6e0df84609179 |
| net.bytebuddy.dynamic.scaffold.InstrumentedType.Factory.Default.2 | 41b308deac98792e |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler | 6b8d18710c99d74d |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.AbstractBase | 86739f96369d83b2 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default | 2800dfa5c3b8aca7 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Harmonizer.ForJavaMethod | 8ef28acf242eae2d |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Harmonizer.ForJavaMethod.Token | 7bd97f7cecaf5018 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key | 88f2458db8c741e7 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Detached | 89750a8204bacfc3 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Harmonized | a31e139e57804f56 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Store | 7716ed339d21e000 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Store.Entry.Ambiguous | 69db800671ee2019 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Store.Entry.Initial | a1eb87bc66f88929 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Store.Entry.Resolved | 9c0eb28c1120302a |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Store.Resolved.Node | 7bbc1aea5064d7 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Key.Store.Graph | bf364158f6188e45 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.Default.Merger.Direction | afc0868aacc8dbb5 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Compiler.ForDeclaredMethods | 6c4c0c83346af9e0 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Empty | 0f8190524ba952c8 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Linked.Delegation | 47e48556a9cda077 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Node.Simple | cf6e675432bc60ba |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Node.Sort | c997ba920aa81bb5 |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Node.Unresolved | c0dc9372315bec1a |
| net.bytebuddy.dynamic.scaffold.MethodGraph.NodeList | ddb63425295dab8e |
| net.bytebuddy.dynamic.scaffold.MethodGraph.Simple | b231d730a2138443 |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Default | e823178c1c962613 |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Default.Compiled | 86804b8f21a02f7e |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Default.Compiled.Entry | b9727d766a3abd06 |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Default.Entry | 0216bf1a4989ddd9 |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Default.Prepared | 2fc66a86a52ec6f1 |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Default.Prepared.Entry | 6c98eecdd8c8c68 |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Handler.ForImplementation | 8db040195f76894f |
| net.bytebuddy.dynamic.scaffold.MethodRegistry.Handler.ForImplementation.Compiled | 3ade648194f6daec |
| net.bytebuddy.dynamic.scaffold.RecordComponentRegistry.Default | 891a9cb069e1ffb3 |
| net.bytebuddy.dynamic.scaffold.RecordComponentRegistry.Default.Compiled | faf0312e50511a4b |
| net.bytebuddy.dynamic.scaffold.TypeInitializer.Drain.Default | 391ec96470191ff5 |
| net.bytebuddy.dynamic.scaffold.TypeInitializer.None | 0aa5903ad3f1d635 |
| net.bytebuddy.dynamic.scaffold.TypeInitializer.Simple | cf379da7dede87a5 |
| net.bytebuddy.dynamic.scaffold.TypeValidation | 50527b71bc87dd74 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default | f27ba1edbba53a3c |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ClassDumpAction.Dispatcher.Disabled | 2cbe19f435bea571 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForCreation | 03b740be560ec44d |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining | ea80afd67a6759b1 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.ContextRegistry | 30f87ac3b6e91d07 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.WithFullProcessing | cc2d500ff979fe0d |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.WithFullProcessing.InitializationHandler.Appending | 78681a98b1676574 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.WithFullProcessing.InitializationHandler.Appending.FrameWriter.NoOp | 4e676b84c2db9221 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.WithFullProcessing.InitializationHandler.Appending.WithoutDrain | 1c35fc86761d78a |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.WithFullProcessing.InitializationHandler.Appending.WithoutDrain.WithoutActiveRecord | a9601fe4f1c04d81 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.WithFullProcessing.InitializationHandler.Creating | 1863bc3b7943bb2b |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ForInlining.WithFullProcessing.RedefinitionClassVisitor | 88eb70c1f7cb268c |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.SignatureKey | 30f02c677c168ffb |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.UnresolvedType | 9b9d5a9877b725fe |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ValidatingClassVisitor | fef9b21649f006c2 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ValidatingClassVisitor.Constraint.Compound | 7658cf279fc7eba |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ValidatingClassVisitor.Constraint.ForClass | d6e8080c2ac49ca3 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ValidatingClassVisitor.Constraint.ForClassFileVersion | 98d826d97325d335 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.Default.ValidatingClassVisitor.ValidatingMethodVisitor | 0f3d60cd71ee55ec |
| net.bytebuddy.dynamic.scaffold.TypeWriter.FieldPool.Record.ForExplicitField | 84d49e949954f821 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.FieldPool.Record.ForImplicitField | bd4c79016e6c7e59 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.MethodPool.Record.AccessBridgeWrapper | 51ec572f133baaf7 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.MethodPool.Record.ForDefinedMethod | f6b5f34a9b4e8f93 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.MethodPool.Record.ForDefinedMethod.WithBody | e26a3dcae9eda3d3 |
| net.bytebuddy.dynamic.scaffold.TypeWriter.MethodPool.Record.ForNonImplementedMethod | 0ebfdc52a422a2d4 |

| | |
|---|-------------------|
| net.bytebuddy.dynamic.scaffold.TypeWriter.MethodPool.Record.Sort | 9c1fc9a17d6e668e |
| net.bytebuddy.dynamic.scaffold.inline.AbstractInliningDynamicTypeBuilder | cf872dafa4e005ce |
| net.bytebuddy.dynamic.scaffold.inline.InliningImplementationMatcher | cd56190fcf8cbef5 |
| net.bytebuddy.dynamic.scaffold.inline.MethodRebaseResolver.Disabled | 7fc8122bae557f2d |
| net.bytebuddy.dynamic.scaffold.inline.RedefinitionDynamicTypeBuilder | d35704c254c6bf65 |
| net.bytebuddy.dynamic.scaffold.subclass.ConstructorStrategy.Default | 30f93c3d785de262 |
| net.bytebuddy.dynamic.scaffold.subclass.ConstructorStrategy.Default.1 | c55a41a20be13da8 |
| net.bytebuddy.dynamic.scaffold.subclass.ConstructorStrategy.Default.2 | 1b59827354f09dbe |
| net.bytebuddy.dynamic.scaffold.subclass.ConstructorStrategy.Default.3 | 03f49005d185f31e |
| net.bytebuddy.dynamic.scaffold.subclass.ConstructorStrategy.Default.4 | 8abb20a81cdfe753 |
| net.bytebuddy.dynamic.scaffold.subclass.ConstructorStrategy.Default.5 | 03da9e9a4baaa343b |
| net.bytebuddy.dynamic.scaffold.subclass.SubclassDynamicTypeBuilder | ce4febfb2d4e7172b |
| net.bytebuddy.dynamic.scaffold.subclass.SubclassDynamicTypeBuilder.InstrumentableMatcher | dd98202dc516302 |
| net.bytebuddy.dynamic.scaffold.subclass.SubclassImplementationTarget | 5c39954e80bc68aa |
| net.bytebuddy.dynamic.scaffold.subclass.SubclassImplementationTarget.Factory | 9507db5376136dc4 |
| net.bytebuddy.dynamic.scaffold.subclass.SubclassImplementationTarget.OriginTypeResolver | e3591b410f1beb48 |
| net.bytebuddy.dynamic.scaffold.subclass.SubclassImplementationTarget.OriginTypeResolver.1 | 307c2f6381acad86 |
| net.bytebuddy.dynamic.scaffold.subclass.SubclassImplementationTarget.OriginTypeResolver.2 | d07ed98e74e119c8 |
| net.bytebuddy.implementation.FieldAccessor | cfb202bdbbb8245 |
| net.bytebuddy.implementation.FieldAccessor.FieldLocation.Relative | 984a923cd14afdb5 |
| net.bytebuddy.implementation.FieldAccessor.FieldLocation.Relative.Prepared | 6f3f6e4a9370fc37 |
| net.bytebuddy.implementation.FieldNameExtractor.ForBeanProperty | 6fc2126f6b4215fc |
| net.bytebuddy.implementation.FieldNameExtractor.ForBeanProperty.1 | 0e09b0a77c7d7add |
| net.bytebuddy.implementation.FieldNameExtractor.ForBeanProperty.2 | 9b5ceca2df3ecd1d |
| net.bytebuddy.implementation.FieldAccessor.ForImplicitProperty | 00ae8b8a36194ac3 |
| net.bytebuddy.implementation.FieldAccessor.ForImplicitProperty.Appender | 2e8d7d416fa1d766 |
| net.bytebuddy.implementation.Implementation.Context.Default | a4f77d0e8c30d8ba |
| net.bytebuddy.implementation.Implementation.Context.Default.AbstractPropertyAccessorMethod | e66307066ee6748f |
| net.bytebuddy.implementation.Implementation.Context.Default.AccessorMethod | 2cbdafd9341571b2 |
| net.bytebuddy.implementation.Implementation.Context.Default.AccessorMethodDelegation | e8f1a8118ac7bb70 |
| net.bytebuddy.implementation.Implementation.Context.Default.CacheValueField | a8f1451ccb79fe81 |
| net.bytebuddy.implementation.Implementation.Context.Default.DelegationRecord | ffeb02bcef268ff1 |
| net.bytebuddy.implementation.Implementation.Context.Default.Factory | 7907cd1700b68712 |
| net.bytebuddy.implementation.Implementation.Context.Default.FieldCacheEntry | 32d2586f8f82b67b |
| net.bytebuddy.implementation.Implementation.Context.Disabled | d226525d9b1661c1 |
| net.bytebuddy.implementation.Implementation.Context.Disabled.Factory | fc5147fce792870c |
| net.bytebuddy.implementation.Implementation.Context.ExtractableView.AbstractBase | fc1194c64d81a8f1 |
| net.bytebuddy.implementation.Implementation.Context.FrameGeneration | 732fc9a6ba6d7f948 |
| net.bytebuddy.implementation.Implementation.Context.FrameGeneration.1 | e1cb50e88c828853 |
| net.bytebuddy.implementation.Implementation.Context.FrameGeneration.2 | eced4d043d2746ae |
| net.bytebuddy.implementation.Implementation.Context.FrameGeneration.3 | 896ff9ab8ad2a703 |
| net.bytebuddy.implementation.Implementation.SpecialMethodInvocation.AbstractBase | c1e6064a9dc7eabe |
| net.bytebuddy.implementation.Implementation.SpecialMethodInvocation.Illegal | ca8b00de46ba4516 |
| net.bytebuddy.implementation.Implementation.SpecialMethodInvocation.Simple | 3b1d460fdb691665 |
| net.bytebuddy.implementation.Implementation.Target.AbstractBase | 2bb25225337712f2 |
| net.bytebuddy.implementation.Implementation.Target.AbstractBase.DefaultMethodInvocation | 95cadf8bb1c40a5e |
| net.bytebuddy.implementation.Implementation.Target.AbstractBase.DefaultMethodInvocation.1 | 1522748b6ede90f9 |
| net.bytebuddy.implementation.Implementation.Target.AbstractBase.DefaultMethodInvocation.2 | 1a1e99939835b649 |
| net.bytebuddy.implementation.LoadedTypeInitializer.NoOp | 079c0db350266bf1 |
| net.bytebuddy.implementation.MethodAccessorFactory.AccessType | e6d2a93e93eddb74 |
| net.bytebuddy.implementation.MethodCall | 803a80b61cf5154e |
| net.bytebuddy.implementation.MethodCall.Appender | 2c97aa6a2fcc90d6 |
| net.bytebuddy.implementation.MethodCall.ArgumentLoader.ForMethodParameter | d998bab2fab7f80 |
| net.bytebuddy.implementation.MethodCall.ArgumentLoader.ForMethodParameter.Factory | c6d5aecc90f60b77 |
| net.bytebuddy.implementation.MethodCall.MethodInvoker.ForContextualInvocation | c62220dd02fbbe5e |
| net.bytebuddy.implementation.MethodCall.MethodInvoker.ForContextualInvocation.Factory | 8e991c8f6c61a26b |
| net.bytebuddy.implementation.MethodCall.MethodInvoker.ForVirtualInvocation.WithImplicitType | 4b9ee3e1372d7a3c |
| net.bytebuddy.implementation.MethodCall.MethodLocator.ForExplicitMethod | 3f52678e927adb2e |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForMethodCall | 168bccd0171653ed |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForMethodCall.Factory | 007987dfc0108802 |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForMethodCall.Resolved | 98d454c731a52565 |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForMethodParameter | e14f2f3fbceefcc9 |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForMethodParameter.Resolved | 2905935a56e53005 |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForSelfOrStaticInvocation | 9ae78c658f9fa129 |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForSelfOrStaticInvocation.Factory | f95bda15a3f3fc2d |
| net.bytebuddy.implementation.MethodCall.TargetHandler.ForSelfOrStaticInvocation.Resolved | f56743558df08d26 |
| net.bytebuddy.implementation.MethodCall.TerminationHandler.Simple | 94548bcde51ac7c5 |
| net.bytebuddy.implementation.MethodCall.TerminationHandler.Simple.1 | 51a0379b030a2561 |
| net.bytebuddy.implementation.MethodCall.TerminationHandler.Simple.2 | 0f85a678e3b7ce29 |
| net.bytebuddy.implementation.MethodCall.TerminationHandler.Simple.3 | 75c93aba20f438fe |
| net.bytebuddy.implementation.MethodCall.WithoutSpecifiedTarget | de94872b451b3f74 |
| net.bytebuddy.implementation.MethodDelegation | a4a99e4dcf919d89 |
| net.bytebuddy.implementation.MethodDelegation.Appender | dc07d3d822bdcefd |
| net.bytebuddy.implementation.MethodDelegation.ImplementationDelegate.Compiled.ForStaticCall | 4e320221c5c177b6 |

| | |
|--|-------------------|
| net.bytebuddy.implementation.MethodDelegation.ImplementationDelegate.ForStaticMethod | dd54ea94cb4466d5 |
| net.bytebuddy.implementation.MethodDelegation.WithCustomProperties | 5f090acf995bc833 |
| net.bytebuddy.implementation.SuperMethodCall | 611355edd9b41de2 |
| net.bytebuddy.implementation.SuperMethodCall.Appender | bc8a17175af42bf |
| net.bytebuddy.implementation.SuperMethodCall.Appender.TerminationHandler | 32d5e0e8182a3358 |
| net.bytebuddy.implementation.SuperMethodCall.Appender.TerminationHandler.1 | 0262af83a3a8b2c1 |
| net.bytebuddy.implementation.SuperMethodCall.Appender.TerminationHandler.2 | ee7147093f933eea |
| net.bytebuddy.implementation.attribute.AnnotationAppender.Default | 5e5b5a601807bb0c |
| net.bytebuddy.implementation.attribute.AnnotationAppender.ForTypeAnnotations | 9c69d7ba808e83bf |
| net.bytebuddy.implementation.attribute.AnnotationAppender.Target.OnField | edb1e187f577e773 |
| net.bytebuddy.implementation.attribute.AnnotationAppender.Target.OnMethod | 696208e270a3f019 |
| net.bytebuddy.implementation.attribute.AnnotationAppender.Target.OnMethodParameter | f9f5e949cd061dd0 |
| net.bytebuddy.implementation.attribute.AnnotationAppender.Target.OnType | 6ab7dc033ee53862 |
| net.bytebuddy.implementation.attribute.AnnotationRetention | da4f57f00f2339fe |
| net.bytebuddy.implementation.attribute.AnnotationValueFilter.Default | c3f974ecaffb54e9 |
| net.bytebuddy.implementation.attribute.AnnotationValueFilter.Default.1 | 16794e96c48a9eb1 |
| net.bytebuddy.implementation.attribute.AnnotationValueFilter.Default.2 | e61de6f9507d0593 |
| net.bytebuddy.implementation.attribute.FieldAttributeAppender.ForInstrumentedField | c640ee703ef26e72 |
| net.bytebuddy.implementation.attribute.MethodAttributeAppender.Compound | dac3bd643b572705 |
| net.bytebuddy.implementation.attribute.MethodAttributeAppender.Factory.Compound | 76745fca04070e30 |
| net.bytebuddy.implementation.attribute.MethodAttributeAppender.ForInstrumentedMethod | a824f27ba759e23b |
| net.bytebuddy.implementation.attribute.MethodAttributeAppender.ForInstrumentedMethod.1 | 496e9baf72f72b4d |
| net.bytebuddy.implementation.attribute.MethodAttributeAppender.ForInstrumentedMethod.2 | 96f514167b7d0a48 |
| net.bytebuddy.implementation.attribute.MethodAttributeAppender.NoOp | b52b89cf16c54ff9 |
| net.bytebuddy.implementation.attribute.TypeAttributeAppender.ForInstrumentedType | 174b3e561dfe2a50 |
| net.bytebuddy.implementation.attribute.TypeAttributeAppender.ForInstrumentedType.Differentiating | 87928a430c8985ff |
| net.bytebuddy.implementation.auxiliary.AuxiliaryType | 6a52cb65ebc696ad |
| net.bytebuddy.implementation.auxiliary.AuxiliaryType.NamingStrategy.SuffixingRandom | a173a333a763c063 |
| net.bytebuddy.implementation.auxiliary.MethodCallProxy | 83f89e06c97f0237 |
| net.bytebuddy.implementation.auxiliary.MethodCallProxy.AssignableSignatureCall | c40a6df584d6950a |
| net.bytebuddy.implementation.auxiliary.MethodCallProxy.ConstructorCall | 5b517e0b210f1158 |
| net.bytebuddy.implementation.auxiliary.MethodCallProxy.ConstructorCall.Appender | 5bd8ad0594ac3af1 |
| net.bytebuddy.implementation.auxiliary.MethodCallProxy.MethodCall | 5e100136005916ca |
| net.bytebuddy.implementation.auxiliary.MethodCallProxy.MethodCall.Appender | ce7052391779cdf9 |
| net.bytebuddy.implementation.auxiliary.MethodCallProxy.PrecomputedMethodGraph | 27a53ff891975b35 |
| net.bytebuddy.implementation.bind.ArgumentTypeResolver | 11f8505cde4d19ff |
| net.bytebuddy.implementation.bind.ArgumentTypeResolver.ParameterIndexToken | 30ba25a897cfab59 |
| net.bytebuddy.implementation.bind.DeclaringTypeResolver | dc04124901f1b333 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.1 | e8e903661a7e1247 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.AmbiguityResolver | 36f5bfa2a412ee72 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.AmbiguityResolver.Compound | 4cf7d1fa48e6cd60 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.AmbiguityResolver.Resolution | 7d702b0ecc9c891b |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.BindingResolver.Default | c33dc6b307529852 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.MethodBinding.Builder | f68e89f0719ade2e |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.MethodBinding.Builder.Build | 33f84e8df9c49416 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.MethodBinding.Illegal | 4802915ba226f6c2 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.MethodInvoker.Simple | 8afbcd842db3e720 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.ParameterBinding.Anonymous | a962db035644cb83 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.ParameterBinding.Illegal | eacaa52a8e4dde34 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.ParameterBinding.Unique | dd8a495565da1d64 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.Processor | ab3647d9a0455d24 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.TerminationHandler.Default | c7d75b957a2f27ed |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.TerminationHandler.Default.1 | 4ec0d44d8c3eceb0 |
| net.bytebuddy.implementation.bind.MethodDelegationBinder.TerminationHandler.Default.2 | 1ad724fd05690ddc |
| net.bytebuddy.implementation.bind.MethodNameEqualityResolver | 0c639151a5005f3c |
| net.bytebuddy.implementation.bind.ParameterLengthResolver | f4653916b8323a8e |
| net.bytebuddy.implementation.bind.annotation.AllArguments.Assignment | 5ff910d53a6f2d0f |
| net.bytebuddy.implementation.bind.annotation.AllArguments.Binder | 758af7bb09f652cd |
| net.bytebuddy.implementation.bind.annotation.Argument.Binder | dc92b4735795f877 |
| net.bytebuddy.implementation.bind.annotation.Argument.BindingMechanic | 99796f5e37a26565 |
| net.bytebuddy.implementation.bind.annotation.Argument.BindingMechanic.1 | 367024ef2f1388a2 |
| net.bytebuddy.implementation.bind.annotation.Argument.BindingMechanic.2 | 38ba3faff880e1c9 |
| net.bytebuddy.implementation.bind.annotation.BindingPriority.Resolver | 04a1eb2c2d2ecfac |
| net.bytebuddy.implementation.bind.annotation.Default.Binder | 3d4b610c3e192abc |
| net.bytebuddy.implementation.bind.annotation.DefaultCall.Binder | d05b63f0a6321461 |
| net.bytebuddy.implementation.bind.annotation.DefaultCallHandle.Binder | 13bdeb27076fd371 |
| net.bytebuddy.implementation.bind.annotation.DefaultMethod.Binder | e4e08f789ce159e2 |
| net.bytebuddy.implementation.bind.annotation.DefaultMethodHandle.Binder | 190a13afdf14a1a6d |
| net.bytebuddy.implementation.bind.annotation.DynamicConstant.Binder | 547b45ae9714e463 |
| net.bytebuddy.implementation.bind.annotation.Empty.Binder | d3025af78526455a |
| net.bytebuddy.implementation.bind.annotation.FieldGetterHandle.Binder | 17342fbbe9d2477 |
| net.bytebuddy.implementation.bind.annotation.FieldGetterHandle.Binder.Delegate | 92ca8f7c9f3a40e2 |
| net.bytebuddy.implementation.bind.annotation.FieldSetterHandle.Binder | e2bdded87bfab797 |
| net.bytebuddy.implementation.bind.annotation.FieldSetterHandle.Binder.Delegate | 2e2db5fe8cbd8444 |

| | |
|--|-------------------|
| net.bytebuddy.implementation.bind.annotation.FieldValue.Binder | 4f15d3f13ecce26 |
| net.bytebuddy.implementation.bind.annotation.FieldValue.Binder.Delegate | b9cd319375aeba70 |
| net.bytebuddy.implementation.bind.annotation.Handle.Binder | 390591101c305db1 |
| net.bytebuddy.implementation.bind.annotation.IgnoreForBinding.Verifier | b6bbe4a67f2ce769 |
| net.bytebuddy.implementation.bind.annotation.Origin.Binder | 63d345bc80320364 |
| net.bytebuddy.implementation.bind.annotation.RuntimeType.Verifier | 0eda7ebdfbc4de97 |
| net.bytebuddy.implementation.bind.annotation.StubValue.Binder | 0fcc840e92bf2eff |
| net.bytebuddy.implementation.bind.annotation.Super.Binder | f816bc17a41fc240 |
| net.bytebuddy.implementation.bind.annotation.SuperCall.Binder | e75122d32ab041df |
| net.bytebuddy.implementation.bind.annotation.SuperCallHandle.Binder | be3a9057258fb61c |
| net.bytebuddy.implementation.bind.annotation.SuperMethod.Binder | acdd28af315c12f1 |
| net.bytebuddy.implementation.bind.annotation.SuperMethodHandle.Binder | 7db979dc52744f1f |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder | 007d937d2ad8614a |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.DelegationProcessor | 98ee5d9e2d9a299b |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.DelegationProcessor.Handler.Bound | 22d197c23907be1d |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.DelegationProcessor.Handler.Unbound | 9d58199e821ad9fa |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.ParameterBinder | ef86cc915aa18bbf |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.ParameterBinder.ForFieldBinding | f96ce20b4cd9ae5d |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.ParameterBinder.ForFixedValue | 339866efba024bf5 |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.ParameterBinder.ForFixedValue.OfConstant | 2a9385611e3f5888 |
| net.bytebuddy.implementation.bind.annotation.TargetMethodAnnotationDrivenBinder.Record | 6ed32a75bea12174 |
| net.bytebuddy.implementation.bind.annotation.This.Binder | eaf168e8e2de8dac |
| net.bytebuddy.implementation.bytecode.ByteCodeAppender.Compound | 36fdff9633edde774 |
| net.bytebuddy.implementation.bytecode.ByteCodeAppender.Simple | e938f03cb7fc36b5 |
| net.bytebuddy.implementation.bytecode.ByteCodeAppender.Size | 6c073455b0742efa |
| net.bytebuddy.implementation.bytecode.Duplication | 6c1bd6edfcba8df8 |
| net.bytebuddy.implementation.bytecode.Duplication.1 | 5e77ff7af88e5e2b |
| net.bytebuddy.implementation.bytecode.Duplication.2 | 4fa72881088b7d9d |
| net.bytebuddy.implementation.bytecode.Duplication.3 | 22711ebeda82835f |
| net.bytebuddy.implementation.bytecode.Removal | 20e0c2619ab9e596 |
| net.bytebuddy.implementation.bytecode.Removal.1 | ec81593288755b57 |
| net.bytebuddy.implementation.bytecode.Removal.2 | 7bafc790d8ad6b0a |
| net.bytebuddy.implementation.bytecode.StackManipulation.AbstractBase | 31ada1cf9b3e1f09 |
| net.bytebuddy.implementation.bytecode.StackManipulation.Compound | 4d0ee6a3594d3abd |
| net.bytebuddy.implementation.bytecode.StackManipulation.Illegal | d208c868604ff6a8 |
| net.bytebuddy.implementation.bytecode.StackManipulation.Size | 9e6fc170da126fed |
| net.bytebuddy.implementation.bytecode.StackManipulation.Trivial | 704241e2b0e40c6e |
| net.bytebuddy.implementation.bytecode.StackSize | 4336788f1a965d2e |
| net.bytebuddy.implementation.bytecode.StackSize.1 | 96b2506c3d21bf91 |
| net.bytebuddy.implementation.bytecode.TypeCreation | 197a19a5e6570c14 |
| net.bytebuddy.implementation.bytecode.assign.Assigner | a12889e04d303449 |
| net.bytebuddy.implementation.bytecode.assign.Assigner.Typing | 5388b1bfde68c6c0 |
| net.bytebuddy.implementation.bytecode.assign.TypeCasting | 94120c4c8cf03f8 |
| net.bytebuddy.implementation.bytecode.assign.primitive.PrimitiveBoxingDelegate | 2247e86fa6ac6dbc |
| net.bytebuddy.implementation.bytecode.assign.primitive.PrimitiveBoxingDelegate.BoxingStackManipulation | b1150ae5e800b606 |
| net.bytebuddy.implementation.bytecode.assign.primitive.PrimitiveTypeAwareAssigner | cda788b56e855a02 |
| net.bytebuddy.implementation.bytecode.assign.primitive.PrimitiveUnboxingDelegate | b288a1833178dbcf |
| net.bytebuddy.implementation.bytecode.assign.primitive.PrimitiveUnboxingDelegate.ImplicitlyTypedUnboxingResponsible | fe27de58ed78c6b7 |
| net.bytebuddy.implementation.bytecode.assign.primitive.PrimitiveWideningDelegate | 15d4895de92b4326 |
| net.bytebuddy.implementation.bytecode.assign.primitive.PrimitiveWideningDelegate.WideningStackManipulation | b95f03864d242799 |
| net.bytebuddy.implementation.bytecode.assign.primitive.VoidAwareAssigner | eb758c0eaff4f960 |
| net.bytebuddy.implementation.bytecode.assign.reference.GenericTypeAwareAssigner | b45b3ebe7424172e |
| net.bytebuddy.implementation.bytecode.assign.reference.ReferenceTypeAwareAssigner | 7928b92c7844ad95 |
| net.bytebuddy.implementation.bytecode.collection.ArrayFactory | 2a2250e0a308dabf |
| net.bytebuddy.implementation.bytecode.collection.ArrayFactory.ArrayCreator | 55f358a6fd1fba94 |
| net.bytebuddy.implementation.bytecode.collection.ArrayFactory.ArrayCreator.ForReferenceType | 46c017724b6b47f9 |
| net.bytebuddy.implementation.bytecode.collection.ArrayFactory.ArrayStackManipulation | c7ca7f485aae94d7 |
| net.bytebuddy.implementation.bytecode.constant.ClassConstant | 84d3d231c511a9e2 |
| net.bytebuddy.implementation.bytecode.constant.ClassConstant.ForReferenceType | 9e6b7175c0b99ca6 |
| net.bytebuddy.implementation.bytecode.constant.DefaultValue | e6636f8b6803b575 |
| net.bytebuddy.implementation.bytecode.constant.DoubleConstant | 4605c2533c4f5ada |
| net.bytebuddy.implementation.bytecode.constant.FloatConstant | a56d418e26b00881 |
| net.bytebuddy.implementation.bytecode.constant.IntegerConstant | cc44c84f8b41799b |
| net.bytebuddy.implementation.bytecode.constant.IntegerConstant.SingleBytePush | 4dcdb3e7a7cd0de0 |
| net.bytebuddy.implementation.bytecode.constant.LongConstant | 472b65a54ff6a910 |
| net.bytebuddy.implementation.bytecode.constant.MethodConstant | 2af3bf9709ff88bc |
| net.bytebuddy.implementation.bytecode.constant.MethodConstant.CachedMethod | 120667b652f8b2ad |
| net.bytebuddy.implementation.bytecode.constant.MethodConstant.ForMethod | 6487c36db906419f |
| net.bytebuddy.implementation.bytecode.constant.NullConstant | ce7ac6225f44f48a |
| net.bytebuddy.implementation.bytecode.constant.TextConstant | 6c2a6544c010c696 |
| net.bytebuddy.implementation.bytecode.member.FieldAccess | 0c427922241a1fc2 |
| net.bytebuddy.implementation.bytecode.member.FieldAccess.AccessDispatcher | 3409f6d79dca6a70 |
| net.bytebuddy.implementation.bytecode.member.FieldAccess.AccessDispatcher.AbstractFieldInstruction | 68899123d8401287 |
| net.bytebuddy.implementation.bytecode.member.FieldAccess.AccessDispatcher.FieldGetInstruction | e9ffb22f539a4501 |
| net.bytebuddy.implementation.bytecode.member.FieldAccess.AccessDispatcher.FieldPutInstruction | 19f58898b800e6f7 |

| | |
|---|-------------------|
| net.bytebuddy.implementation.bytecode.member.MethodInvocation | f6ad313aeb1817d3 |
| net.bytebuddy.implementation.bytecode.member.MethodInvocation.Invocation | fa4fc5234c9a7c93 |
| net.bytebuddy.implementation.bytecode.member.MethodReturn | 031a5f07b7745997 |
| net.bytebuddy.implementation.bytecode.member.MethodVariableAccess | 2442be9ad3856ab6 |
| net.bytebuddy.implementation.bytecode.member.MethodVariableAccess.MethodLoading | 4c94266b0a306562 |
| net.bytebuddy.implementation.bytecode.member.MethodVariableAccess.MethodLoading.TypeCastingHandler.NoOp | fce3bb47777272e1 |
| net.bytebuddy.implementation.bytecode.member.MethodVariableAccess.OffsetLoading | c68c431573d3f1a2 |
| net.bytebuddy.implementation.bytecode.member.MethodVariableAccess.OffsetWriting | 428ad03e94bc1d66 |
| net.bytebuddy.jar.asm.AnnotationVisitor | 823759e238bb495b |
| net.bytebuddy.jar.asm.AnnotationWriter | 59e14608f0f0fc16 |
| net.bytebuddy.jar.asm.Attribute | e6480519ef45eaba |
| net.bytebuddy.jar.asm.ByteVector | 29f90958ccc2d657 |
| net.bytebuddy.jar.asm.ClassReader | 7c1a216e338347c8 |
| net.bytebuddy.jar.asm.ClassVisitor | 1753f680b0943b55 |
| net.bytebuddy.jar.asm.ClassWriter | 9aacb0d7c169551b |
| net.bytebuddy.jar.asm.ConstantDynamic | ea32b72ebf5c88ed |
| net.bytebuddy.jar.asm.Context | a881f26b77892c9d |
| net.bytebuddy.jar.asm.FieldVisitor | 2fef241e92a2948 |
| net.bytebuddy.jar.asm.FieldWriter | 7b8af1d3e89c08fe |
| net.bytebuddy.jar.asm.Handle | f00028b1416f621c |
| net.bytebuddy.jar.asm.Handler | 357ee9a4f87e5091 |
| net.bytebuddy.jar.asm.Label | c329ef00234aa4ba |
| net.bytebuddy.jar.asm.MethodVisitor | 91abf2ef44da98a7 |
| net.bytebuddy.jar.asm.MethodWriter | f98aa1935839115a |
| net.bytebuddy.jar.asm.Opcodes | 85defa2a27116c7f |
| net.bytebuddy.jar.asm.Symbol | 09ab9f266ba03e77 |
| net.bytebuddy.jar.asm.SymbolTable | a234d10951b906bb |
| net.bytebuddy.jar.asm.SymbolTable.Entry | f06a931baef45238 |
| net.bytebuddy.jar.asm.Type | 76fc57d12696f74f |
| net.bytebuddy.jar.asm.TypeReference | 0bfc56de38a7304f |
| net.bytebuddy.jar.asm.signature.SignatureVisitor | 2a359c79b449cd9d |
| net.bytebuddy.jar.asm.signature.SignatureWriter | 628941f852b053ca |
| net.bytebuddy.matcher.AnnotationTypeMatcher | 6f4a3b90208f0ec3 |
| net.bytebuddy.matcher.BooleanMatcher | 9209f695fbdc9526 |
| net.bytebuddy.matcher.CollectionErasureMatcher | e3ac7764b945369a |
| net.bytebuddy.matcher.CollectionItemMatcher | 1424fe72e0998e47 |
| net.bytebuddy.matcher.CollectionOneToOneMatcher | 121fe499dd94549c |
| net.bytebuddy.matcher.CollectionSizeMatcher | f8080735551b5869 |
| net.bytebuddy.matcher.DeclaringAnnotationMatcher | de87dd7e2883e9aa |
| net.bytebuddy.matcher.DeclaringTypeMatcher | 94491a21a3a6198c |
| net.bytebuddy.matcher.DescriptorMatcher | 8ccbfb89ae42c9c79 |
| net.bytebuddy.matcher.ElementMatcher.Junction.AbstractBase | 6e29ac5d43cf6bf5 |
| net.bytebuddy.matcher.ElementMatcher.Junction.Conjunction | 0c2d173352f518aa |
| net.bytebuddy.matcher.ElementMatcher.Junction.Disjunction | f2b0bfd4258f9323 |
| net.bytebuddy.matcher.ElementMatcher.Junction.ForNonNullValues | b442c91a882c9145 |
| net.bytebuddy.matcher.ElementMatchers | 229439fc61b9d724 |
| net.bytebuddy.matcher.EqualityMatcher | 65263674c3290275 |
| net.bytebuddy.matcher.ErasureMatcher | 5b951c67564a7bbf |
| net.bytebuddy.matcher.FailSafeMatcher | 16bb480ee9bc8e47 |
| net.bytebuddy.matcher.FilterableList.AbstractBase | cb407e29a62800ed |
| net.bytebuddy.matcher.FilterableList.Empty | 008aecb1de0a03fe |
| net.bytebuddy.matcher.LatentMatcher.Disjunction | b3238bc9859c2c6c |
| net.bytebuddy.matcher.LatentMatcher.ForFieldToken | 494f840fb29b826b |
| net.bytebuddy.matcher.LatentMatcher.ForFieldToken.ResolvedMatcher | c7f93ec3f36db445 |
| net.bytebuddy.matcher.LatentMatcher.ForMethodToken | 6a133bef32427d76 |
| net.bytebuddy.matcher.LatentMatcher.ForMethodToken.ResolvedMatcher | 4f4db49515ba05ba |
| net.bytebuddy.matcher.LatentMatcher.Resolved | 1d0baa61e9c597f8 |
| net.bytebuddy.matcher.MethodParameterTypeMatcher | ecc479943c35ad37 |
| net.bytebuddy.matcher.MethodParameterTypesMatcher | 35a1fbf9e120aea8 |
| net.bytebuddy.matcher.MethodParametersMatcher | 8ab3379e24c8d19e |
| net.bytebuddy.matcher.MethodReturnTypeMatcher | 7befd3ad928ab6cc |
| net.bytebuddy.matcher.MethodSortMatcher | 600d8d63f4cc2251 |
| net.bytebuddy.matcher.MethodSortMatcher.Sort | 87af87837374271b |
| net.bytebuddy.matcher.MethodSortMatcher.Sort.1 | 04a56e4f4f82d5c0 |
| net.bytebuddy.matcher.MethodSortMatcher.Sort.2 | a13399cf408b62c0 |
| net.bytebuddy.matcher.MethodSortMatcher.Sort.3 | 6a11f7e01098dfb5 |
| net.bytebuddy.matcher.MethodSortMatcher.Sort.4 | 1e8c22b2e17c5f88 |
| net.bytebuddy.matcher.MethodSortMatcher.Sort.5 | aae69164dd78b1e8 |
| net.bytebuddy.matcher.ModifierMatcher | 1f940a6dd9fa9613 |
| net.bytebuddy.matcher.ModifierMatcher.Mode | b59c67438c4008d8 |
| net.bytebuddy.matcher.NameMatcher | c95f2d97c50d769e |
| net.bytebuddy.matcher.NegatingMatcher | e10261097b62acbb |
| net.bytebuddy.matcher.SignatureTokenMatcher | 2b22b0f9e11d2a2b |
| net.bytebuddy.matcher.StringMatcher | 398d01f869388e91 |

| | |
|---|-------------------|
| net.bytebuddy.matcher.StringMatcher.Mode | 6f8b4d3695faa058 |
| net.bytebuddy.matcher.StringMatcher.Mode.1 | 015bef3b6828cdac |
| net.bytebuddy.matcher.StringMatcher.Mode.2 | 8eeeb4556b9485c0 |
| net.bytebuddy.matcher.StringMatcher.Mode.3 | f1c175ea33668dcba |
| net.bytebuddy.matcher.StringMatcher.Mode.4 | 84a59a498fb10fed |
| net.bytebuddy.matcher.StringMatcher.Mode.5 | 4ca9c62936b66a24 |
| net.bytebuddy.matcher.StringMatcher.Mode.6 | f993707f30c5fce2 |
| net.bytebuddy.matcher.StringMatcher.Mode.7 | e0ba3c2aa14bde78 |
| net.bytebuddy.matcher.StringMatcher.Mode.8 | 4ff86085d0aa4ddf |
| net.bytebuddy.matcher.StringMatcher.Mode.9 | 1dcf4a6fe509c5b5 |
| net.bytebuddy.matcher.SuperTypeMatcher | af5ab86fce6ba126 |
| net.bytebuddy.matcher.TypeSortMatcher | 7fa32c06737e7231 |
| net.bytebuddy.matcher.VisibilityMatcher | e1f91ccaffe7652 |
| net.bytebuddy.pool.TypePool.AbstractBase | ebf3f44b769f4faf |
| net.bytebuddy.pool.TypePool.AbstractBase.Hierarchical | 8051a701b4b0eed |
| net.bytebuddy.pool.TypePool.CacheProvider.NoOp | 5fe0b4fee828d480 |
| net.bytebuddy.pool.TypePool.CacheProvider.Simple | 5840a865f0dcfb51 |
| net.bytebuddy.pool.TypePool.ClassLoading | 4ff35dae9556b998 |
| net.bytebuddy.pool.TypePool.Default | 7aacf1f71b81e173 |
| net.bytebuddy.pool.TypePool.Default.ReaderMode | a033f5a27d0fa2d6 |
| net.bytebuddy.pool.TypePool.Empty | 256686863a3e855f |
| net.bytebuddy.pool.TypePool.Explicit | 58709ce964584fa6 |
| net.bytebuddy.utility.AsmClassReader | 8a59b649de2583f3 |
| net.bytebuddy.utility.AsmClassReader.Factory.Default | fa0b1b24f69493ab |
| net.bytebuddy.utility.AsmClassReader.Factory.Default.1 | 03c829950d692f4b |
| net.bytebuddy.utility.AsmClassReader.Factory.Default.2 | ff7ba2ccbe1277cf |
| net.bytebuddy.utility.AsmClassReader.Factory.Default.3 | d8e2a9d1378063fa |
| net.bytebuddy.utility.AsmClassReader.Factory.Default.4 | b200bb40f16587c9 |
| net.bytebuddy.utility.AsmClassReader.Factory.Default.5 | f9eab20e73a925e7 |
| net.bytebuddy.utility.AsmClassReader.ForAsm | c3d16f9e1e03618e |
| net.bytebuddy.utility.AsmClassWriter.Factory.Default | df7a47693cf75e36 |
| net.bytebuddy.utility.AsmClassWriter.Factory.Default.1 | d17bf1b9989584cf |
| net.bytebuddy.utility.AsmClassWriter.Factory.Default.2 | 3ebcd096754fad75 |
| net.bytebuddy.utility.AsmClassWriter.Factory.Default.3 | 6d6ee87b0257e85e |
| net.bytebuddy.utility.AsmClassWriter.Factory.Default.4 | 582d01970af50976 |
| net.bytebuddy.utility.AsmClassWriter.Factory.Default.5 | af553c430b6af86a |
| net.bytebuddy.utility.AsmClassWriter.Factory.Default.EmptyAsmClassReader | 481bb30e4232c02d |
| net.bytebuddy.utility.AsmClassWriter.ForAsm | 94e2175d9a1aa7a |
| net.bytebuddy.utility.AsmClassWriter.FrameComputingClassWriter | 943f69745cdb7621 |
| net.bytebuddy.utility.CompoundList | 41fe1faec9b96005 |
| net.bytebuddy.utility.ConstantValue.Simple | 5f7232051686a271 |
| net.bytebuddy.utility.ConstructorComparator | 3566d64bbe3006a2 |
| net.bytebuddy.utility.FieldComparator | 5dff49d0b60b3a07 |
| net.bytebuddy.utility.GraalImageCode | 20c183e97cdf38cc |
| net.bytebuddy.utility.Invoker.Dispatcher | b9b5f67cf01bb049 |
| net.bytebuddy.utility.JavaConstant.Simple | d18c769228ec6798 |
| net.bytebuddy.utility.JavaConstant.Simple.OfTrivialValue | 870135d683945e69 |
| net.bytebuddy.utility.JavaConstant.Simple.OfTrivialValue.ForString | 11ffe7957b103e25 |
| net.bytebuddy.utility.JavaModule | fb89d312129d6105 |
| net.bytebuddy.utility.JavaType | 4ea4e7db3d74934a |
| net.bytebuddy.utility.JavaType.LatentTypeWithSimpleName | c99024b8d6d1bb99 |
| net.bytebuddy.utility.MethodComparator | 2a643ad1fb57d7e8 |
| net.bytebuddy.utility.OpenedClassReader | 3495a559539ff5e7 |
| net.bytebuddy.utility.RandomString | a0583349bb66a97c |
| net.bytebuddy.utility.StreamDrainer | e95aa53cfc417c6 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher | 619d3930cae44455 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.Dispatcher.ForContainerCreation | 9f5631d45e9601e9 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.Dispatcher.ForInstanceCheck | ec7e2f5ae4036bb1 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.Dispatcher.ForNonStaticMethod | 3ff776c0c7a05881 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.Dispatcher.ForStaticMethod | faa6cd7f5e4cbd85 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.DynamicClassLoader | e55e160d8d2ebd92 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.DynamicClassLoader.Resolver.CreationAction | 862b139a62f264a8 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.DynamicClassLoader.Resolver.ForModuleSystem | 1a2ee856ca3b45b6 |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.InvokerCreationAction | f5b3814ad34536fc |
| net.bytebuddy.utility.dispatcher.JavaDispatcher.ProxyInvocationHandler | f1e193453808acbe |
| net.bytebuddy.utility.privilege.GetSystemPropertyAction | 67f0615a1253ad61 |
| net.bytebuddy.utility.visitor.ExceptionTableSensitiveMethodVisitor | a6c239974eefe586 |
| net.bytebuddy.utility.visitor.LineNumberPrependingMethodVisitor | 56aead9b55055305 |
| net.bytebuddy.utility.visitor.MetadataAwareClassVisitor | 0c557b09efdf9fd6 |
| net.bytebuddy.utility.visitor.StackAwareMethodVisitor | 3dd1c2239bb0bf19 |
| org.apache.maven.plugin.surefire.log.api.NullConsoleLogger | 50e0945fec76b333 |
| org.apache.maven.surefire.api.booter.BaseProviderFactory | da939a0152866a4b |
| org.apache.maven.surefire.api.booter.BiProperty | ed0281592f3976b4 |
| org.apache.maven.surefire.api.booter.Command | 52d7b732759793ff |

| | |
|---|-------------------|
| org.apache.maven.surefire.api.booter.Constants | 8f58b0da27218c74 |
| org.apache.maven.surefire.api.booter.DumpErrorSingleton | ea25742803c9e73f |
| org.apache.maven.surefire.api.booter.ForkedProcessEvent | 4f32ae2d4e670365 |
| org.apache.maven.surefire.api.booter.ForkingReporterFactory | be06f83accc5a8aa |
| org.apache.maven.surefire.api.booter.ForkingRunListener | c34d0a9f28f66585 |
| org.apache.maven.surefire.api.booter.MasterProcessCommand | fc8c116a509256d1 |
| org.apache.maven.surefire.api.booter.Shutdown | 47a37ed2a684ef1d |
| org.apache.maven.surefire.api.cli.CommandLineOption | 5825f848ee2abcd7 |
| org.apache.maven.surefire.api.provider.AbstractProvider | 0fea65ed91d7c12a |
| org.apache.maven.surefire.api.report.ConsoleOutputCapture | 7ee3451cf95e2f70 |
| org.apache.maven.surefire.api.report.ConsoleOutputCapture.ForwardingPrintStream | 804935f758ebaee3 |
| org.apache.maven.surefire.api.report.ConsoleOutputCapture.NullOutputStream | a81300d2d50decb6 |
| org.apache.maven.surefire.api.report.ReporterConfiguration | bf4075c0385296c2 |
| org.apache.maven.surefire.api.report.RunMode | 70edc0a9dea60143 |
| org.apache.maven.surefire.api.report.SimpleReportEntry | 5acc6a35bed0445f |
| org.apache.maven.surefire.api.stream.AbstractStreamDecoder | c6f3b2781f9ac881 |
| org.apache.maven.surefire.api.stream.AbstractStreamDecoder.BufferedStream | 11f69a75bc1c7211 |
| org.apache.maven.surefire.api.stream.AbstractStreamDecoder.Memento | e504a9e8cfc028af |
| org.apache.maven.surefire.api.stream.AbstractStreamDecoder.Segment | 773004ac6cd115ef |
| org.apache.maven.surefire.api.stream.AbstractStreamDecoder.StreamReadStatus | 8d5ee1d510b5c935 |
| org.apache.maven.surefire.api.stream.AbstractStreamEncoder | 9547668418a858ad |
| org.apache.maven.surefire.api.stream.SegmentType | 77b0d78ed3dd126 |
| org.apache.maven.surefire.api.suite.RunResult | 0eef4ae883b6fc当地 |
| org.apache.maven.surefire.api.testset.DirectoryScannerParameters | 529e83b831c47f72 |
| org.apache.maven.surefire.api.testset.IncludedExcludedPatterns | e12220ce508068df |
| org.apache.maven.surefire.api.testset.ResolvedTest | 119a5faa0ae08a91 |
| org.apache.maven.surefire.api.testset.ResolvedTest.ClassMatcher | cb9dd1b6069a872b |
| org.apache.maven.surefire.api.testset.ResolvedTest.MethodMatcher | 1d5196f3dfcebd52 |
| org.apache.maven.surefire.api.testset.ResolvedTest.Type | 6f46eedd1917ca66 |
| org.apache.maven.surefire.api.testset.RunOrderParameters | f74f6b3eb9f1a132 |
| org.apache.maven.surefire.api.testset.TestArtifactInfo | 6d162cddde2db959 |
| org.apache.maven.surefire.api.testset.TestListResolver | 0f4645f0d7fd02c8 |
| org.apache.maven.surefire.api.testset.TestRequest | 1cb2946d8f0dc9e4 |
| org.apache.maven.surefire.api.util.CloseableIterator | 01846c357efacab7b |
| org.apache.maven.surefire.api.util.DefaultRunOrderCalculator | 21a42ec0f6d63b8e |
| org.apache.maven.surefire.api.util.DefaultScanResult | 01695a339c66ab8d |
| org.apache.maven.surefire.api.util.ReflectionUtils | 7f9a430ae144c985 |
| org.apache.maven.surefire.api.util.RunOrder | 93376844e6d709d3 |
| org.apache.maven.surefire.api.util.TestsToRun | db4e8195893ece6d |
| org.apache.maven.surefire.api.util.TestsToRun.ClassesIterator | 543f26bfbdd04ce0 |
| org.apache.maven.surefire.api.util.internal.AbstractNoninterruptibleReadableChannel | 6826ce793980b64e |
| org.apache.maven.surefire.api.util.internal.AbstractNoninterruptibleWritableChannel | 484afc5593fbc9a |
| org.apache.maven.surefire.api.util.internal.Channels | eb60281181a1dc33 |
| org.apache.maven.surefire.api.util.internal.Channels.3 | 605144c3f67338aa |
| org.apache.maven.surefire.api.util.internal.Channels.4 | 4834cf9402eabd28 |
| org.apache.maven.surefire.api.util.internal.ClassMethod | 817ad544e129b000 |
| org.apache.maven.surefire.api.util.internal.DaemonThreadFactory | b2161e778265b95d |
| org.apache.maven.surefire.api.util.internal.DaemonThreadFactory.NamedThreadFactory | e3fb668fa8792230 |
| org.apache.maven.surefire.api.util.internal.DumpFileUtils | 9cc0f89ffba46ba32 |
| org.apache.maven.surefire.api.util.internal.ImmutableMap | c7398d64c0977b06 |
| org.apache.maven.surefire.api.util.internal.ImmutableMap.Node | 3a9862055afaee58 |
| org.apache.maven.surefire.api.util.internal.ObjectUtils | 992d9f9f62042416 |
| org.apache.maven.surefire.booter.AbstractPathConfiguration | f8b4034fe9c934d2 |
| org.apache.maven.surefire.booter.BooterDeserializer | d2b4a565d2c195cc |
| org.apache.maven.surefire.booter.ClassLoaderConfiguration | c511fbfeb1f35c23 |
| org.apache.maven.surefire.booter.Classpath | d05af49602124353 |
| org.apache.maven.surefire.booter.ClasspathConfiguration | d14c58928ac6aa7b |
| org.apache.maven.surefire.booter.CommandReader | 8bc1181d0c5af474 |
| org.apache.maven.surefire.booter.CommandReader.1 | 72a8e2906ddc1c93 |
| org.apache.maven.surefire.booter.CommandReader.CommandRunnable | f6a6b02be2fb0964 |
| org.apache.maven.surefire.booter.ForkedBooter | c8ce6ed3be8ec9bc |
| org.apache.maven.surefire.booter.ForkedBooter.1 | 68f2dae15ae26cc2 |
| org.apache.maven.surefire.booter.ForkedBooter.3 | fc217f2c1d87c099 |
| org.apache.maven.surefire.booter.ForkedBooter.4 | 2afb302f7c81f991 |
| org.apache.maven.surefire.booter.ForkedBooter.6 | 850ef2748b5ef5e6 |
| org.apache.maven.surefire.booter.ForkedBooter.7 | 9577114e02a5bdef |
| org.apache.maven.surefire.booter.ForkedBooter.8 | 3c8feb047cd2b0c |
| org.apache.maven.surefire.booter.ForkedBooter.PingScheduler | c83e3af27d5d3c47 |
| org.apache.maven.surefire.booter.ForkedNodeArg | 9dbb0ff22dfc1303 |
| org.apache.maven.surefire.booter.PpidChecker | f83a9169197e13b1 |
| org.apache.maven.surefire.booter.ProcessCheckerType | e554be35191ff5a7 |
| org.apache.maven.surefire.booter.PropertiesWrapper | 1e4e30276db2e62e |
| org.apache.maven.surefire.booter.ProviderConfiguration | ec2cd1e39ec4278e |
| org.apache.maven.surefire.booter.StartupConfiguration | 70176a3dd903d57a |
| org.apache.maven.surefire.booter.SystemPropertyManager | a843c08e9b5c79ad |

| | |
|---|-------------------|
| org.apache.maven.surefire.booter.TypeEncodedValue | 355d20d53741b604 |
| org.apache.maven.surefire.booter.spi.AbstractMasterProcessChannelProcessorFactory | 67a1c051e3809086 |
| org.apache.maven.surefire.booter.spi.AbstractMasterProcessChannelProcessorFactory.1 | cc936f6c85f9235a |
| org.apache.maven.surefire.booter.spi.AbstractMasterProcessChannelProcessorFactory.2 | a1fa70e4af42c555 |
| org.apache.maven.surefire.booter.spi.CommandChannelDecoder | 6684e6bad0b7c71e |
| org.apache.maven.surefire.booter.spi.EventChannelEncoder | b69d9287bf010b1a |
| org.apache.maven.surefire.booter.spi.EventChannelEncoder.StackTrace | 265e85a5e039b0af |
| org.apache.maven.surefire.booter.spi.LegacyMasterProcessChannelProcessorFactory | 3b29862697f79d34 |
| org.apache.maven.surefire.booter.spi.SurefireMasterProcessChannelProcessorFactory | 8c14c673718fba9e |
| org.apache.maven.surefire.booter.stream.CommandDecoder | a23a4082e2bbd1ed |
| org.apache.maven.surefire.booter.stream.CommandDecoder.1 | 950700970edca54a |
| org.apache.maven.surefire.booter.stream.EventEncoder | 7c894cb22c8c16ca |
| org.apache.maven.surefire.junitplatform.JUnitPlatformProvider | 958f7eb4311b3c2f |
| org.apache.maven.surefire.junitplatform.LazyLauncher | a3841276826f155c |
| org.apache.maven.surefire.junitplatform.RunListenerAdapter | 0d7041faa0298e70 |
| org.apache.maven.surefire.junitplatform.RunListenerAdapter.1 | 967ebdaaeeef83363 |
| org.apache.maven.surefire.junitplatform.TestPlanScannerFilter | db2b13639af3176e |
| org.apache.maven.surefire.report.ClassMethodIndexer | 0e8f3008aec84fc8 |
| org.apache.maven.surefire.shared.lang3.JavaVersion | a902b52c460c0348 |
| org.apache.maven.surefire.shared.lang3.StringUtils | 4628d7808116e372 |
| org.apache.maven.surefire.shared.lang3.SystemProperties | 6b2fea785d2a2915 |
| org.apache.maven.surefire.shared.lang3.SystemUtils | 2518da556699ab1e |
| org.apache.maven.surefire.shared.lang3.function.Suppliers | 6cb739fdbd96d7c1 |
| org.apache.maven.surefire.shared.lang3.math.NumberUtils | 99f301ade68669b7 |
| org.apache.maven.surefire.shared.utils.StringUtils | abd8480c7152bf46 |
| org.apache.maven.surefire.shared.utils.cli.ShutdownHookUtils | 011b23cd829ec86c |
| org.apiguardian.api.API.Status | 95d0ffea805fc01a |
| org.junit.jupiter.api.AssertDoesNotThrow | 36b9cb12d1985e50 |
| org.junit.jupiter.api.Assertions | 02e79388fd0ddf18 |
| org.junit.jupiter.api.IsFalse | dea6dc33450c92f0 |
| org.junit.jupiter.api.IsNotNull | 34eb9c4ee51b2816 |
| org.junit.jupiter.api.IsNotNull | 36f7b673f5497507 |
| org.junit.jupiter.api.AreEqual | 77061b15f086bbc5 |
| org.junit.jupiter.api.AreSame | 2e413933639a681e |
| org.junit.jupiter.api.AssertThrows | 6ef3923800860200 |
| org.junit.jupiter.api.IsTrue | a580a647f9b0d1af |
| org.junit.jupiter.api.AssumptionUtils | a837ed10bf9804f2 |
| org.junit.jupiter.api.Assertions | 1c70d4d828122f05 |
| org.junit.jupiter.api.DisplayNameGenerator | b23b44fe1a1ae4b6 |
| org.junit.jupiter.api.DisplayNameGenerator.IndicativeSentences | 45af1f815eb3bfc6 |
| org.junit.jupiter.api.DisplayNameGenerator.ReplaceUnderscores | 3587fc3bd5ac68a7 |
| org.junit.jupiter.api.DisplayNameGenerator.Simple | 232bffa51a0c4e |
| org.junit.jupiter.api.DisplayNameGenerator.Standard | 235138c6ffffd45f1 |
| org.junit.jupiter.api.TestInstanceLifecycle | fc311dfabd3a0e23 |
| org.junit.jupiter.api.extension.ConditionEvaluationResult | dacb7330135ba8f9 |
| org.junit.jupiter.api.extension.ExtensionContext | eb8d03782ab35d64 |
| org.junit.jupiter.api.extension.ExtensionContext.Namespace | 288780f400093c7c |
| org.junit.jupiter.api.extension.ExtensionContext.Store | 695ac2a6b4b9c7e4 |
| org.junit.jupiter.api.extension.InvocationInterceptor | 011031d0b1fe58db |
| org.junit.jupiter.engine.JupiterTestEngine | 9da5fe6b78ad9a14 |
| org.junit.jupiter.engine.config.CachingJupiterConfiguration | bbee9c72790c271d |
| org.junit.jupiter.engine.config.DefaultJupiterConfiguration | 433eec982a6fabbc |
| org.junit.jupiter.engine.config.EnumConfigurationParameterConverter | d2270f0957971443 |
| org.junit.jupiter.engine.config.InstantiatingConfigurationParameterConverter | 6b3fc41ad8b41d4f |
| org.junit.jupiter.engine.descriptor.AbstractExtensionContext | 414ee653c9e673cf |
| org.junit.jupiter.engine.descriptor.ClassBasedTestDescriptor | e804dacaeaf4a6a |
| org.junit.jupiter.engine.descriptor.ClassExtensionContext | 2f87db51b4485e07 |
| org.junit.jupiter.engine.descriptor.ClassTestDescriptor | b1b7d61e94c58605 |
| org.junit.jupiter.engine.descriptor.DefaultTestInstanceFactoryContext | 8a6f8eeb3e12ddf6 |
| org.junit.jupiter.engine.descriptor.DisplayNameUtils | 43a683ad1b768e92 |
| org.junit.jupiter.engine.descriptor.ExtensionUtils | 3d2bdbdce296b041 |
| org.junit.jupiter.engine.descriptor.JupiterEngineDescriptor | 7146ce9988edfce2 |
| org.junit.jupiter.engine.descriptor.JupiterEngineExtensionContext | 67ad750cdb2cb53b |
| org.junit.jupiter.engine.descriptor.JupiterTestDescriptor | 286eb923d0b68032 |
| org.junit.jupiter.engine.descriptor.LifecycleMethodUtils | f531f49451e39050 |
| org.junit.jupiter.engine.descriptor.MethodBasedTestDescriptor | b5abe6523f4a32d7 |
| org.junit.jupiter.engine.descriptor.MethodExtensionContext | a247fc379f47df66 |
| org.junit.jupiter.engine.descriptor.TestInstanceLifecycleUtils | 35334f82ecefa63c |
| org.junit.jupiter.engine.descriptor.TestMethodTestDescriptor | 90b10f2d90d7b01b |
| org.junit.jupiter.engine.discovery.AbstractAnnotatedDescriptorWrapper | f8eb297929c247eb |
| org.junit.jupiter.engine.discovery.AbstractOrderingVisitor | c8e1585f8474ed61 |
| org.junit.jupiter.engine.discovery.AbstractOrderingVisitor.DescriptorWrapperOrderer | 1f09fc1c6b9779bb |
| org.junit.jupiter.engine.discovery.ClassOrderingVisitor | e25bb2b197bc8493 |
| org.junit.jupiter.engine.discovery.ClassSelectorResolver | 9064f3528773a161 |
| org.junit.jupiter.engine.discovery.DefaultClassDescriptor | 5dc6be896f50996f |
| org.junit.jupiter.engine.discovery.DiscoverySelectorResolver | |

| | |
|--|-------------------|
| org.junit.jupiter.engine.discovery.MethodFinder | 621c8591e557439a |
| org.junit.jupiter.engine.discovery.MethodOrderingVisitor | 7d9864ceb818e1 |
| org.junit.jupiter.engine.discovery.MethodSelectorResolver | 679c52dec5ee3cd2 |
| org.junit.jupiter.engine.discovery.MethodSelectorResolver.MethodType | 2ca704c5264882ae |
| org.junit.jupiter.engine.discovery.MethodSelectorResolver.MethodType.1 | b3bc3007a7dfdaa0 |
| org.junit.jupiter.engine.discovery.MethodSelectorResolver.MethodType.2 | 598aec8eefef85e3 |
| org.junit.jupiter.engine.discovery.MethodSelectorResolver.MethodType.3 | e8fd5325e2431a2b |
| org.junit.jupiter.engine.discovery.predicates.IsInnerClass | d746bcff9a71ec26 |
| org.junit.jupiter.engine.discovery.predicates.IsNestedTestClass | f75dfd9ee2347890 |
| org.junit.jupiter.engine.discovery.predicates.IsPotentialTestContainer | 909f14a1b9fe84dc |
| org.junit.jupiter.engine.discovery.predicates.IsTestClassWithTests | 34690a186bfcf3ac |
| org.junit.jupiter.engine.discovery.predicates.IsTestFactoryMethod | 941a8af0d47a68fd |
| org.junit.jupiter.engine.discovery.predicates.IsTestMethod | f2039dbd13fce110 |
| org.junit.jupiter.engine.discovery.predicates.IsTestTemplateMethod | c13a4260435c18a8 |
| org.junit.jupiter.engine.discovery.predicates.IsTestableMethod | 4be487dee199f633 |
| org.junit.jupiter.engine.execution.ConditionEvaluator | df91d94b180fe511 |
| org.junit.jupiter.engine.execution.ConstructorInvocation | 60b80968f2bdedc3 |
| org.junit.jupiter.engine.execution.DefaultExecutableInvoker | 97f15d1e3151968f |
| org.junit.jupiter.engine.execution.DefaultTestInstances | 0fc6d90567826bc4 |
| org.junit.jupiter.engine.execution.InterceptingExecutableInvoker | 42cb185ff5e76387 |
| org.junit.jupiter.engine.execution.InterceptingExecutableInvoker.ReflectiveInterceptorCall | 7e154d03f7a732e5 |
| org.junit.jupiter.engine.execution.InvocationInterceptorChain | 9798b2a812d2015d |
| org.junit.jupiter.engine.execution.InvocationInterceptorChain.InterceptedInvocation | 199eef1acbe0b316 |
| org.junit.jupiter.engine.execution.InvocationInterceptorChain.ValidatingInvocation | f064b1c2c4a4bf86 |
| org.junit.jupiter.engine.execution.JupiterEngineExecutionContext | b48cc2a96dab0116 |
| org.junit.jupiter.engine.execution.JupiterEngineExecutionContext.Builder | d1557432e23d2776 |
| org.junit.jupiter.engine.execution.JupiterEngineExecutionContext.State | 3926323ef1c7fb03 |
| org.junit.jupiter.engine.execution.MethodInvocation | 8b8fd00463d994df |
| org.junit.jupiter.engine.execution.NamespaceAwareStore | 00e5ea1337f34969 |
| org.junit.jupiter.engine.execution.ParameterResolutionUtils | 5aba48e342016f8f |
| org.junit.jupiter.engine.execution.TestInstancesProvider | 357bca6226069e7b |
| org.junit.jupiter.engine.extension.DisabledCondition | 1604b4e34c1363e4 |
| org.junit.jupiter.engine.extension.ExtensionRegistry | 687649643dbb04fc |
| org.junit.jupiter.engine.extension.MutableExtensionRegistry | 4daca7ba95c88845 |
| org.junit.jupiter.engine.extension.RepeatedTestExtension | 7a30afad0f944ea5 |
| org.junit.jupiter.engine.extension.TempDirectory | d2ce4804a30f8d8c |
| org.junit.jupiter.engine.extension.TempDirectory.Scope | 3a056889e3e86fe7 |
| org.junit.jupiter.engine.extension.TestInfoParameterResolver | 3c520f8376f91ff7 |
| org.junit.jupiter.engine.extension.TestReporterParameterResolver | 7187071bfc76c6ac |
| org.junit.jupiter.engine.extension.TimeoutConfiguration | 44b8593a8e980687 |
| org.junit.jupiter.engine.extension.TimeoutDurationParser | bb6a412c3829dae9 |
| org.junit.jupiter.engine.extension.TimeoutExtension | 13bcdadb20fc7bb |
| org.junit.jupiter.engine.support.JupiterThrowableCollectorFactory | 46546a446de4c9c0 |
| org.junit.jupiter.engine.support.OpenTest4AndJUnit4AwareThrowableCollector | e9ee7d4e1adeccdd1 |
| org.junit.platform.commons.function.Try | 5200e6adc191344c |
| org.junit.platform.commons.function.Try.Failure | 5d1cf7b52cd7a7ea |
| org.junit.platform.commons.function.Try.Success | 98cdc5b539e1abfd |
| org.junit.platform.commons.logging.LoggerFactory | 39fdfe1f67bc0eda |
| org.junit.platform.commons.logging.LoggerFactory.DelegatingLogger | c71dcf008235901c |
| org.junit.platform.commons.support.AnnotationSupport | 4b0c63263b83acb5 |
| org.junit.platform.commons.support.ReflectionSupport | db9de9450da5225a |
| org.junit.platform.commons.util.AnnotationUtils | efebc064783617e1 |
| org.junit.platform.commons.util.ClassLoaderUtils | 0d0959e2f6aa173e |
| org.junit.platform.commons.util.ClassNamePatternFilterUtils | e725a6f058746f53 |
| org.junit.platform.commons.util.ClassUtils | 60a2276f3701443f |
| org.junit.platform.commons.util.ClasspathScanner | 54e3df9bb2092b52 |
| org.junit.platform.commons.util.CollectionUtils | d47999c87f911057 |
| org.junit.platform.commons.util.Preconditions | 2c2a6e13cda880d4 |
| org.junit.platform.commons.util.ReflectionUtils | 172cf9786a51e883 |
| org.junit.platform.commons.util.ReflectionUtils.HierarchyTraversalMode | 349d54e51f2ffb44 |
| org.junit.platform.commons.util.StringUtils | 237c0cb03ac19254 |
| org.junit.platform.commons.util.UnrecoverableExceptions | e906a774e770e7d4 |
| org.junit.platform.engine.CompositeFilter | 6a52e5b4f7292f48 |
| org.junit.platform.engine.CompositeFilter.1 | cc0aadcc5880fb4e4 |
| org.junit.platform.engine.EngineDiscoveryListener | c3024068e43bb7f4 |
| org.junit.platform.engine.EngineDiscoveryListener.1 | a4cdcb8dd38d8f57 |
| org.junit.platform.engine.EngineExecutionListener | 693fee5cbd4c2df0 |
| org.junit.platform.engine.EngineExecutionListener.1 | 999902b68f81dd9a |
| org.junit.platform.engine.ExecutionRequest | b74e001541d12dd1 |
| org.junit.platform.engine.Filter | 5ffaaa90df97ca04 |
| org.junit.platform.engine.FilterResult | a787a89e1f12d534 |
| org.junit.platform.engine.SelectorResolutionResult | ca52e15a278dcf5c |
| org.junit.platform.engine.SelectorResolutionResult.Status | c505c2274f89f01d |
| org.junit.platform.engine.TestDescriptor | a828437d5cd2ea4f |

| | |
|---|-------------------|
| org.junit.platform.engine.TestDescriptor.Type | 7628a7c639ef3a60 |
| org.junit.platform.engine.TestExecutionResult | 6b1b512d17bb680e |
| org.junit.platform.engine.TestExecutionResult.Status | ad256e9fb4407e04 |
| org.junit.platform.engine.UniqueId | 4308af7bfbd4ba1 |
| org.junit.platform.engine.UniqueId.Segment | f2d36a9ca9d14367 |
| org.junit.platform.engine.UniqueIdFormat | 6c86362ad62a1954 |
| org.junit.platform.engine.discovery.ClassSelector | 3174b37b3ba53b7e |
| org.junit.platform.engine.discovery.DiscoverySelectors | 7863536f4276f4dd |
| org.junit.platform.engine.discovery.MethodSelector | 3fe9eccb2ba205d2 |
| org.junit.platform.engine.support.descriptor.AbstractTestDescriptor | b9c965daf4d9a476 |
| org.junit.platform.engine.support.descriptor.ClassSource | 37bd92069360f773 |
| org.junit.platform.engine.support.descriptor.EngineDescriptor | 8f2f77769ee0e9c9 |
| org.junit.platform.engine.support.descriptor.MethodSource | 1d55ac49f5cab20 |
| org.junit.platform.engine.support.discovery.ClassContainerSelectorResolver | dc6114dc7e983729 |
| org.junit.platform.engine.support.discovery.EngineDiscoveryRequestResolution | 506a6b871d2fd8fe |
| org.junit.platform.engine.support.discovery.EngineDiscoveryRequestResolution.DefaultContext | db18f59764ea1f2a |
| org.junit.platform.engine.support.discovery.EngineDiscoveryRequestResolver | e7fb3042ea8112f0 |
| org.junit.platform.engine.support.discovery.EngineDiscoveryRequestResolver.Builder | d86618af76b95613 |
| org.junit.platform.engine.support.discovery.EngineDiscoveryRequestResolver.DefaultInitializationContext | 1904819635770d62 |
| org.junit.platform.engine.support.discovery.SelectorResolver | e64e4fd796d9641d |
| org.junit.platform.engine.support.discovery.SelectorResolver.Match | 789c682356298d75 |
| org.junit.platform.engine.support.discovery.SelectorResolver.Match.Type | 1761e56439c8d93c |
| org.junit.platform.engine.support.discovery.SelectorResolver.Resolution | ab713bdee405d17 |
| org.junit.platform.engine.support.hierarchical.ExclusiveResource | c29acbe41918b09a |
| org.junit.platform.engine.support.hierarchical.ExclusiveResource.LockMode | 96e95d210b150f97 |
| org.junit.platform.engine.support.hierarchical.HierarchicalTestEngine | 3ac292151741b7fc |
| org.junit.platform.engine.support.hierarchical.HierarchicalTestExecutor | 963cba9b029b4b19 |
| org.junit.platform.engine.support.hierarchical.LockManager | 5aedd3bd3957b5a6 |
| org.junit.platform.engine.support.hierarchical.Node | 5c68850150771b6e |
| org.junit.platform.engine.support.hierarchical.Node.SkipResult | 5aca1404ff0f9294 |
| org.junit.platform.engine.support.hierarchical.NodeExecutionAdvisor | 7c2670c7a35cfba6 |
| org.junit.platform.engine.support.hierarchical.NodeTestTask | f652d8cc5e11bdc5 |
| org.junit.platform.engine.support.hierarchical.NodeTestTask.DefaultDynamicTestExecutor | abd00dd511d28b2f |
| org.junit.platform.engine.support.hierarchical.NodeTestTaskContext | bdf88cd3834282a5 |
| org.junit.platform.engine.support.hierarchical.NodeTreeWalker | c689092b060d0b12 |
| org.junit.platform.engine.support.hierarchical.NodeUtils | a7ec8f66d373c169 |
| org.junit.platform.engine.support.hierarchical.NodeUtils.1 | 5a44a7e2cbf864b4 |
| org.junit.platform.engine.support.hierarchical.SameThreadHierarchicalTestExecutorService | 2f3b283eba81629f |
| org.junit.platform.engine.support.hierarchical.SingleLock | 2036ec8b92a38105 |
| org.junit.platform.engine.support.hierarchical.ThrowableCollector | 6fd7a27676be3c50 |
| org.junit.platform.engine.support.store.NamespacedHierarchicalStore | f773d297d7dc3275 |
| org.junit.platform.engine.support.store.NamespacedHierarchicalStore.CompositeKey | 3f8758b273ff41a9 |
| org.junit.platform.engine.support.store.NamespacedHierarchicalStore.EvaluatedValue | 3362298f87d9b160 |
| org.junit.platform.engine.support.store.NamespacedHierarchicalStore.MemoizingSupplier | be04f7b805ba11e1 |
| org.junit.platform.engine.support.store.NamespacedHierarchicalStore.StoredValue | 8e79d12821d1a835 |
| org.junit.platform.launcher.EngineDiscoveryResult | 44ae55d9c94cdd13 |
| org.junit.platform.launcher.EngineDiscoveryResult.Status | c6f73a818e869b3a |
| org.junit.platform.launcher.LauncherDiscoveryListener | c8e17526e895636b |
| org.junit.platform.launcher.LauncherDiscoveryListener.1 | 8959ed22ae756aca |
| org.junit.platform.launcher.LauncherSessionListener | fd09754de5a01f16 |
| org.junit.platform.launcher.LauncherSessionListener.1 | 44b3640faa83f474 |
| org.junit.platform.launcher.TestExecutionListener | f482f6546d6593dc |
| org.junit.platform.launcher.TestIdentifier | 2b393a1d76332bc4 |
| org.junit.platform.launcher.TestPlan | 125780e74ba9c50c |
| org.junit.platform.launcher.core.CompositeEngineExecutionListener | cea0030887322419 |
| org.junit.platform.launcher.core.CompositeTestExecutionListener | 283b3c281a0728e5 |
| org.junit.platform.launcher.core.DefaultDiscoveryRequest | 5706e3938a47edbc |
| org.junit.platform.launcher.core.DefaultLauncher | 0bd6690ec3f385ab |
| org.junit.platform.launcher.core.DefaultLauncherConfig | 6fbfe73d83f861ce |
| org.junit.platform.launcher.core.DefaultLauncherSession | 593c9fadcd439bc2 |
| org.junit.platform.launcher.core.DefaultLauncherSession.1 | 4e7ad5e44df7008e |
| org.junit.platform.launcher.core.DefaultLauncherSession.ClosedLauncher | 1fe238faa78c4ee2 |
| org.junit.platform.launcher.core.DelegatingEngineExecutionListener | 98129d4f91790da1 |
| org.junit.platform.launcher.core.DelegatingLauncher | 443e4e7cef8118ba |
| org.junit.platform.launcher.core.EngineDiscoveryOrchestrator | 9260ad30b5b1dcba |
| org.junit.platform.launcher.core.EngineDiscoveryOrchestrator.Phase | c5da52319ffdb6cc |
| org.junit.platform.launcher.core.EngineDiscoveryResultValidator | 241befbef6ea2edf |
| org.junit.platform.launcher.core.EngineExecutionOrchestrator | 61a7d44fcacf1fd6d |
| org.junit.platform.launcher.core.EngineFilterer | 5886e10a3932fe3b |
| org.junit.platform.launcher.core.EnginelValidator | a3cbf4111f4706bd |
| org.junit.platform.launcher.core.ExecutionListenerAdapter | 027b702b863a1b7b |
| org.junit.platform.launcher.core.InternalTestPlan | 6c1da5c749fc1754 |
| org.junit.platform.launcher.core.IterationOrder | 67fb bac106398c55 |
| org.junit.platform.launcher.core.IterationOrder.1 | c32d4c631876b3d3 |

| | |
|--|-------------------|
| org.junit.platform.launcher.core.IterationOrder\$2 | b3c544910702c338 |
| org.junit.platform.launcher.core.LauncherConfig | 58100dc14c875cb9 |
| org.junit.platform.launcher.core.LauncherConfig\$Builder | b0426f929eec8a53 |
| org.junit.platform.launcher.core.LauncherConfigurationParameters | 443c9d189d7662aa |
| org.junit.platform.launcher.core.LauncherConfigurationParameters\$Builder | 89b3d95a424a68ea |
| org.junit.platform.launcher.core.LauncherConfigurationParameters\$ParameterProvider | da0ae1240b20de42 |
| org.junit.platform.launcher.core.LauncherConfigurationParameters\$ParameterProvider\$2 | 481aeb52e3ac15c4 |
| org.junit.platform.launcher.core.LauncherConfigurationParameters\$ParameterProvider\$3 | 2d8e65fa362495e2 |
| org.junit.platform.launcher.core.LauncherDiscoveryRequestBuilder | 8aa84e8c1156fc9d |
| org.junit.platform.launcher.core.LauncherDiscoveryResult | 6ba764b26de92159 |
| org.junit.platform.launcher.core.LauncherFactory | 7c870cd17431cb9d |
| org.junit.platform.launcher.core.LauncherListenerRegistry | 64d5f2a8ac991f94 |
| org.junit.platform.launcher.core.ListenerRegistry | 387fd40f10f1e6b5 |
| org.junit.platform.launcher.core.OutcomeDelayingEngineExecutionListener | 4c68ad66a29b4dd7 |
| org.junit.platform.launcher.core.OutcomeDelayingEngineExecutionListener\$Outcome | b6ca0889820c3cca |
| org.junit.platform.launcher.core.ServiceLoaderRegistry | 2a95faa488a889e7 |
| org.junit.platform.launcher.core.ServiceLoaderTestEngineRegistry | 69f4349cc7042ed7 |
| org.junit.platform.launcher.core.StackTracePruningEngineExecutionListener | dbf05583a874b58d |
| org.junit.platform.launcher.core.StreamInterceptingTestExecutionListener | 36972af5e542435 |
| org.junit.platform.launcher.listeners.UniqueIdTrackingListener | f828b9fe46e426f0 |
| org.junit.platform.launcher.listeners.discovery.AbortOnFailureLauncherDiscoveryListener | ee6720edc40a9ccf |
| org.junit.platform.launcher.listeners.discovery.LauncherDiscoveryListeners | 03063623efb5e8b2 |
| org.junit.platform.listeners.discovery.LauncherDiscoveryListeners\$LauncherDiscoveryListenerType | e18e1a0e62e22287 |
| org.junit.platform.launcher.listeners.session.LauncherSessionListeners | 792ecbf10e49d607 |
| org.mindrot.jbcrypt.BCrypt | 6688b2728af0e4ae |
| org.mockito.Answers | af86bd70185fc83 |
| org.mockito.ArgumentMatchers | 30453f217beea892 |
| org.mockito.Mockito | c6bfaacf92b6f13dc |
| org.mockito.configuration.DefaultMockitoConfiguration | b174879ae8ed115e |
| org.mockito.internal.MockedStaticImpl | 0ddd8f20e599e4f9 |
| org.mockito.internal.MockitoCore | f49fbfc3fe5350e5 |
| org.mockito.internal.PremainAttachAccess | 868bee0538961f6a |
| org.mockito.internal.configuration.CaptorAnnotationProcessor | 2f21a4570b50b64a |
| org.mockito.internal.configuration.ClassPathLoader | 173a7c62160e6dbf |
| org.mockito.internal.configuration.DefaultDoNotMockEnforcer | 6a7cb49285062e7d |
| org.mockito.internal.configuration.GlobalConfiguration | 0df96c19dabdcfc0 |
| org.mockito.internal.configuration.IndependentAnnotationEngine | 54aaab1155cc41fd |
| org.mockito.internal.configuration.InjectingAnnotationEngine | 3402d3906098d7e2 |
| org.mockito.internal.configuration.MockAnnotationProcessor | f32d9954d5c65205 |
| org.mockito.internal.configuration.SpyAnnotationEngine | 6b53375c8a8a5cc1 |
| org.mockito.internal.configuration.plugins.DefaultMockitoPlugins | f2b7ceb1ff6789f3 |
| org.mockito.internal.configuration.plugins.DefaultPluginSwitch | bae35df711d1f747 |
| org.mockito.internal.configuration.plugins.PluginFinder | bd3ccb4ee283ccc1 |
| org.mockito.internal.configuration.plugins.PluginInitializer | 072018c08a02e9ee |
| org.mockito.internal.configuration.plugins.PluginLoader | 1702b486e8f8c9ad |
| org.mockito.internal.configuration.plugins.PluginRegistry | eba7ea1c6a85364 |
| org.mockito.internal.configuration.plugins.Plugins | b0a44acc68acddd8 |
| org.mockito.internal.creation.DelegatingMethod | aa9a3605cadc5938 |
| org.mockito.internal.creation.MockSettingsImpl | 3040d7b3b87e5329 |
| org.mockito.internal.creation.SuspendMethod | 5807a496dfc9c4c6 |
| org.mockito.internal.creation.bytebuddy.ByteBuddyCrossClassLoaderSerializationSupport | b07e753f8b4d10a1 |
| org.mockito.internal.creation.bytebuddy.BytecodeGenerator | b96181544d17b32a |
| org.mockito.internal.creation.bytebuddy.InlineByteBuddyMockMaker | a1a0ac895421946d |
| org.mockito.internal.creation.bytebuddy.InlineBytecodeGenerator | b3e304d2e253eac8 |
| org.mockito.internal.creation.bytebuddy.InlineBytecodeGenerator\$ParameterWritingVisitorWrapper | 391fcc69ac063d03 |
| org.mockito.internal.creation.bytebuddy.InlineBytecodeGenerator\$ParameterWritingVisitorWrapper\$MethodParameterStrippingMethodVisitor | 4daaaa2d160cb215 |
| org.mockito.internal.creation.bytebuddy.InlineBytecodeGenerator\$ParameterWritingVisitorWrapper\$ParameterAddingClassVisitor | 3ea0bd6b68949e7f |
| org.mockito.internal.creation.bytebuddy.InlineDelegateByteBuddyMockMaker | 2e5a8e3521b0d290 |
| org.mockito.internal.creation.bytebuddy.InlineDelegateByteBuddyMockMaker\$1 | 0c4b020122fedbe3 |
| org.mockito.internal.creation.bytebuddy.InlineDelegateByteBuddyMockMaker\$InlineStaticMockControl | 465e7d1306587f39 |
| org.mockito.internal.creation.bytebuddy.MockFeatures | 3948e31575d7acd |
| org.mockito.internal.creation.bytebuddy.MockMethodAdvice | 1db8cad96c916600 |
| org.mockito.internal.creation.bytebuddy.MockMethodAdvice\$ConstructorShortcut | ca8092fe743e119f |
| org.mockito.internal.creation.bytebuddy.MockMethodAdvice\$ConstructorShortcut\$1 | d1c660410fc1f7ca |
| org.mockito.internal.creation.bytebuddy.MockMethodAdvice\$RealMethodCall | c22f1055be0ba4d7 |
| org.mockito.internal.creation.bytebuddy.MockMethodAdvice\$ReturnValueWrapper | 5aec1a35f7fcfc57 |
| org.mockito.internal.creation.bytebuddy.MockMethodAdvice\$SelfCallInfo | 97851b59e4893c98 |
| org.mockito.internal.creation.bytebuddy.MockMethodAdvice\$StaticMethodCall | 31ce90c04a443966 |
| org.mockito.internal.creation.bytebuddy.ModuleHandler | addb759457799176 |
| org.mockito.internal.creation.bytebuddy.ModuleHandler\$3 | dbeec40635b946f97 |
| org.mockito.internal.creation.bytebuddy.ModuleHandler\$ModuleSystemFound | b4c7c55be2a75dc |
| org.mockito.internal.creation.bytebuddy.StackWalkerChecker | 68e569e3f7178506 |
| org.mockito.internal.creation.bytebuddy.SubclassBytecodeGenerator | d1fe149222160108 |
| org.mockito.internal.creation.bytebuddy.TypeCachingBytecodeGenerator | 9b77b7f9f15ce65b |

| | |
|---|-------------------|
| org.mockito.internal.creation.bytebuddy.TypeCachingBytecodeGenerator.MockitoMockKey | 6ab1c48e921f0e050 |
| org.mockito.internal.creation.bytebuddy.TypeCachingBytecodeGenerator.TypeCachingLock | f3718822abb34b6b |
| org.mockito.internal.creation.bytebuddy.TypeSupport | d98a4f5abd337730 |
| org.mockito.internal.creation.bytebuddy.access.MockMethodInterceptor | 7e19a7250dd27860 |
| org.mockito.internal.creation.bytebuddy.access.MockMethodInterceptor.DispatcherDefaultingToRealMethod | bc608d13bed06ccb |
| org.mockito.internal.creation.bytebuddy.codegen.Clip.MockitoMock.7xUI5mSN | 3565391c3a530a4a |
| org.mockito.internal.creation.bytebuddy.codegen.FloatControl.MockitoMock.KLgCKSLz | 198f019e66439ccf |
| org.mockito.internal.creation.instance.DefaultInstantiatorProvider | 844386c7887007f1 |
| org.mockito.internal.creation.instance.ObjenesisInstantiator | 7a7c1771759c8b2f |
| org.mockito.internal.creation.settings.CreationSettings | 1d27e65e4db303b9 |
| org.mockito.internal.debugging.Localized | df79022d5f63dcbe |
| org.mockito.internal.debugging.LocationFactory | 6e3846984f7d9362 |
| org.mockito.internal.debugging.LocationFactory.DefaultLocationFactory | 962892aa849afdfc |
| org.mockito.internal.debugging.LocationImpl | 530a4709e6947f60 |
| org.mockito.internal.debugging.LocationImpl.MetadataShim | e0e02617575176a4 |
| org.mockito.internal.exceptions.stacktrace.ConditionalStackTraceFilter | 3ae97774773f8cc1 |
| org.mockito.internal.exceptions.stacktrace.DefaultStackTraceCleaner | ccefda75b25508d |
| org.mockito.internal.exceptions.stacktrace.DefaultStackTraceCleanerProvider | b96ca03f68c6b0bc |
| org.mockito.internal.exceptions.stacktrace.StackTraceFilter | f1e5dd123e555b09 |
| org.mockito.internal.handler.InvocationNotifierHandler | 81a88d2a9823ca2e |
| org.mockito.internal.handler.MockHandlerFactory | 60aaf611c9f037ba |
| org.mockito.internal.handler.MockHandlerImpl | 40af730c41726d19 |
| org.mockito.internal.handler.NullResultGuardian | 887855f598dc7f26 |
| org.mockito.internal.invocation.ArgumentsProcessor | 48a63d334fbe1568 |
| org.mockito.internal.invocation.DefaultInvocationFactory | fd7a2f1ca0abf244 |
| org.mockito.internal.invocation.InterceptedInvocation | 2bc1759562590122 |
| org.mockito.internal.invocation.InterceptedInvocation.1 | a808ee7e12b0c370 |
| org.mockito.internal.invocation.InvocationMarker | f36ccd569efd70f7 |
| org.mockito.internal.invocation.InvocationMatcher | a60a277cd788c00 |
| org.mockito.internal.invocation.InvocationMatcher.1 | d5c4b96c4388ebd1 |
| org.mockito.internal.invocation.InvocationsFinder | d125e72698a81c75 |
| org.mockito.internal.invocation.MatcherApplicationStrategy | e8e6c5931cd35a1f |
| org.mockito.internal.invocation.MatchersBinder | 7855054a8c7718ee |
| org.mockito.internal.invocation.RealMethod.isIllegal | 64f3caff97edef01 |
| org.mockito.internal.invocation.StubInfoImpl | 9766984c92e9959b |
| org.mockito.internal.invocation.TypeSafeMatching | db8fac8befb40512 |
| org.mockito.internal.invocation.mockref.MockWeakReference | 1fbf38ee01ef223b |
| org.mockito.internal.listeners.StubbingLookupNotifier | 6f87fdb14780b091 |
| org.mockito.internal.listeners.VerificationStartedNotifier | bc97b666d47bf342 |
| org.mockito.internal.matchers.Any | ddca80baaf883a0c |
| org.mockito.internal.matchers.ArrayEquals | 27688bbffae1861 |
| org.mockito.internal.matchers.Equality | 341b019eaeb85d40 |
| org.mockito.internal.matchers.Equals | 84c89c09d537ae1d |
| org.mockito.internal.matchers.InstanceOf | 5c0eab070e987a7e |
| org.mockito.internal.matchers.LocalizedMatcher | f3da081806496e9c |
| org.mockito.internal.progress.ArgumentParserStorageImpl | d9b8becac423331b |
| org.mockito.internal.progress.MockingProgressImpl | b4b478523e99786f |
| org.mockito.internal.progress.MockingProgressImpl.1 | 9f7db825fdcdf194 |
| org.mockito.internal.progress.SequenceNumber | a68ee1dd45f51b97 |
| org.mockito.internal.progress.ThreadSafeMockingProgress | 452aa6e38ddf43e |
| org.mockito.internal.progress.ThreadSafeMockingProgress.1 | 79ae9726492f0c4f |
| org.mockito.internal.stubbing.BaseStubbing | 7fb9abb0c3eadb7f |
| org.mockito.internal.stubbing.ConsecutiveStubbing | 557234368bf5ca41 |
| org.mockito.internal.stubbing.DoAnswerStyleStubbing | 6e7ca0308caa0784 |
| org.mockito.internal.stubbing.InvocationContainerImpl | 6fc98009157aaa61 |
| org.mockito.internal.stubbing.OngoingStubbingImpl | 747b28f7f0499aba |
| org.mockito.internal.stubbing.StubbedInvocationMatcher | 73693c29bdb5f1d4 |
| org.mockito.internal.stubbing.StubberImpl | 0598892699f1408c |
| org.mockito.internal.stubbing.answers.AbstractThrowsException | 4d8774faf570ee8b |
| org.mockito.internal.stubbing.answers.CallsRealMethods | e57edbc58b0e39e6 |
| org.mockito.internal.stubbing.answers.DefaultAnswerValidator | bc157688cbf26d9c |
| org.mockito.internal.stubbing.answers.DoesNothing | fe0bc81f8af9ad94 |
| org.mockito.internal.stubbing.answers.InvocationInfo | 35c4a7d4431e2604 |
| org.mockito.internal.stubbing.answers.Returns | 419d0de7c8cd9ec2 |
| org.mockito.internal.stubbing.answers.ThrowsException | d5453a6546fff054 |
| org.mockito.internal.stubbing.answers.ThrowsExceptionForClassType | c40bbb544982fb63 |
| org.mockito.internal.stubbing.defaultanswers.GloballyConfiguredAnswer | b4af5d0cc4127c43 |
| org.mockito.internal.stubbing.defaultanswers.ReturnsDeepStubs | 9290a19f5dbdf1b2 |
| org.mockito.internal.stubbing.defaultanswers.ReturnsEmptyValues | dbc025659648ed02 |
| org.mockito.internal.stubbing.defaultanswers.ReturnsMocks | 99d9220ab6ee9e86 |
| org.mockito.internal.stubbing.defaultanswers.ReturnsMoreEmptyValues | 708bd411a28382b5 |
| org.mockito.internal.stubbing.defaultanswers.ReturnsSmartNulls | f434f2f732e6e80e |
| org.mockito.internal.stubbing.defaultanswers.TriesToReturnSelf | 13e6f22c3923267d |
| org.mockito.internal.util.Checks | 693b7ec3dc9db88a |

| | |
|---|-------------------|
| org.mockito.internal.util.ConsoleMockitoLogger | f6ec54a756328702 |
| org.mockito.internal.util.DefaultMockingDetails | 445b7a7104a677ce |
| org.mockito.internal.util.KotlinInlineClassUtil | 8f2e65801baf9ad5 |
| org.mockito.internal.util.MockCreationValidator | 83a10f2760252cf3 |
| org.mockito.internal.util.MockNameImpl | 009cd5fc276ed0dd |
| org.mockito.internal.util.MockUtil | 8cedd1d6aa623c6c |
| org.mockito.internal.util.ObjectMethodsGuru | e958146f93547352 |
| org.mockito.internal.util.Platform | 6c061a17b8fd556f |
| org.mockito.internal.util.Primitives | 6b6a08aaaf147839f |
| org.mockito.internal.util.collections.Iterables | f7eb3a38de601237 |
| org.mockito.internal.util.concurrent.DetachedThreadLocal | 24c845c0cee0c23b |
| org.mockito.internal.util.concurrent.DetachedThreadLocal.1 | defaf890898faa64 |
| org.mockito.internal.util.concurrent.DetachedThreadLocal.Cleaner | fe82f09eda153c82 |
| org.mockito.internal.util.concurrent.WeakConcurrentMap | 317df0cbc9bf65e4 |
| org.mockito.internal.util.concurrent.WeakConcurrentMap.LatentKey | 49d0008ff01c2270 |
| org.mockito.internal.util.concurrent.WeakConcurrentMap.WeakKey | c9b8ab481aee9c32 |
| org.mockito.internal.util.concurrent.WeakConcurrentMap.WithInlinedExpunction | 2900bb8f66594337 |
| org.mockito.internal.util.concurrent.WeakConcurrentSet | 01665a2956990716 |
| org.mockito.internal.util.concurrent.WeakConcurrentSet.Cleaner | 8e47207f365780a7 |
| org.mockito.internal.util.reflection.GenericMetadataSupport | 9acdffd63f655b49 |
| org.mockito.internal.util.reflection.GenericMetadataSupport.FromClassGenericMetadataSupport | 99c88b2ac93b8f3f |
| org.mockito.internal.util.reflection.GenericMetadataSupport.NotGenericReturnTypeSupport | 8c611ef213f94120 |
| org.mockito.internal.util.reflection.GenericMetadataSupport.ParameterizedReturnType | 25e3bc9dabf2fa9d |
| org.mockito.internal.util.reflection.GenericMetadataSupport.TypeVarBoundedType | b7f4d58732c20401 |
| org.mockito.internal.util.reflection.GenericMetadataSupport.TypeVariableReturnType | 6d8641bfb4e9554b |
| org.mockito.internal.util.reflection.InstrumentationMemberAccessor | 00b247ec3e952710 |
| org.mockito.internal.util.reflection.ModuleMemberAccessor | df6459cdb157634f |
| org.mockito.internalverification.AtLeast | 742247f606a7d3c4 |
| org.mockito.internalverification.DefaultRegisteredInvocations | 48184ff2108397ec |
| org.mockito.internalverification.MockAwareVerificationMode | 140035dc62d048aa |
| org.mockito.internalverification.Times | ce9c3ea7bcd93e4 |
| org.mockito.internalverification.VerificationDataImpl | 960bc0e7bc4cb209 |
| org.mockito.internalverification.VerificationEventImpl | c5d4f54b84a790e3 |
| org.mockito.internalverification.VerificationModeFactory | 9d00ab6e5382924b |
| org.mockito.internalverification.checkers.AtLeastXNumberofInvocationsChecker | 129db1498c6df606 |
| org.mockito.internalverification.checkers.MissingInvocationChecker | bc21ba1054e605b2 |
| org.mockito.internalverification.checkers.NumberOfInvocationsChecker | bd3469583e9e6716 |
| org.mockito.mock.MockType | 0b5105452bbd0790 |
| org.mockito.mock.SerializableMode | d8db118920e53367 |
| org.mockito.plugins.DoNotMockEnforcer | d84a80636a3b2091 |
| org.mockito.plugins.DoNotMockEnforcer.Cache | 7ad724250d9a8fd2 |
| org.objenesis.ObjenesisBase | 0c1d2fd83029257f |
| org.objenesis.ObjenesisStd | f35c83a75caea811 |
| org.objenesis.instantiator.sun.SunReflectionFactoryHelper | d17e7b3403696605 |
| org.objenesis.instantiator.sun.SunReflectionFactoryInstantiator | 6156947e7d7c507c |
| org.objenesis.strategy.BaselineInstantiatorStrategy | b0aaa6460452f5ce |
| org.objenesis.strategy.PlatformDescription | c6456f671febfd7c |
| org.objenesis.strategy.StdInstantiatorStrategy | abae05ba56ea35a6 |
| org.slf4j.LoggerFactory | d4f8bf028cb667a7 |
| org.slf4j.helpers.AbstractLogger | 0927772f80afa51d |
| org.slf4j.helpers.BasicMDCAdapter | 354fafb117483fdb |
| org.slf4j.helpers.BasicMDCAdapter.1 | 8f0671fb507009fb |
| org.slf4j.helpers.BasicMarkerFactory | d8e0b7e9d11b515c |
| org.slf4j.helpers.LegacyAbstractLogger | 9c0bab469712e43b |
| org.slf4j.helpers.NOPLoggerFactory | eaf704972ef7000c |
| org.slf4j.helpers.NOPMDCAdapter | d816a97d0b663014 |
| org.slf4j.helpers.NOP_FallbackServiceProvider | 44c4aa253bad3620 |
| org.slf4j.helpers.Reporter | c9b912a7116daa87 |
| org.slf4j.helpers.Reporter.Level | 07530b930aa1c996 |
| org.slf4j.helpers.Reporter.TargetChoice | 0aa347cd82827a6b |
| org.slf4j.helpers.SubstituteLoggerFactory | 2c5fb1b0f92b644d |
| org.slf4j.helpers.SubstituteServiceProvider | 1caf06178d203dfd |
| org.slf4j.helpers.ThreadLocalMapOfStacks | 2b24a935616f8730 |
| org.slf4j.helpers.Util | 859d67cf0632e467 |
| org.slf4j.simple.OutputChoice | 1210473ae6c23a02 |
| org.slf4j.simple.OutputChoice.OutputChoiceType | b1a3560a9741f5b4 |
| org.slf4j.simple.SimpleLogger | 1d2d4d9ea2830f3d |
| org.slf4j.simple.SimpleLoggerConfiguration | 1f2875dbbf5941d4 |
| org.slf4j.simple.SimpleLoggerFactory | e94c2f64e0e452e9 |
| org.slf4j.simple.SimpleServiceProvider | 99b66f9433a7c345 |
| org.sqlite.ExtendedCommand | c50b1f7fa92b5727 |
| org.sqlite.JDBC | 8e6db90d48c26646 |
| org.sqlite.SQLiteConfig | e5bc293c5fc5a401 |
| org.sqlite.SQLiteConfig.DateClass | d291e525fe22bbab |

| | |
|---|-------------------|
| org.sqlite.SQLiteConfig.DatePrecision | 48b2cfccf8b6cbcbf |
| org.sqlite.SQLiteConfig.Encoding | 3da2a60c689ecb45 |
| org.sqlite.SQLiteConfig.HexKeyMode | b96f57e39f13322c |
| org.sqlite.SQLiteConfig.JournalMode | 35850632da3b340e |
| org.sqlite.SQLiteConfig.LockingMode | e9325b088d34aed6 |
| org.sqlite.SQLiteConfig.Pragma | 87df8fb81c84ddba |
| org.sqlite.SQLiteConfig.SynchronousMode | a29bf5ffa16c164c |
| org.sqlite.SQLiteConfig.TempStore | b1b405b6ad7dee3c |
| org.sqlite.SQLiteConfig.TransactionMode | 2f404b6104bcee7c |
| org.sqlite.SQLiteConnection | ed855927d8eeb04b |
| org.sqlite.SQLiteConnectionConfig | bcd564a9722e9f02 |
| org.sqlite.SQLiteErrorCode | 5abe51ede2cd0696 |
| org.sqlite.SQLiteException | 55db16725b912d5c |
| org.sqlite.SQLiteJDBCLoader | a6594526880a50cc |
| org.sqlite.SQLiteJDBCLoader.VersionHolder | 9280009235d99717 |
| org.sqlite.SQLiteLimits | 5de2416ef4a27971 |
| org.sqlite.SQLiteOpenMode | 8359ea3e35c0f022 |
| org.sqlite.core.CoreDatabaseMetaData | 7051aebb4552574e |
| org.sqlite.core.CorePreparedStatement | 750078529300087e |
| org.sqlite.core.CoreResultSet | 1acb626cccfd5d |
| org.sqlite.core.CoreStatement | a1ceabd06ac323b0 |
| org.sqlite.core.DB | b6fc4681448d547c |
| org.sqlite.core.NativeDB | 500d057ceb6ec321 |
| org.sqlite.core.SafeStmtPtr | e50aa62317dfce9c |
| org.sqlite.date.FastDateFormat | 452c95a00328b9cc |
| org.sqlite.date.FastDateFormat.1 | 46fbf48faf2148ae |
| org.sqlite.date.FastDateParser | fae281e33d96261d |
| org.sqlite.date.FastDateParser.1 | ae45c588680f9086 |
| org.sqlite.date.FastDateParser.2 | 13d88cc5c2eb8734 |
| org.sqlite.date.FastDateParser.3 | 4cc4a0b1ba9c0895 |
| org.sqlite.date.FastDateParser.4 | e0ed4ea7b45392ea |
| org.sqlite.date.FastDateParser.CopyQuotedStrategy | edf865cc9ad02d15 |
| org.sqlite.date.FastDateParser.ISO8601TimeZoneStrategy | 80f87e347be6d3f2 |
| org.sqlite.date.FastDateParser.NumberStrategy | c35d8b87ff103836 |
| org.sqlite.date.FastDateParser.Strategy | 3aff03d37d59cc72 |
| org.sqlite.date.FastDatePrinter | aadc011f46d3de8a |
| org.sqlite.date.FastDatePrinter.CharacterLiteral | 84f3f16ae96b7cb8 |
| org.sqlite.date.FastDatePrinter.PaddedNumberField | 81a67ba8156f5efa |
| org.sqlite.date.FastDatePrinter.TwoDigitMonthField | 86f24deb0f79ff8e |
| org.sqlite.date.FastDatePrinter.TwoDigitNumberField | a85da15723b8b680 |
| org.sqlite.date.FormatCache | 9860c57b5c859df9 |
| org.sqlite.date.FormatCache.MultipartKey | 469164d2c8315f67 |
| org.sqlite.jdbc3.JDBC3Connection | 5e27d2f7139a637b |
| org.sqlite.jdbc3.JDBC3DatabaseMetaData | 5bf62e00b2309290 |
| org.sqlite.jdbc3.JDBC3PreparedStatement | 612ff277d1eb1a39 |
| org.sqlite.jdbc3.JDBC3ResultSet | 904eb93adc35ae17 |
| org.sqlite.jdbc3.JDBC3Statement | 74fa6af2dbbcf659 |
| org.sqlite.jdbc4.JDBC4Connection | a06b30caeb027651 |
| org.sqlite.jdbc4.JDBC4DatabaseMetaData | f7bac5219bb5b0af |
| org.sqlite.jdbc4.JDBC4PreparedStatement | 2e47f390feb2663e |
| org.sqlite.jdbc4.JDBC4ResultSet | bade386a08cd541c |
| org.sqlite.jdbc4.JDBC4Statement | 5abde92f58724875 |
| org.sqlite.util.LibraryLoaderUtil | 4df105564098d269 |
| org.sqlite.util.OSInfo | 50a8b6f7f0e697a7 |
| org.sqlite.util.ProcessRunner | eab1d4463e729110 |
| sun.util.resources.cldr.provider.CLDRLocaleDataMetaInfo | cea799461486d92b |

GameScreenController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| loadSidebar() | | 0% | | n/a | 1 | 1 | 9 | 9 | 1 | 1 |
| GameScreenController() | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| Total | 32 of 40 | 20% | 0 of 0 | n/a | 1 | 2 | 9 | 11 | 1 | 2 |

Created with JaCoCo 0.8.12.202403310830

GameScreenController.java

```
1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.services.Logger;
4. import com.example.project.services.SceneManager;
5. import com.example.project.services.Session;
6. import javafx.fxml.FXMLLoader;
7. import javafx.scene.Parent;
8.
9. import java.io.IOException;
10.
11. /**
12. * Game screen controller that has some startup on screen thing todo. Level, shop controllers..
13. */
14. public abstract class GameScreenController
15. {
16.     protected Logger logger = new Logger();
17.
18.     protected GameScreenController() {}
19.
20.     /**
21.      * Called when the game scene is changed to one of the GameScenes Login, Level, Shop etc.
22.      */
23.     public abstract void onSceneChangedToThis();
24.
25.     /**
26.      * create models and inject session and scene manager.
27.      * @param session session.
28.      * @param sceneManager scene manager.
29.      */
30.     public abstract void setup(Session session, SceneManager sceneManager);
31.
32.     protected record SidebarLoaded(Parent node, SidebarController controller) {}
33.
34.     protected SidebarLoaded loadSidebar()
35.     {
36.         var sidebarPath = "/com/example/project/gameScreens/sidebar.fxml";
37.         FXMLLoader loader = new FXMLLoader(getClass().getResource(sidebarPath));
38.         Parent sidebarNode = null;
39.         try{
40.             sidebarNode = loader.load();
41.         }
42.         catch (IOException exception){
43.             this.logger.logError("failed to load sidebar fxml file.");
44.         }
45.
46.         SidebarLoaded side = new SidebarLoaded(sidebarNode, loader.getController());
47.         return side;
48.     }
49. }
```

GameScreenFactory

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed Lines | Missed Methods |
|---|---------------------|------|-----------------|------|--------|------|--------------|----------------|
| preloadPage(GameScene, String, PageLoader) | | 0% | | 0% | 3 | 3 | 15 | 15 |
| loadPages(PageLoader) | | 0% | | n/a | 1 | 1 | 6 | 6 |
| loadGameScreens(RootLayoutController, FXMLPageLoader) | | 0% | | n/a | 1 | 1 | 5 | 5 |
| GameScreenFactory(Session) | | 0% | | n/a | 1 | 1 | 5 | 5 |
| Total | 154 of 154 | 0% | 4 of 4 | 0% | 6 | 6 | 31 | 31 |

Created with JaCoCo 0.8.12.202403310830

ConfettiAnimation

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|-------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| playConfetti(Pane) | | 0% | | 0% | 2 | 2 | 14 | 14 | 1 | 1 |
| ConfettiAnimation() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 81 of 81 | 0% | 2 of 2 | 0% | 3 | 3 | 15 | 15 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

GameScreenFactory.java

```

1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.controllers.RootLayoutController;
4. import com.example.project.services.*;
5. import javafx.scene.Parent;
6.
7. import java.io.IOException;
8. import java.util.HashMap;
9. import java.util.Map;
10.
11. /**
12. * create game screen controllers.
13. */
14. public class GameScreenFactory {
15.
16.     private final Session session;
17.     private SceneManager sceneManager;
18.     private final Map<GameScene, Parent> pages = new HashMap<>();
19.     private final Map<GameScene, GameScreenController> controllers = new HashMap<>();
20.
21.     /**
22.      * Constructor.
23.      * @param session session.
24.      */
25.     public GameScreenFactory(Session session) {
26.         this.session = session;
27.     }
28.
29.
30.     /**
31.      * Load all main game screens.
32.      * @param rootController root.
33.      * @param loader loader.
34.      */
35.     public void loadGameScreens(RootLayoutController rootController, FXMLLoaderPageLoader loader)
36.     {
37.         this.sceneManager = new SceneManager(rootController, controllers, pages);
38.         rootController.setSceneManager(this.sceneManager);
39.         this.loadPages(loader);
40.         sceneManager.switchScene(GameScene.LOGIN);
41.     }
42.
43.     private void loadPages(PageLoader loader)
44.     {
45.         preloadPage(GameScene.LOGIN, "/com/example/project/gameScreens/login-view.fxml", loader);
46.         preloadPage(GameScene.MAINMENU, "/com/example/project/gameScreens/main-menu-view.fxml", loader);
47.         preloadPage(GameScene.DAILY_REWARD, "/com/example/project/gameScreens/dailyReward-view.fxml", loader);
48.         preloadPage(GameScene.LEVEL, "/com/example/project/gameScreens/level-view.fxml", loader);
49.         preloadPage(GameScene.SHOP, "/com/example/project/gameScreens/shop-view.fxml", loader);
50.     }
51.
52.     private void preloadPage(GameScene type, String fxmlPath, PageLoader loader)
53.     {
54.         Parent page;
55.         try{
56.             page = loader.load(fxmlPath);
57.         }
58.         catch (IOException e){
59.             throw new RuntimeException(String.format("Exception when loading page: %s. Message: %s, cause: %s", fxmlPath, e.getMessage(), e.getCause()));
60.         }
61.         catch (IllegalStateException e){
62.             throw new RuntimeException(String.format("game scene fxml path for %s not correct: %s, cause: %s", fxmlPath, e.getMessage(), e.getCause()));
63.         }
64.
65.         if (page == null)
66.         {
67.             throw new IllegalArgumentException("Page not loaded: " + type);
68.         }
69.
70.         GameScreenController controller = loader.getController();
71.         if (controller == null){
72.             throw new RuntimeException("must have a controller on the game screen that is a gameScreenController");
73.         }
74.
75.         controller.setup(this.session, this.sceneManager);
76.         controllers.put(type, controller);
77.         pages.put(type, page);
78.     }
79. }
```

LevelController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setup(Session_SceneManager) | | 0% | n/a | | 1 | 1 | 26 | 26 | 1 | 1 |
| lambda\$onPlayButton\$9(int, ActionEvent) | | 0% | | 0% | 8 | 8 | 25 | 25 | 1 | 1 |
| onSceneChangedToThis() | | 0% | n/a | | 1 | 1 | 12 | 12 | 1 | 1 |
| syncLevelWonText() | | 0% | | 0% | 3 | 3 | 16 | 16 | 1 | 1 |
| setupDefinitionPopup() | | 0% | n/a | | 1 | 1 | 11 | 11 | 1 | 1 |
| syncRedrawWindow() | | 0% | | 0% | 4 | 4 | 11 | 11 | 1 | 1 |
| onPlayButton() | | 0% | n/a | | 1 | 1 | 9 | 9 | 1 | 1 |
| syncDefinitionWindow(boolean) | | 0% | | 0% | 3 | 3 | 9 | 9 | 1 | 1 |
| lambda\$runTotalScoreAnimation\$10(Label, Paint, int, ActionEvent) | | 0% | n/a | | 1 | 1 | 10 | 10 | 1 | 1 |
| syncPlayButton() | | 0% | | 0% | 5 | 5 | 4 | 4 | 1 | 1 |
| runTotalScoreAnimation(int, int) | | 0% | n/a | | 1 | 1 | 7 | 7 | 1 | 1 |
| LevelController() | | 0% | n/a | | 1 | 1 | 5 | 5 | 1 | 1 |
| syncRedrawButton() | | 0% | | 0% | 3 | 3 | 4 | 4 | 1 | 1 |
| lambda\$setupDefinitionPopup\$6(ObservableValue, Boolean, Boolean) | | 0% | | 0% | 2 | 2 | 5 | 5 | 1 | 1 |
| onLetterTileClicked(LetterTileController) | | 0% | | 0% | 2 | 2 | 3 | 3 | 1 | 1 |
| lambda\$setupDefinitionPopup\$5(MouseEvent) | | 0% | | 0% | 2 | 2 | 4 | 4 | 1 | 1 |
| lambda\$onSceneChangedToThis\$4() | | 0% | | 0% | 2 | 2 | 1 | 1 | 1 | 1 |
| lambda\$onPlayButton\$8(int, int, ActionEvent) | | 0% | n/a | | 1 | 1 | 3 | 3 | 1 | 1 |
| syncConfirmRedrawButton() | | 0% | n/a | | 1 | 1 | 2 | 2 | 1 | 1 |
| lambda\$syncDefinitionWindow\$7(boolean, ActionEvent) | | 0% | | 0% | 2 | 2 | 3 | 3 | 1 | 1 |
| lambda\$setup\$3(ObservableValue, Boolean, Boolean) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$syncLevelWonText\$12(ActionEvent) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$syncLevelWonText\$11(ActionEvent) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$setup\$2(ObservableValue, Boolean, Boolean) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$setup\$1(ObservableValue, Number, Number) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$setup\$0(ObservableValue, Number, Number) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| onRedrawButton() | | 91% | | 50% | 1 | 2 | 0 | 3 | 0 | 1 |
| onConfirmRedrawButton() | | 91% | | 50% | 1 | 2 | 0 | 3 | 0 | 1 |
| LevelController(LevelModel) | | 100% | n/a | | 0 | 1 | 0 | 5 | 0 | 1 |
| onSkipButton() | | 100% | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 752 of 807 | 6% | 52 of 54 | 3% | 53 | 57 | 165 | 177 | 26 | 30 |

LevelController.java

```

1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.models.tileGroups.LetterTileGroup;
4. import com.example.project.models.tileGroups.UpgradeTileGroup;
5. import com.example.project.controllers.gameScreens.animations.LevelScoreSequence;
6. import com.example.project.controllers.gameScreens.animations.ScoreTimeline;
7. import com.example.project.controllers.gameScreens.animations.TextEmphasisAnimation;
8. import com.example.project.controllers.popupControllers.DefinitionController;
9. import com.example.project.controllers.tiles.LetterTileController;
10. import com.example.project.models.gameScreens.LevelModel;
11. import com.example.project.models.popups.DefinitionPopup;
12. import com.example.project.services.GameScene;
13. import com.example.project.services.PopupLoader;
14. import com.example.project.services.SceneManager;
15. import com.example.project.services.Session;
16. import com.example.project.services.sound.GameSoundPlayer;
17. import javafx.animation.*;
18. import javafx.beans.binding.Bindings;
19. import javafx.fxml.FXML;
20. import javafx.scene.Cursor;
21. import javafx.scene.control.Button;
22. import javafx.scene.control.Label;
23. import javafx.scene.layout.HBox;
24. import javafx.scene.layout.StackPane;
25. import javafx.scene.paint.Color;
26. import javafx.util.Duration;
27. import javafx.scene.layout.VBox;
28. import java.util.*;
29.
30.
31. /**
32. * Controller for the level screen.
33. */
34. public class LevelController extends GameScreenController
35. {
36.     @FXML Label levelWonLostText;
37.     @FXML private StackPane sidebar;
38.     @FXML HBox tileRackContainer;
39.     @FXML HBox wordWindowContainer;
40.     @FXML HBox upgradeTilesContainer;
41.     @FXML Button playButton;
42.     @FXML Button redrawButton;
43.     @FXML VBox redrawContainer;
44.     @FXML Button confirmRedrawButton;
45.     @FXML private StackPane root;
46.     @FXML private StackPane definitionContainer;
47.
48.     private static LevelModel levelModel;
49.     private final DefinitionPopup definitionPopup = new DefinitionPopup();
50.     private UpgradeTileGroup upgradeGroup;
51.     private LetterTileGroup tileRack;
52.     private LetterTileGroup wordTilesRow;
53.     private LetterTileGroup redrawTilesColumn;
54.     private SidebarController sidebarController;
55.
56.     private final GameSoundPlayer spillSound = new GameSoundPlayer("/com/example/project/Sounds/spill2.wav");
57.     private final GameSoundPlayer shakeSound = new GameSoundPlayer("/com/example/project/Sounds/shake2.wav");
58.     private final GameSoundPlayer wonSound = new GameSoundPlayer("/com/example/project/Sounds/win.wav");
59.
60.     /**
61.      * Constructor only called once each time application opened.
62.     */
63.     public LevelController() {}
64.
65.     protected LevelController(LevelModel model) { levelModel = model; }
66.
67.     @Override
68.     public void setup(Session session, SceneManager sceneManager)
69.     {
70.         levelModel = new LevelModel(session, sceneManager);
71.
72.         // Setup Listeners. (automatically updates each property when they're changed)
73.         levelModel.getSession().getCurrentRedraws().addListener((obs, oldVal, newVal) -> syncRedrawButton());
74.         levelModel.getSession().getCurrentPlays().addListener((obs, oldVal, newVal) -> syncPlayButton());
75.
76.         levelModel.getIsRedrawActive().addListener((obs, oldVal, newVal) -> syncRedrawWindow());
77.         definitionPopup.getIsDefinitionActive().addListener((obs, oldVal, newVal) -> syncDefinitionWindow(newVal));
78.
79.         tileRack = new LetterTileGroup(levelModel.getSession().getHandSize(), tileRackContainer,
80.                                         levelModel.getTileRackTilesProperty(), this::onLetterTileClicked);
81.
82.         wordTilesRow = new LetterTileGroup(levelModel.getSession().getWordWindowSize(), wordWindowContainer,
83.                                           levelModel.getWordWindowTilesProperty(), this::onLetterTileClicked,
84.                                           List.of(this::syncPlayButton));
85.
86.         redrawTilesColumn = new LetterTileGroup(levelModel.getSession().getWordWindowSize(), redrawContainer,
87.                                               levelModel.getRedrawWindowTilesProperty(), this::onLetterTileClicked,
88.                                               List.of(this::syncRedrawButton, this::syncConfirmRedrawButton));
89.
90.         upgradeGroup = new UpgradeTileGroup(upgradeTilesContainer, levelModel.getSession().getPlayersUpgradesProperty());
91.         upgradeGroup.syncTiles();
92.
93.         setupDefinitionPopup();
94.
95.         var loadedSidebar = this.loadSidebar();
96.         var sidebarNode = ((StackPane) loadedSidebar.node());
97.         this.sidebar.getChildren().add(sidebarNode);
98.         this.sidebarController = loadedSidebar.controller();
99.         sidebarController.setupProperties(levelModel);
100.
101.        // initial sync.
102.        syncPlayButton();
103.        syncRedrawButton();
104.        syncConfirmRedrawButton();
105.        syncRedrawWindow();
106.    }
107.
108.    @Override
109.    public void onSceneChangedToThis()
110.    {
111.        levelModel.setupNewLevel();
112.        this.logger.logMessage("level page loaded.");

```

```

113.     levelWonLostText.setText("");
114.
115.     tileRack.syncTiles();
116.     wordTilesRow.syncTiles();
117.     redrawTilesColumn.syncTiles();
118.     upgradeGroup.syncTiles();
119.
120.     // level outcome text banner will only be visible when there is text
121.     levelWonLostText.visibleProperty().bind(Bindings.createBooleanBinding(
122.         () -> !levelWonLostText.getText().isEmpty(), levelWonLostText.textProperty() ));
123.     levelWonLostText.managedProperty().bind(levelWonLostText.visibleProperty());
124.     levelWonLostText.prefWidthProperty().bind(levelWonLostText.widthProperty().add(500));
125. }
126.
127. private void setupDefinitionPopup()
128. {
129.     PopupLoader popupLoader = new PopupLoader();
130.     DefinitionController definitionController = popupLoader.createPopupController(definitionPopup);
131.     definitionContainer.setMouseTransparent(true);
132.     definitionContainer.setVisible(false);
133.
134.     var popupRoot = definitionController.getStack();
135.     definitionContainer.getChildren().add(popupRoot);
136.
137.     popupRoot.managedProperty().bind(definitionPopup.getIsDefinitionActive());
138.     definitionPopup.setIsDefinitionActive(false);
139.
140.     // Listener for auto hiding the popup when clicking the container area outside the popup's content
141.     definitionContainer.setOnMouseClicked(event -> {
142.         if (event.getTarget() == definitionContainer) {
143.             definitionPopup.setIsDefinitionActive(false);
144.         }
145.         event.consume();
146.     });
147.
148.     // Listener for changing cursor appearance anywhere on screen
149.     definitionPopup.getIsDefinitionActive().addListener((activeObs, wasActive, isActive) -> {
150.         if (isActive) {
151.             definitionContainer.setCursor(Cursor.HAND);
152.         } else {
153.             definitionContainer.setCursor(Cursor.DEFAULT);
154.             this.syncLevelWonText();
155.         }
156.     });
157. }
158.
159. private void syncRedrawWindow()
160. {
161.     var isRedrawActive = levelModel.getIsRedrawActive().get();
162.     var distance = isRedrawActive ? -50 : 300; // slide on if inactive. slide out if active.
163.     TranslateTransition redrawWindowSlide = new TranslateTransition(Duration.millis(500), redrawContainer);
164.     redrawWindowSlide.setToX(distance);
165.     redrawWindowSlide.play();
166.     syncRedrawButton();
167.     if (levelModel.getSceneManager().getCurrentScene() != GameScene.LEVEL){
168.         return;
169.     }
170.
171.     var soundToPlay = isRedrawActive ? shakeSound : spillSound;
172.     soundToPlay.replay();
173. }
174.
175. private void syncDefinitionWindow(boolean isDefinitionActive){
176.     if (isDefinitionActive) {
177.         definitionContainer.setVisible(true);
178.         definitionContainer.setMouseTransparent(false);
179.     }
180.     var distance = isDefinitionActive ? 300 : 1000; // slide on if inactive. slide out if active.
181.     TranslateTransition definitionWindowSlide = new TranslateTransition(Duration.millis(500), definitionContainer);
182.     definitionWindowSlide.setToX(distance);
183.     definitionWindowSlide.setOnFinished(e -> {
184.         if (!isDefinitionActive) {
185.             definitionContainer.setVisible(false);
186.         }
187.     });
188.     definitionWindowSlide.play();
189. }
190.
191.
192. private void syncRedrawButton()
193. {
194.     var redraws = levelModel.getSession().getCurrentRedraws().get();
195.     redrawButton.setDisable(redraws == 0);
196.     redrawButton.setText(levelModel.getIsRedrawActive().get() ? "Cancel" : "Redraw Letters");
197. }
198.
199.
200. private void syncPlayButton()
201. {
202.     var plays = levelModel.getSession().getCurrentPlays().get();
203.     playButton.setDisable(plays == 0 || !levelModel.isCurrentWordValid() || levelModel.getWordWindowTilesProperty().isEmpty() || levelModel.getIsRedrawActive().get());
204.     playButton.setText("Play Word");
205. }
206.
207. private void syncConfirmRedrawButton(){
208.     confirmRedrawButton.setDisable(levelModel.getRedrawWindowTilesProperty().isEmpty());
209. }
210.
211. /**
212. * Handle tile clicks.
213. */
214. private void onLetterTileClicked(LetterTileController tileController)
215. {
216.     boolean moved = levelModel.tryMoveTile(tileController.getModel());
217.     if (!moved) { this.logger.logMessage("Cannot move tile - no space available or tile not found."); }
218. }
219.
220. /**
221. * Handle play button
222. */
223. @FXML
224. protected void onPlayButton()
225. {
226.     definitionPopup.setPopup(levelModel.getCurrentWord());
227.     playButton.setDisable(true);
228.     var rawPoints = sidebarController.getRawPointsLabel();
229.     var multiplier = sidebarController.getMultiplierLabel();
230. }

```

```

231.     int startScore = levelModel.getPlayersTotalPoints().get();
232.     var tileScoringSequence = new LevelScoreSequence(wordTilesRow.getControllers(), levelModel, rawPoints, multiplier);
233.     tileScoringSequence.setOnFinished(e ->
234.     {
235.         // Capturing scores before upgrades run
236.         int preUpgradePoints = levelModel.wordPointsProperty().get();
237.         int preUpgradeMulti = levelModel.wordMultiProperty().get();
238.
239.         int endScore = startScore + levelModel.calcTotalWordScore(); // Upgrades run here
240.
241.         // Capturing scores after upgrades run
242.         int postUpgradePoints = levelModel.wordPointsProperty().get();
243.         int postUpgradeMulti = levelModel.wordMultiProperty().get();
244.
245.         // Check if any upgrade changed the points or multiplier
246.         boolean pointsChanged = preUpgradePoints != postUpgradePoints;
247.         boolean multiChanged = preUpgradeMulti != postUpgradeMulti;
248.         boolean upgradesApplied = pointsChanged || multiChanged;
249.
250.         SequentialTransition upgradeSequence = new SequentialTransition();
251.
252.         if (upgradesApplied) {
253.             if (pointsChanged) {
254.                 var pointsLabel = sidebarController.getRawPointsLabel();
255.                 var scoreInitialColour = pointsLabel.getTextFill();
256.                 TextEmphasisAnimation pointsEmphasis = new TextEmphasisAnimation(pointsLabel, scoreInitialColour, scoreInitialColour, Duration.seconds(0));
257.                 upgradeSequence.getChildren().addAll(pointsEmphasis.getChildren());
258.                 upgradeSequence.getChildren().add(new javafx.animation.PauseTransition(Duration.millis(1)));
259.             }
260.             if (multiChanged) {
261.                 var comboMultiplierLabel = sidebarController.getComboLabel();
262.                 var multiColour = comboMultiplierLabel.getTextFill();
263.                 TextEmphasisAnimation multiEmphasis = new TextEmphasisAnimation(comboMultiplierLabel, multiColour, multiColour, Duration.seconds(0));
264.                 upgradeSequence.getChildren().addAll(multiEmphasis.getChildren());
265.             }
266.
267.             upgradeSequence.setOnFinished(g -> {
268.                 levelModel.getFileScoreSoundPlayer().playNextNote();
269.                 runTotalScoreAnimation(startScore, endScore);
270.             });
271.
272.             upgradeSequence.play();
273.         } else {
274.             // If no upgrades changed the score, skip the upgrade animation and run total score immediately
275.             runTotalScoreAnimation(startScore, endScore);
276.         }
277.     });
278.
279.     tileScoringSequence.play();
280. }
281.
282. private void runTotalScoreAnimation(int startScore, int endScore) {
283.     ScoreTimeline totalScoreTimeline = new ScoreTimeline();
284.     var currentScoreLabel = sidebarController.getCurrentScoreLabel();
285.     var scoreInitialColour = currentScoreLabel.getTextFill();
286.     Timeline timeline = totalScoreTimeline.animateTotalScore(startScore, endScore, currentScoreLabel, 1000);
287.     timeline.setOnFinished(f ->
288.     {
289.         TextEmphasisAnimation scoreEmphasis = new TextEmphasisAnimation(currentScoreLabel, scoreInitialColour, scoreInitialColour, Duration.seconds(0));
290.         scoreEmphasis.play();
291.         levelModel.getFileScoreSoundPlayer().playNextNote();
292.         playButton.setDisable(false);
293.         levelModel.playTiles();
294.         levelModel.resetCombo();
295.         levelModel.setTotalScore(endScore);
296.         levelModel.getFileScoreSoundPlayer().reset();
297.         definitionPopup.setIsDefinitionActive(true);
298.     });
299.     timeline.play();
300. }
301.
302. private void syncLevelWonText()
303. {
304.     if (levelModel.hasWon()){
305.         levelWonLostText.setText("VICTORY ACHIEVED");
306.         TextEmphasisAnimation youWonSequence = new TextEmphasisAnimation(levelWonLostText, Color.YELLOW, Color.BLACK, Duration.seconds(1));
307.         wonSound.replay();
308.         youWonSequence.setOnFinished(e -> levelModel.onWonLevel());
309.         youWonSequence.play();
310.     }
311.     else if (levelModel.hasLost()){
312.         levelWonLostText.setText("YOU DIED");
313.         TextEmphasisAnimation animSequence =
314.             new TextEmphasisAnimation(levelWonLostText, Color.RED, Color.BLACK, Duration.seconds(1));
315.         animSequence.setOnFinished(e -> levelModel.onLostLevel());
316.         animSequence.play();
317.     }
318.     else{
319.         levelWonLostText.setText("");
320.     }
321. }
322.
323. @FXML
324. protected void onSkipButton() { levelModel.getSceneManager().switchScene(GameScene.SHOP); }
325.
326. /**
327. * redraw button opens or cancels the redraw.
328. */
329. @FXML
330. protected void onRedrawButton()
331. {
332.     levelModel.setIsRedrawActive(!levelModel.getIsRedrawActive().get());
333.     levelModel.returnRedrawTilesToTheRack();
334. }
335.
336. /**
337. * Handle redraw confirm button.
338. */
339. @FXML
340. protected void onConfirmRedrawButton() {
341.     levelModel.setIsRedrawActive(!levelModel.getIsRedrawActive().get());
342.     levelModel.redrawTiles();
343. }
344. }

```

LoginController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setup(Session, SceneManager) | | 0% | | 0% | 3 | 3 | 17 | 17 | 1 | 1 |
| onSceneChangedToThis() | | 0% | n/a | | 1 | 1 | 3 | 3 | 1 | 1 |
| onLoginButtonClick() | | 0% | n/a | | 1 | 1 | 2 | 2 | 1 | 1 |
| onSignupButtonClick() | | 0% | n/a | | 1 | 1 | 2 | 2 | 1 | 1 |
| LoginController() | | 0% | n/a | | 1 | 1 | 2 | 2 | 1 | 1 |
| skipLogin() | | 0% | n/a | | 1 | 1 | 2 | 2 | 1 | 1 |
| lambda\$onSceneChangedToThis\$0(ObservableValue, String, String) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 167 of 167 | 0% | 4 of 4 | 0% | 9 | 9 | 29 | 29 | 7 | 7 |

Created with JaCoCo 0.8.12.202403310830

LoginController.java

```
1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation;
4. import com.example.project.models.gameScreens.LoginModel;
5. import com.example.project.models.tiles.LetterTileModel;
6. import com.example.project.services.GameScene;
7. import com.example.project.services.PasswordHasher;
8. import com.example.project.controllers.tiles.TileControllerFactory;
9. import com.example.project.services.SceneManager;
10. import com.example.project.services.sqlite.dAOs.UsersDAO;
11. import com.example.project.services.Session;
12. import javafx.fxml.FXML;
13. import javafx.scene.control.Label;
14. import javafx.scene.control.TextField;
15. import javafx.scene.image.Image;
16. import javafx.scene.image.ImageView;
17. import javafx.scene.layout.Pane;
18.
19. import java.util.ArrayList;
20. import java.util.List;
21. import java.util.Objects;
22.
23. /**
24. * Controller for the login scene.
25. */
26. public class LoginController extends GameScreenController
27. {
28.     @FXML
29.     Pane root;
30.
31.     @FXML
32.     Pane titleRow;
33.
34.     @FXML
35.     ImageView imageBG;
36.
37.     @FXML
38.     Pane backgroundContainer;
39.
40.     @FXML
41.     private Label infoText;
42.
43.     @FXML
44.     private TextField usernameTextField;
45.
46.     @FXML
47.     private TextField passwordTextField;
48.
49.     private LoginModel loginModel;
50.     private final TileControllerFactory tileControllerFactory = new TileControllerFactory();
51.
52.     /**
53.      * No arg constructor.
54.     */
55.     public LoginController() { }
56.
57.     @Override
58.     public void onSceneChangedToThis()
59.     {
60.         this.logger.logMessage("Login page loaded.");
61.
62.         loginModel.getWelcomeTextProperty().addListener((obs, oldVal, newVal) ->
63.             this.infoText.setText(newVal));
64.     }
65.
66.     @Override
67.     public void setup(Session session, SceneManager sceneManager)
68.     {
69.         this.loginModel = new LoginModel(session, sceneManager, new UsersDAO(), new PasswordHasher());
70.
71.         var newIm = new Image(Objects.requireNonNull(getClass().getResource("/com/example/project/gameScreens/loginBgImage.jpg")).toExternalForm());
72.         imageBG.setImage(newIm);
73.         imageBG.fitWidthProperty().bind(backgroundContainer.widthProperty());
74.         imageBG.fitHeightProperty().bind(backgroundContainer.heightProperty());
75.
76.         List<LetterTileModel> lettersInWordPlayWord = new ArrayList<>();
77.         var letters = List.of('w', 'o', 'r', 'd', 'p', 'l', 'a', 'y');
78.         for (char letter : letters){
79.             lettersInWordPlayWord.add(new LetterTileModel(letter));
80.         }
81.
82.         for (var t : lettersInWordPlayWord){
83.             var ltController = tileControllerFactory.createLetterTileController(t);
84.             titleRow.getChildren().add(ltController.getRoot());
85.             InfiniteFloatingAnimation fa = new InfiniteFloatingAnimation();
86.             fa.apply(ltController.getRoot(), 4);
87.         }
88.     }
89.
90.     @FXML
91.     protected void onLoginButtonClick()
92.     {
93.         loginModel.onLoginClicked(usernameTextField.getText(), passwordTextField.getText());
94.     }
95.
```

```
96.     @FXML
97.     protected void onSignupButtonClick()
98.     {
99.         this.loginModel.onSignUpClicked(usernameTextField.getText(), passwordTextField.getText());
100.    }
101.
102.    // TODO: remove later
103.    @FXML
104.    private void skipLogin()
105.    {
106.        loginModel.getSceneManager().switchScene(GameScene.LEVEL);
107.    }
108.}
```

Created with JaCoCo 0.8.12.202403310830

MainMenuController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setup(Session, SceneManager) | | 0% | | 0% | 3 | 3 | 20 | 20 | 1 | 1 |
| onSceneChangedToThis() | | 0% | | 0% | 5 | 5 | 9 | 9 | 1 | 1 |
| onFastForwardClick() | | 0% | | n/a | 1 | 1 | 6 | 6 | 1 | 1 |
| MainMenuController() | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| updateDailyRewardUI() | | 0% | | 0% | 2 | 2 | 2 | 2 | 1 | 1 |
| onLoadButtonClick() | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| onDailyRewardClicked() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| onStartButtonClick() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| onLogoutButtonClick() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| Total | 240 of 240 | 0% | 14 of 14 | 0% | 16 | 16 | 48 | 48 | 9 | 9 |

Created with JaCoCo 0.8.12.202403310830

MainMenuController.java

```

1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation;
4. import com.example.project.controllers.tiles.TileControllerFactory;
5. import com.example.project.models.gameScreens.MainMenuModel;
6. import com.example.project.models.tiles.LetterTileModel;
7. import com.example.project.services.SceneManager;
8. import com.example.project.services.Session;
9. import com.example.project.services.sqlite.dAOs.UsersDAO;
10. import javafx.fxml.FXML;
11. import javafx.scene.image.Image;
12. import javafx.scene.image.ImageView;
13. import javafx.scene.layout.Pane;
14. import java.util.ArrayList;
15. import java.util.List;
16. import java.util.Objects;
17. import com.example.project.services.GameScene;
18. import javafx.scene.control.Button;
19. import javafx.scene.layout.VBox;
20.
21. import java.time.LocalDate;
22.
23. /**
24. * controller for the main menu.
25. */
26. public class MainMenuController extends GameScreenController
27. {
28.
29.     @FXML
30.     Pane root;
31.
32.     @FXML
33.     Pane titleRow;
34.
35.     @FXML
36.     ImageView imageBG;
37.
38.     @FXML
39.     Pane backgroundContainer;
40.
41.     @FXML private Button dailyRewardButton;
42.     @FXML private Button loadButton;
43.
44.     private MainMenuModel mainMenuModel;
45.     private final TileControllerFactory tileControllerFactory = new TileControllerFactory();
46.     private final UsersDAO usersDAO = new UsersDAO();
47.
48. /**
49. * No arg constructor.
50. */
51. public MainMenuController() {}
52.
53. private VBox loadButtonParent;
54.
55. @Override
56. public void onSceneChangedToThis()
57. {
58.     var haveData = this.usersDAO.hasSaveData(this.mainMenuModel.getSession().getUser());
59.     var containsButton = loadButtonParent.getChildrenUnmodifiable().contains(loadButton);
60.     ◆◆ var hideButton = !haveData && containsButton;
61.     ◆◆ if (hideButton) {
62.         loadButtonParent.getChildren().remove(loadButton);
63.     }
64.     ◆◆ else if (!containsButton) {
65.         loadButtonParent.getChildren().add(loadButton);
66.     }
67.
68.     updateDailyRewardUI();
69. }
70.
71. @Override
72. public void setup(Session session, SceneManager sceneManager)
73. {
74.     loadButtonParent = (VBox) this.loadButton.getParent();
75.
76.     this.mainMenuModel = new MainMenuModel(session, sceneManager);
77.
78.     var newIm = new Image(Objects.requireNonNull(getClass().getResource("/com/example/project/gameScreens/loginBgImage.jpg")).toExternalForm());
79.     imageBG.setImage(newIm);
80.     imageBG.fitWidthProperty().bind(backgroundContainer.widthProperty());
81.     imageBG.fitHeightProperty().bind(backgroundContainer.heightProperty());
82.
83.     List<LetterTileModel> lettersInWordPlayWord = new ArrayList<>();
84.     var letters = List.of('w', 'o', 'r', 'd', 'p', 'l', 'a', 'y');
85.     ◆◆ for (char letter : letters){
86.         lettersInWordPlayWord.add(new LetterTileModel(letter));
87.     }
88.
89.     ◆◆ for (var t : lettersInWordPlayWord){
90.         var ltController = tileControllerFactory.createLetterTileController(t);
91.         titleRow.getChildren().add(ltController.getRoot());
92.         InfiniteFloatingAnimation fa = new InfiniteFloatingAnimation();
93.         fa.apply(ltController.getRoot(), 4);
94.     }
95.

```

```

96.     TooltipSetup tooltip = new TooltipSetup();
97.     tooltip.setupTooltip(dailyRewardButton, TooltipSetup.Element.DAILY, mainMenuModel);
98. }
99.
100.
101. @FXML
102. protected void onStartButtonClick()
103. {
104.     mainMenuModel.onStartClicked();
105.
106.     @FXML
107.     protected void onLoadButtonClick()
108.     {
109.         this.mainMenuModel.getSession().load();
110.         this.mainMenuModel.getSceneManager().switchScene(GameScene.LEVEL);
111.     }
112.
113.     @FXML
114.     protected void onLogoutButtonClick()
115.     {
116.         mainMenuModel.onLogoutClicked();
117.     }
118.
119.
120.     @FXML
121.     protected void onDailyRewardClicked() { mainMenuModel.getSceneManager().switchScene(GameScene.DAILY_REWARD); }
122.
123.     @FXML
124.     protected void onFastForwardClick() {
125.         // Fast-forward by clearing reward state and resetting money for testing
126.         var session = mainMenuModel.getSession();
127.         session.setLastRewardDate(LocalDate.now().minusDays(1));
128.         session.resetMoney();
129.         updateDailyRewardUI();
130.
131.         logger.logMessage("Fast forwarded to the next day, to next reward date and reset money.");
132.     }
133.
134. /**
135. * Updates the UI for the daily reward button and label.
136. * Hides the button if already claimed today.
137. */
138. private void updateDailyRewardUI()
139. {
140.     dailyRewardButton.setVisible(!mainMenuModel.getSession().hasClaimedRewardToday());
141. }
142.

```

ShopController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setup(Session, SceneManager) | | 0% | n/a | 1 | 1 | 13 | 13 | 1 | 1 | |
| syncReroll() | | 0% | | 0% | 2 | 2 | 4 | 4 | 1 | 1 |
| save() | | 0% | n/a | 1 | 1 | 2 | 2 | 1 | 1 | |
| ShopController() | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | |
| lambda\$setup\$1(ObservableValue, Number, Number) | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | |
| lambda\$setup\$0(ObservableValue, Number, Number) | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | |
| onSceneChangedToThis() | | 100% | n/a | 0 | 1 | 0 | 6 | 0 | 1 | |
| ShopController(ShopModel, UpgradeTileGroup, UpgradeTileGroup, Logger) | | 100% | n/a | 0 | 1 | 0 | 6 | 0 | 1 | |
| onUpgradeClicked(UpgradeTileController) | | 100% | n/a | 0 | 1 | 0 | 3 | 0 | 1 | |
| onNextLevelPressed() | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| onRerollPressed() | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| Total | 114 of 164 | 30% | 2 of 2 | 0% | 7 | 12 | 20 | 39 | 6 | 11 |

Created with JaCoCo 0.8.12.202403310830

ShopController.java

```

1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.models.tileGroups.UpgradeTileGroup;
4. import com.example.project.controllers.tiles.UpgradeTileController;
5. import com.example.project.models.gameScreens.ShopModel;
6. import com.example.project.services.*;
7. import javafx.fxml.FXML;
8. import javafx.scene.control.Button;
9. import javafx.scene.layout.HBox;
10. import javafx.scene.layout.Pane;
11. import javafx.scene.layout.StackPane;
12.
13. /**
14. * Shop view controller.
15. */
16. public class ShopController extends GameScreenController
17. {
18.     @FXML private Button rerollButton;
19.     @FXML private Pane root;
20.     @FXML private HBox shopItemsContainer;
21.     @FXML private HBox playersUpgradesContainer;
22.     @FXML private StackPane sidebar;
23.
24.     private ShopModel shopModel;
25.     private UpgradeTileGroup playersUpgrades;
26.     private UpgradeTileGroup shopItemsGroup;
27.
28.     /**
29.      * no arg constructor.
30.     */
31.     public ShopController() { }
32.
33.     /**
34.      * protected constructor for unit testing with mock model injection.
35.     */
36.     protected ShopController(ShopModel model, UpgradeTileGroup players, UpgradeTileGroup shopItems, Logger logger) {
37.         super();
38.         this.logger = logger;
39.         this.shopModel = model;
40.         this.playersUpgrades = players;
41.         this.shopItemsGroup = shopItems;
42.     }
43.
44.     @Override
45.     public void setup(Session session, SceneManager sceneManager)
46.     {
47.         this.shopModel = new ShopModel(session, sceneManager);
48.
49.         playersUpgrades = new UpgradeTileGroup(playersUpgradesContainer, shopModel.getSession().getPlayersUpgradesProperty());
50.         shopItemsGroup = new UpgradeTileGroup(shopItemsContainer, shopModel.getCurrentShopItemsProperty(),
51.             this::onUpgradeClicked);
52.
53.         var loadedSidebar = this.loadSidebar();
54.         var sidebarNode = ((StackPane) loadedSidebar.node());
55.         this.sidebar.getChildren().add(sidebarNode);
56.         SidebarController sidebarController = loadedSidebar.controller();
57.         sidebarController.bindPersistentInfo(session);
58.         sidebarController.hideLevelInfo();
59.
60.         this.shopModel.getRerollCostProperty().addListener((obs, oldStr, newStr) -> syncReroll());
61.         this.shopModel.getSession().getMoneyProperty().addListener((obs, oldStr, newStr) -> syncReroll());
62.         syncReroll();
63.     }
64.
65.     @Override
66.     public void onSceneChangedToThis()
67.     {
68.         this.logger.logMessage("Scene changed to shop");
69.         shopModel.regenerateShopItems();
70.         shopModel.getSession().resetPlaysRedraws();
71.         playersUpgrades.syncTiles();
72.         shopItemsGroup.syncTiles();
73.     }
74.
75.     private void syncReroll()
76.     {
77.         var rerollCost = this.shopModel.getRerollCostProperty().get();
78.         rerollButton.setDisable(rerollCost > this.shopModel.getSession().getMoneyProperty().get());
79.         rerollButton.setText(String.format("Reroll ($%d)", rerollCost));
80.     }
81.

```

```
82. /**
83. *
84. * @param controller upgrade tile ui element clicked on.
85. */
86. protected void onUpgradeClicked(UpgradeTileController controller)
87. {
88.     var model = controller.getModel();
89.     shopModel.tryPurchase(model);
90. }
91.
92. @FXML
93. protected void onNextLevelPressed()
94. {
95.     this.shopModel.onNextLevelPressed();
96. }
97.
98. @FXML
99. protected void onRerollPressed()
100. {
101.    this.shopModel.reroll();
102. }
103.
104. @FXML
105. protected void save()
106. {
107.    this.shopModel.getSession().Save();
108. }
109. }
```

Created with JaCoCo 0.8.12.202403310830

SidebarController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setupProperties(LevelModel) | | 0% | | n/a | 1 | 1 | 10 | 10 | 1 | 1 |
| bindPersistentInfo(Session) | | 0% | | n/a | 1 | 1 | 10 | 10 | 1 | 1 |
| syncScoreToBeat() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| syncMoney() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| hideLevelInfo() | | 0% | | n/a | 1 | 1 | 4 | 4 | 1 | 1 |
| syncwordMultiProperty(Number) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| syncwordPointsProperty(Number) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| syncTotalScoreProperty(Number) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| syncPlaysCount() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| syncRedrawsCount() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| lambda\$setupProperties\$6(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$setupProperties\$5(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$setupProperties\$4(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| SidebarController() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getComboLabel() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getRawPointsLabel() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getCurrentScoreLabel() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getmultiplierLabel() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$setupProperties\$7(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$bindPersistentInfo\$3(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$bindPersistentInfo\$2(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$bindPersistentInfo\$1(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$bindPersistentInfo\$0(ObservableValue, Number, Number) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 225 of 225 | 0% | 0 of 0 | n/a | 23 | 23 | 43 | 43 | 23 | 23 |

ConfettiAnimation.java

```
1. package com.example.project.controllers.gameScreens.animations;
2.
3. import javafx.scene.layout.Pane;
4. import javafx.scene.paint.Color;
5. import javafx.scene.shape.Rectangle;
6. import javafx.util.Duration;
7.
8. import java.util.Random;
9.
10. /**
11.  * class to play confetti animation.
12. */
13. public class ConfettiAnimation
14. {
15.     /**
16.      * play animation.
17.      * @param confettiLayer pane to play confetti on.
18.      */
19.     public static void playConfetti(Pane confettiLayer)
20.     {
21.         confettiLayer.getChildren().clear();
22.         Random rand = new Random();
23.         for (int i = 0; i < 75; i++) {
24.             Rectangle confetti = new Rectangle(6, 12);
25.             confetti.setFill(Color.hsb(rand.nextInt(360), 1.0, 1.0));
26.             confetti.setLayoutX(confettiLayer.getWidth() / 2);
27.             confetti.setLayoutY(confettiLayer.getHeight() / 2);

28.             var drop = new javafx.animation.TranslateTransition(Duration.seconds(1 + rand.nextDouble()), confetti);
29.             drop.setByX(rand.nextDouble() * 400 - 200);
30.             drop.setByY(rand.nextDouble() * 400 - 100);
31.             drop.setCycleCount(1);
32.             confettiLayer.getChildren().add(confetti);
33.             drop.play();
34.         }
35.     }
36. }
```

SidebarController.java

```
1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.models.gameScreens.LevelModel;
4. import com.example.project.services.Session;
5. import javafx.fxml.FXML;
6. import javafx.scene.control.Label;
7. import javafx.scene.layout.Pane;
8.
9. /**
10. * The sidebar on shop and level with the current round info.
11. */
12. public class SidebarController
13. {
14.     @FXML
15.     Pane root;
16.
17.     /**
18.      * This runs after the constructor and after all @FXML fields are initialized once each time application opened.
19.      */
20.     @FXML
21.     private Label moneyLabel;
22.
23.     @FXML private Label playsLeftLabel;
24.
25.     @FXML private Label redrawsLeftLabel;
26.
27.     @FXML Label scoreToBeatLabel;
28.     @FXML Label currentScoreLabel;
29.     @FXML Label rawPoints;
30.     @FXML Label multiplier;
31.     @FXML private Label ScoreAtLeastText;
32.     @FXML private Pane playersScoreContainer;
33.     @FXML private Pane wordScoreContainer;
34.
35.     private Session session;
36.
37.     /**
38.      * Binds things used in multiple screens, money, redraws plays.
39.      * @param session session.
40.      */
41.     public void bindPersistentInfo(Session session)
42.     {
43.         this.session = session;
44.
45.         this.session.getMoneyProperty().addListener((obs, oldVal, newVal) -> syncMoney());
46.         this.session.getCurrentPlays().addListener((obs, oldVal, newVal) -> syncPlaysCount());
47.         this.session.getCurrentRedraws().addListener((obs, oldVal, newVal) -> syncRedrawsCount());
48.         this.session.getLevelRequirement().addListener((obs, oldVal, newVal) -> syncScoreToBeat());
49.
50.         // TODO get tooltip setup to work with sidebar containers
51.
52.         syncPlaysCount();
53.         syncRedrawsCount();
54.         syncMoney();
55.         syncScoreToBeat();
56.     }
57.
58.     /**
59.      * Only show things relevant to upgrades or the shop. Money, redraws, plays.
60.      */
61.     public void hideLevelInfo()
62.     {
63.         playersScoreContainer.visibleProperty().set(false);
64.         wordScoreContainer.visibleProperty().set(false);
65.         ScoreAtLeastText.setText("Next level");
66.     }
67.
68.     /**
69.      * Returns combo label.
70.      * @return combo label.
71.      */
72.     public Label getComboLabel(){
73.         return this.multiplier;
74.     }
75.
76.     /**
77.      * Setup sync able properties.
78.      * @param levelModel level model.
```

```

79. */
80. public void setupProperties(LevelModel levelModel)
81. {
82.     // Binds the money display to Session money property for automatic updates
83.     bindPersistentInfo(levelModel.getSession());
84.     levelModel.wordPointsProperty().addListener((obs, oldVal, newVal) -> syncwordPointsProperty(newVal));
85.     levelModel.wordMultiProperty().addListener((obs, oldVal, newVal) -> syncwordMultiProperty(newVal));
86.     levelModel.getPlayersTotalPoints().addListener((obs, oldVal, newVal) -> syncTotalScoreProperty(newVal));
87.     levelModel.getSession().getLevelRequirement().addListener((obs, oldVal, newVal) -> syncScoreToBeat());
88.
89.     syncwordPointsProperty(levelModel.wordPointsProperty().get());
90.     syncwordMultiProperty(levelModel.wordMultiProperty().get());
91.     syncTotalScoreProperty(levelModel.getPlayersTotalPoints().get());
92.     syncScoreToBeat();
93. }
94.
95. private void syncScoreToBeat(){
96.     scoreToBeatLabel.setText(String.format("%s", this.session.getLevelRequirement().get()));
97. }
98.
99. private void syncMoney(){
100. {
101.     moneyLabel.setText(String.format("Funds: $%.2f", this.session.getMoneyProperty().get()));
102. }
103.
104. private void syncPlaysCount(){
105.     playsLeftLabel.setText(String.valueOf(this.session.getCurrentPlays().get()));
106. }
107.
108. private void syncRedrawsCount(){
109.     redrawsLeftLabel.setText(String.valueOf(this.session.getCurrentRedraws().get()));
110. }
111.
112. /**
113. * raw points label to be multiplied by multiplier.
114. * @return label.
115. */
116. public Label getRawPointsLabel(){
117.     return this.rawPoints;
118. }
119.
120. /**
121. * current score label.
122. * @return label.
123. */
124. public Label getCurrentScoreLabel(){
125.     return this.currentScoreLabel;
126. }
127.
128. /**
129. * word score multiplier label.
130. * @return label.
131. */
132. public Label getmultiplierLabel(){
133.     return this.multiplier;
134. }
135.
136. private void syncwordMultiProperty(Number newVal)
137. {
138.     this.multiplier.setText(String.format("%s", newVal));
139. }
140.
141. private void syncwordPointsProperty(Number newVal)
142. {
143.     this.rawPoints.setText(String.format("%s", newVal));
144. }
145.
146. private void syncTotalScoreProperty(Number newVal)
147. {
148.     this.currentScoreLabel.setText(String.format("%s", newVal));
149. }
150. }

```

TooltipSetup.Element

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| static {...} | 0% | | n/a | | 1 | 1 | 6 | 6 | 1 | 1 |
| Total | 33 of 33 | 0% | 0 of 0 | n/a | 1 | 1 | 6 | 6 | 1 | 1 |

Created with JaCoCo 0.8.12.202403310830

TooltipSetup

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| createTooltipContainer() | | 0% | | n/a | 1 | 1 | 9 | 9 | 1 | 1 |
| createUpgradeGraphic(VBox, Object) | | 0% | | 0% | 2 | 2 | 8 | 8 | 1 | 1 |
| createStyledLabel(String, Color, Color) | | 0% | | n/a | 1 | 1 | 6 | 6 | 1 | 1 |
| setTooltipContent(TooltipSetup.Element, Object) | | 0% | | 0% | 6 | 6 | 9 | 9 | 1 | 1 |
| setupTooltip(Node, TooltipSetup.Element, Object) | | 0% | | n/a | 1 | 1 | 8 | 8 | 1 | 1 |
| createTooltipGraphic(VBox, String) | | 0% | | n/a | 1 | 1 | 5 | 5 | 1 | 1 |
| TooltipSetup() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 259 of 259 | 0% | 8 of 8 | 0% | 13 | 13 | 46 | 46 | 7 | 7 |

Created with JaCoCo 0.8.12.202403310830

TooltipSetup.java

```

1. package com.example.project.controllers.gameScreens;
2.
3. import com.example.project.models.tiles.UpgradeTileModel;
4. import javafx.geometry.Insets;
5. import javafx.geometry.Pos;
6. import javafx.scene.Node;
7. import javafx.scene.control.Tooltip;
8. import javafx.scene.control.Label;
9. import javafx.scene.layout.*;
10. import javafx.scene.paint.Color;
11. import javafx.util.Duration;
12.
13. /**
14. * sets a tooltip for desired element
15. */
16. public class TooltipSetup {
17.
18.     /**
19.      * different types of elements that require a tooltip
20.      */
21.     public enum Element {
22.         /**
23.          * tooltip for upgrade tiles
24.          */
25.         UPGRADE,
26.         /**
27.          * tooltip for daily rewards
28.          */
29.         DAILY,
30.         /**
31.          * tooltip for target score label
32.          */
33.         SCORETOBEAT,
34.         /**
35.          * tooltip for round score label
36.          */
37.         ROUNDSCORE,
38.         /**
39.          * tooltip for game funds label
40.          */
41.         FUNDS,
42.
43.     }
44.
45.     /**
46.      * @param node Node to attach tooltip to
47.      * @param element Element type
48.      * @param model Model associated with element
49.     */
50.     public void setupTooltip(Node node, Element element, Object model) {
51.         Node content = setTooltipContent(element, model);
52.
53.         // default constructor with empty string
54.         Tooltip tooltip = new Tooltip("");
55.
56.         // Set the custom graphic content
57.         tooltip.setGraphic(content);
58.
59.         tooltip.setPrefWidth(200);
60.         tooltip.setStyle("-fx-padding: 0;");
61.         tooltip.setShowDelay(Duration.seconds(0));
62.
63.         Tooltip.install(node, tooltip);
64.
65.     }
66.
67.     private Node setTooltipContent(Element element, Object model) {
68.
69.         VBox container = createTooltipContainer();
70.
71.         ◆ return switch (element) {
72.             case UPGRADE -> createUpgradeGraphic(container, model);
73.             case DAILY -> createTooltipGraphic(container, "Spin the wheel for your daily reward!");
74.             case SCORETOBEAT -> createTooltipGraphic(container, "The score you must reach before the number of plays run out! Be sure not to spend all of your redraws if you want some $$!");
75.             case ROUNDSCORE -> createTooltipGraphic(container, "The score you get after you play a word.");
76.             case FUNDS -> createTooltipGraphic(container, "How much $$$ you have. You earn more based on how many redraws you have left after winning!");
77.             default -> {
78.                 String description = "no description";
79.                 yield createTooltipGraphic(container, description);
80.             }
81.         };
82.     }
83.
84.     private VBox createTooltipContainer() {
85.         VBox container = new VBox(3); // 5 is the spacing between labels
86.         container.setPadding(new Insets(5));
87.
88.         container.setAlignment(Pos.TOP_CENTER);
89.
90.         container.setMaxSize(VBox.USE_PREF_SIZE, VBox.USE_PREF_SIZE);
91.
92.         Color bgColor = Color.color(1.0, 1.0, 1.0, 0.1);
93.         Color borderColor = Color.WHITE;
94.
95.         container.setBackground(new Background(new BackgroundFill(bgColor, new CornerRadii(6), Insets.EMPTY)));
96.         container.setBorder(new Border(new BorderStroke(
97.             borderColor,
98.             BorderStrokeStyle.SOLID,
99.             new CornerRadii(6),
100.            new BorderWidths(2) // 2px white border
101.        )));
102.         return container;
103.     }
104.
105.     private Node createTooltipGraphic(VBox container, String description) {
106.
107.         Label descriptionLabel = createStyledLabel(description, Color.web("#d9d8c7"), Color.BLACK);
108.
109.         descriptionLabel.setWrapText(true);
110.
111.         descriptionLabel.setPrefWidth(180);
112.
113.         container.getChildren().addAll(descriptionLabel);
114.         return container;
115.     }
116.
117.     private Node createUpgradeGraphic(VBox container, Object model) {
118.         if (!(model instanceof UpgradeTileModel upgradeModel)) {
119.             return new Label("Error: Invalid model type or missing data.");
120.         }
121.
122.         Label nameLabel = createStyledLabel(upgradeModel.getName(), Color.web("#9f1124"), Color.WHITE);
123.     }

```

```
124.         container.getChildren().add(nameLabel);
125.
126.         createTooltipGraphic(container, upgradeModel.getDescription());
127.
128.         Label costLabel = createStyledLabel(String.format("Cost: %.2f", upgradeModel.getCost()), Color.web("#f98c3f"), Color.WHITE);
129.
130.         container.getChildren().add(costLabel);
131.
132.         return container;
133.     }
134.
135.     private Label createStyledLabel(String text, Color bgColor, Color txtColor) {
136.         Label label = new Label(text);
137.         label.setTextFill(txtColor);
138.
139.         label.setBackground(new Background(new BackgroundFill(bgColor, new CornerRadii(4), Insets.EMPTY)));
140.         label.setPadding(new Insets(2, 8, 2, 8));
141.
142.         label.setStyle("-fx-font-family: 'Inter'; -fx-font-weight: bold; -fx-font-size: 14px;");
143.
144.         return label;
145.     }
146. }
```

Created with JaCoCo 0.8.12.202403310830

com.example.project.controllers.gameScreens

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C LevelController | | 6% | | 3% | 53 | 57 | 165 | 177 | 26 | 30 | 0 | 1 |
| C TooltipSetup | | 0% | | 0% | 13 | 13 | 46 | 46 | 7 | 7 | 1 | 1 |
| C MainMenuController | | 0% | | 0% | 16 | 16 | 48 | 48 | 9 | 9 | 1 | 1 |
| C SidebarController | | 0% | | n/a | 23 | 23 | 43 | 43 | 23 | 23 | 1 | 1 |
| C LoginController | | 0% | | 0% | 9 | 9 | 29 | 29 | 7 | 7 | 1 | 1 |
| C GameScreenFactory | | 0% | | 0% | 6 | 6 | 31 | 31 | 4 | 4 | 1 | 1 |
| C DailyRewardController | | 0% | | 0% | 11 | 11 | 37 | 37 | 7 | 7 | 1 | 1 |
| C ShopController | | 30% | | 0% | 7 | 12 | 20 | 39 | 6 | 11 | 0 | 1 |
| C TooltipSetup.Element | | 0% | | n/a | 1 | 1 | 6 | 6 | 1 | 1 | 1 | 1 |
| C GameScreenController | | 20% | | n/a | 1 | 2 | 9 | 11 | 1 | 2 | 0 | 1 |
| C GameScreenController.SidebarLoaded | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 2,127 of 2,240 | 5% | 91 of 93 | 2% | 141 | 151 | 435 | 468 | 92 | 102 | 8 | 11 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.controllers.gameScreens

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| LevelController.java | | 6% | | 3% | 53 | 57 | 165 | 177 | 26 | 30 | 0 | 1 |
| TooltipSetup.java | | 0% | | 0% | 14 | 14 | 52 | 52 | 8 | 8 | 2 | 2 |
| MainMenuController.java | | 0% | | 0% | 16 | 16 | 48 | 48 | 9 | 9 | 1 | 1 |
| SidebarController.java | | 0% | | n/a | 23 | 23 | 43 | 43 | 23 | 23 | 1 | 1 |
| LoginController.java | | 0% | | 0% | 9 | 9 | 29 | 29 | 7 | 7 | 1 | 1 |
| GameScreenFactory.java | | 0% | | 0% | 6 | 6 | 31 | 31 | 4 | 4 | 1 | 1 |
| DailyRewardController.java | | 0% | | 0% | 11 | 11 | 37 | 37 | 7 | 7 | 1 | 1 |
| ShopController.java | | 30% | | 0% | 7 | 12 | 20 | 39 | 6 | 11 | 0 | 1 |
| GameScreenController.java | | 16% | | n/a | 2 | 3 | 10 | 12 | 2 | 3 | 1 | 2 |
| Total | 2,127 of 2,240 | 5% | 91 of 93 | 2% | 141 | 151 | 435 | 468 | 92 | 102 | 8 | 11 |

Created with JaCoCo 0.8.12.202403310830

DefinitionController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| syncwordProperty(String) | | 0% | | 0% | 3 | 3 | 14 | 14 | 1 | 1 |
| initialize(DefinitionPopup) | | 0% | | n/a | 1 | 1 | 5 | 5 | 1 | 1 |
| syncDefinitionProperty(String) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| DefinitionController() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| getModel() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$initialize\$1(ObservableValue, String, String) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| lambda\$initialize\$0(ObservableValue, String, String) | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getStack() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 139 of 139 | 0% | 4 of 4 | 0% | 10 | 10 | 25 | 25 | 8 | 8 |

Created with JaCoCo 0.8.12.202403310830

DefinitionController.java

```
1. package com.example.project.controllers.popupControllers;
2.
3. import com.example.project.controllers.gameScreens.animations.InfiniteFloatingAnimation;
4. import com.example.project.controllers.tiles.TileControllerFactory;
5. import com.example.project.models.tiles.LetterTileModel;
6. import javafx.fxml.FXML;
7. import javafx.scene.image.ImageView;
8. import javafx.scene.control.Label;
9. import javafx.scene.layout.Pane;
10. import javafx.scene.Node;
11. import javafx.scene.layout.StackPane;
12. import com.example.project.models.popups.DefinitionPopup;
13. import java.util.ArrayList;
14. import java.util.List;
15.
16.
17. /**
18. * Controller for definition popup
19. */
20. public class DefinitionController extends PopupController<DefinitionPopup>
21. {
22.     @FXML
23.     private StackPane defStack;
24.
25.     @FXML
26.     Pane wordRow;
27.
28.     @FXML
29.     Label definitionLabel;
30.
31.     @FXML
32.     ImageView definitionImage;
33.
34.     /**
35.      * @return returns Node
36.     */
37.     public Node getStack() { return defStack; }
38.
39.     public DefinitionPopup getModel(){
40.         return model;
41.     }
42.
43.     private final TileControllerFactory tileControllerFactory = new TileControllerFactory();
44.
45.     @Override
46.     public void initialize(DefinitionPopup model)
47.     {
48.         this.model = model;
49.         wordRow.setMouseTransparent(true);
50.         model.definitionProperty().addListener((obs, oldStr, newStr) -> syncDefinitionProperty(newStr));
51.         model.wordProperty().addListener((obs, oldStr, newStr) -> syncwordProperty(newStr));
52.     }
53.
54.     private void syncDefinitionProperty(String newStr){
55.         this.definitionLabel.setText(String.format("%s", newStr));
56.     }
57.
58.     private void syncwordProperty(String newStr){
59.         // TODO: turn this into a reusable external function (same used in LoginController)
60.         wordRow.getChildren().clear();
61.         List<LetterTileModel> lettersInWordPlayed = new ArrayList<>();
62.         char[] charArray = newStr.toCharArray();
63.
64.         for (char letter : charArray){
65.             lettersInWordPlayed.add(new LetterTileModel(letter));
66.         }
67.
68.         for (var t : lettersInWordPlayed){
69.             var ltController = tileControllerFactory.createLetterTileController(t);
70.             ltController.getRoot().setScaleX(-(wordRow.getMaxWidth()/1.5));
71.             ltController.getRoot().setScaleY(-(wordRow.getMaxWidth()/1.5));
72.             wordRow.getChildren().add(ltController.getRoot());
73.             InfiniteFloatingAnimation fa = new InfiniteFloatingAnimation();
74.             fa.apply(ltController.getRoot(), 4);
75.         }
76.     }
77.
78. }
```


PopupController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|-----------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| PopupController() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getModel() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 6 of 6 | 0% | 0 of 0 | n/a | 2 | 2 | 2 | 2 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

PopupController.java

```
1. package com.example.project.controllers.popupControllers;
2.
3. import com.example.project.models.popups.PopupModel;
4. import javafx.fxml.FXML;
5. import javafx.scene.Node;
6.
7. /**
8. * Parent class for PopupControllers
9. * @param <P> Popup model type
10.*/
11. public abstract class PopupController<P extends PopupModel>
12. {
13.     /**
14.      * root ui node.
15.      * @return root ui Node
16.      */
17.     public abstract Node getStack();
18.
19.     /**
20.      * sets elements of the ui
21.      * @param popup popup to initialize
22.      */
23.     @FXML
24.     public abstract void initialize(P popup);
25.
26.     protected P model;
27.
28.     /**
29.      * gets the popup model.
30.      * @return model.
31.      */
32.     public P getModel()
33.     {
34.         return this.model;
35.     }
36. }
```

InfiniteFloatingAnimation

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| moveToRandom(Node, double, double) | | 0% | n/a | 1 | 1 | 10 | 10 | 1 | 1 | |
| InfiniteFloatingAnimation() | | 0% | n/a | 1 | 1 | 2 | 2 | 1 | 1 | |
| apply(Node, double) | | 0% | n/a | 1 | 1 | 2 | 2 | 1 | 1 | |
| lambda\$moveToRandom\$0(Node, double, double, ActionEvent) | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | |
| Total | 68 of 68 | 0% | 0 of 0 | n/a | 4 | 4 | 14 | 14 | 4 | 4 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.controllers.popupControllers

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--------------------------------------|--|--|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| DefinitionController |  0% |  0% | 10 | 10 | 25 | 25 | 8 | 8 | 1 | 1 | 1 | 1 |
| PopupController |  0% | n/a | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| Total | 145 of 145 | 0% | 4 of 4 | 0% | 12 | 12 | 27 | 27 | 10 | 10 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.controllers.popupControllers

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---|--|--|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| DefinitionController.java |  0% |  0% | 10 | 10 | 25 | 25 | 8 | 8 | 1 | 1 | | |
| PopupController.java |  0% | n/a | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | | |
| Total | 145 of 145 | 0% | 4 of 4 | 0% | 12 | 12 | 27 | 27 | 10 | 10 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

EmptyTileSlotController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setLetter(LetterTileController) | 100% | 100% | 100% | 100% | 0 | 3 | 0 | 6 | 0 | 1 |
| updateView() | 100% | 100% | 100% | 100% | 0 | 2 | 0 | 4 | 0 | 1 |
| bind(EmptyTileSlotModel) | 100% | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| EmptyTileSlotController() | 100% | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getRoot() | 100% | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 50 | 100% | 0 of 6 | 100% | 0 | 8 | 0 | 14 | 0 | 5 |

Created with JaCoCo 0.8.12.202403310830

EmptyTileSlotController.java

```
1. package com.example.project.controllers.tiles;
2.
3. import com.example.project.models.tiles.EmptyTileSlotModel;
4. import javafx.fxml.FXML;
5. import javafx.scene.Node;
6. import javafx.scene.layout.StackPane;
7.
8. /**
9. * Empty Tile controller for EmptyTile view.
10.*/
11. public class EmptyTileSlotController extends TileController<EmptyTileSlotModel>
12. {
13.     @FXML
14.     protected StackPane root;
15.
16.     @FXML
17.     protected StackPane slotForLetterTile;
18.
19.     public void bind(EmptyTileSlotModel modelToUse)
20.     {
21.         this.model = modelToUse;
22.     }
23.
24.     private LetterTileController letterTileController;
25.
26.     public Node getRoot()
27.     {
28.         return root;
29.     }
30.
31.     /**
32.      * updates the tile slot with a new letter tile controller.
33.      * @param letterController the letter tiles controller.
34.      */
35.     public void setLetter(LetterTileController letterController)
36.     {
37.         if (this.model == null){
38.             throw new RuntimeException("model was null. call bind first.");
39.         }
40.
41.         this.model.setTile(letterController == null ? null : letterController.getModel());
42.         letterTileController = letterController;
43.         updateView();
44.     }
45.
46.     private void updateView()
47.     {
48.         slotForLetterTile.getChildren().clear();
49.
50.         if (this.letterTileController != null){
51.             slotForLetterTile.getChildren().add(this.letterTileController.getRoot());
52.         }
53.     }
54. }
```

LetterTileController

| Element | Missed Instructions Cov. | Missed Branches Cov. | Missed Cxts | Missed Lines | Missed Methods |
|--|-----------------------------|-------------------------|----------------|-----------------|-------------------|
| bind(LetterTileModel) | 0% | n/a | 1 1 | 10 10 | 1 1 |
| getModel() | 0% | n/a | 1 1 | 1 1 | 1 1 |
| LetterTileController() | 0% | n/a | 1 1 | 1 1 | 1 1 |
| getRoot() | 0% | n/a | 1 1 | 1 1 | 1 1 |
| Total | 54 of 54 | 0% 0 of 0 | n/a | 4 4 | 13 13 |

Created with JaCoCo 0.8.12.202403310830

LetterTileController.java

```
1. package com.example.project.controllers.tiles;
2.
3. import com.example.project.models.tiles.LetterTileModel;
4. import javafx.fxml.FXML;
5. import javafx.scene.CacheHint;
6. import javafx.scene.Node;
7. import javafx.scene.control.Label;
8. import javafx.scene.image.ImageView;
9. import javafx.scene.layout.StackPane;
10.
11. /**
12. * Controls the layout of the letterUIModel. Which is a letter tile.
13. */
14. public class LetterTileController extends TileController<LetterTileModel>
15. {
16.     @FXML
17.     private StackPane root;
18.
19.     @FXML
20.     ImageView tileImage;
21.
22.     @FXML
23.     Label letterLabel;
24.
25.     @FXML
26.     Label valueLabel;
27.
28.     public Node getRoot()
29.     {
30.         return root;
31.     }
32.
33.     public LetterTileModel getModel(){
34.         return model;
35.     }
36.
37.     @Override
38.     public void bind(LetterTileModel tile)
39.     {
40.         this.model = tile;
41.         letterLabel.setText(String.valueOf(model.getLetter()));
42.         valueLabel.setText(String.valueOf(model.getValue()));
43.
44.         letterLabel.setCache(true);
45.         letterLabel.setCacheHint(CacheHint.QUALITY);
46.
47.         valueLabel.setCache(true);
48.         valueLabel.setCacheHint(CacheHint.QUALITY);
49.
50.         tileImage.setCache(true);
51.         tileImage.setCacheHint(CacheHint.QUALITY);
52.     }
53. }
```

TileController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|-------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setModel(TileModel) | 100% | | n/a | | 0 | 1 | 0 | 2 | 0 | 1 |
| TileController() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| getModel() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 10 | 100% | 0 of 0 | n/a | 0 | 3 | 0 | 4 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

TileController.java

```
1. package com.example.project.controllers.tiles;
2.
3. import com.example.project.models.tiles.TileModel;
4. import javafx.scene.Node;
5.
6. /**
7.  * Parent class of TileControllers.
8.  * @param <T> tile model type.
9. */
10. public abstract class TileController<T extends TileModel>
11. {
12.     /**
13.      * root ui node.
14.      * @return root ui Node
15.      */
16.     public abstract Node getRoot();
17.
18.     /**
19.      * Binds tile to the controller as its model.
20.      * @param tile tile to bind.
21.      */
22.     public abstract void bind(T tile);
23.
24.     /**
25.      * Set the model this controller will observe.
26.      * @param tile tile model class.
27.      */
28.     public void setModel(T tile){
29.         model = tile;
30.     }
31.
32.     protected T model;
33.
34.     /**
35.      * gets the tile model.
36.      * @return model.
37.      */
38.     public T getModel()
39.     {
40.         return this.model;
41.     }
42. }
```

TileControllerFactory

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| lambda\$createLetterTileController\$3(LetterTileController) | | 0% | n/a | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| createTileController(TileModel, Class) | | 100% | | 100% | 0 | 4 | 0 | 6 | 0 | 1 |
| createGenericTileController(TileModel) | | 100% | n/a | 0 | 1 | 0 | 8 | 0 | 0 | 1 |
| createUpgradeTileController(UpgradeTileModel) | | 100% | n/a | 0 | 1 | 0 | 4 | 0 | 0 | 1 |
| createLetterTileController(LetterTileModel) | | 100% | n/a | 0 | 1 | 0 | 4 | 0 | 0 | 1 |
| TileControllerFactory(FXMLLoaderPageLoader) | | 100% | n/a | 0 | 1 | 0 | 4 | 0 | 0 | 1 |
| addHoverEffects(Node, Runnable) | | 100% | n/a | 0 | 1 | 0 | 3 | 0 | 0 | 1 |
| lambda\$addHoverEffects\$0(Node, Runnable, MouseEvent) | | 100% | n/a | 0 | 1 | 0 | 4 | 0 | 0 | 1 |
| TileControllerFactory() | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 0 | 1 |
| lambda\$addHoverEffects\$1(Node, MouseEvent) | | 100% | n/a | 0 | 1 | 0 | 3 | 0 | 0 | 1 |
| lambda\$createUpgradeTileController\$2(UpgradeTileController) | | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| createEmptyTileController(EmptyTileSlotModel) | | 100% | n/a | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| Total | 5 of 167 | 97% | 0 of 4 | 100% | 1 | 15 | 0 | 38 | 1 | 12 |

Created with JaCoCo 0.8.12.202403310830

TileControllerFactory.java

```
1. package com.example.project.controllers.tiles;
2.
3. import com.example.project.models.tiles.EmptyTileSlotModel;
4. import com.example.project.models.tiles.LetterTileModel;
5. import com.example.project.models.tiles.TileModel;
6. import com.example.project.models.tiles.UpgradeTileModel;
7. import com.example.project.services.FXMLPageLoader;
8. import javafx.scene.Node;
9.
10. import java.io.IOException;
11.
12. /**
13.  * class to load the tile model controller for that tile model type.
14. */
15. public class TileControllerFactory
16. {
17.     private FXMLPageLoader loader = new FXMLPageLoader();
18.
19.     /**
20.      * Creates instance of {@link TileControllerFactory}.
21.     */
22.     public TileControllerFactory() {}
23.
24.     protected TileControllerFactory(FXMLPageLoader loader)
25.     {
26.         this.loader = loader;
27.     }
28.
29.     /**
30.      * create tile controller.
31.      * @param tileObject tile object.
32.      * @param <C> class of tile controller.
33.      * @param <T> class of tile.
34.      * @return returns controller of the tile.
35.     */
36.     private <C extends TileController<T>, T extends TileModel> C createGenericTileController(T tileObject)
37.     {
38.         try
39.         {
40.             String fxmlPath = tileObject.getFXMLPath();
41.             loader.load(fxmlPath);
42.             C controller = loader.getController();
43.             controller.setModel(tileObject);
44.             controller.bind(tileObject);
45.             return controller;
46.         } catch (IOException e) {
47.             throw new RuntimeException("Failed to create tile controller: " + tileObject.getFXMLPath(), e);
48.         }
49.     }
50.
51.     private void addHoverEffects(Node pane, Runnable onEnter)
52.     {
53.         pane.setOnMouseEntered(e -> {
54.             pane.setScaleX(1.1);
55.             pane.setScaleY(1.1);
56.             onEnter.run();
57.         });
58.         pane.setOnMouseExited(e -> {
59.             pane.setScaleX(1.0);
60.             pane.setScaleY(1.0);
61.         });
62.     }
63.
64.     /**
65.      * @param upgradeTile upgrade tile model.
66.      * @return new upgrade tile controller.
67.     */
68.     private UpgradeTileController createUpgradeTileController(UpgradeTileModel upgradeTile)
69.     {
70.         UpgradeTileController upgradeTileController = createGenericTileController(upgradeTile);
71.         var pane = upgradeTileController.getRoot();
72.         addHoverEffects(pane, () -> upgradeTileController.getModel().getHoverSoundPlayer().replay());
73.         return upgradeTileController;
74.     }
75.
76.     /**
77.      * Returns controller of type.
78.      * @param tile tile model.
```

```
79.     * @param controllerType controller type.
80.     * @return return controller.
81.     * @param <C> controller type.
82.     * @param <T> tile type.
83.     */
84.     public <C extends TileController<T>, T extends TileModel> C createTileController(T tile, Class<C> controllerType)
85.     {
86.         var controller = switch (tile)
87.         {
88.             case UpgradeTileModel upgradeTileModel -> createUpgradeTileController(upgradeTileModel);
89.             case LetterTileModel letterTile -> createLetterTileController(letterTile);
90.             case EmptyTileSlotModel emptyTileSlotModel -> createEmptyTileController(emptyTileSlotModel);
91.             default -> throw new IllegalArgumentException("Unsupported tile type: " + tile.getClass());
92.         };
93.
94.         return controllerType.cast(controller);
95.     }
96.
97.     /**
98.      * @param lt letter tile model.
99.      * @return returns letter tile controller.
100.     */
101.    public LetterTileController createLetterTileController(LetterTileModel lt)
102.    {
103.        LetterTileController controller = createGenericTileController(lt);
104.        var pane = controller.getRoot();
105.        addHoverEffects(pane, () -> controller.getModel().getHoverSoundPlayer().replay());
106.        return controller;
107.    }
108.
109.    /**
110.     * @param emptyTile empty tile model.
111.     * @return returns empty tile controller.
112.     */
113.    private EmptyTileSlotController createEmptyTileController(EmptyTileSlotModel emptyTile){
114.        return createGenericTileController(emptyTile);
115.    }
116. }
```

InfiniteFloatingAnimation.java

```
1. package com.example.project.controllers.gameScreens.animations;
2.
3. import javafx.animation.TranslateTransition;
4. import javafx.scene.Node;
5. import javafx.util.Duration;
6.
7. import java.util.Random;
8.
9. /**
10. * floating tile animation.
11. */
12. public class InfiniteFloatingAnimation
13. {
14.     private final Random random = new Random();
15.
16.     /**
17.      * apply to node.
18.      * @param node node.
19.      * @param intervalSeconds time between changing directions.
20.      */
21.     public void apply(Node node, double intervalSeconds)
22.     {
23.         moveToRandom(node, intervalSeconds, intervalSeconds);
24.     }
25.
26.     private void moveToRandom(Node node, double minD, double maxD)
27.     {
28.         // pick random target position
29.         double x = random.nextDouble(0, 10);
30.         double y = random.nextDouble(0, 10);
31.
32.         double duration = minD + random.nextDouble() * (maxD - minD);
33.
34.         TranslateTransition tt = new TranslateTransition(Duration.seconds(duration), node);
35.         tt.setToX(x);
36.         tt.setToY(y);
37.         tt.setInterpolator(javafx.animation.Interpolator.LINEAR); // constant speed
38.
39.         // when done, pick a new target
40.         tt.setOnFinished(e -> moveToRandom(node, minD, maxD));
41.         tt.play();
42.     }
43. }
```

UpgradeTileController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| bind(UpgradeTileModel) | | 18% | | 50% | 1 | 2 | 14 | 16 | 0 | 1 |
| updateCount(int) | | 0% | | 0% | 2 | 2 | 6 | 6 | 1 | 1 |
| getRoot() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| UpgradeTileController() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 83 of 99 | 16% | 3 of 4 | 25% | 4 | 6 | 21 | 24 | 2 | 4 |

Created with JaCoCo 0.8.12.202403310830

UpgradeTileController.java

```
1. package com.example.project.controllers.tiles;
2.
3. import com.example.project.models.tiles.TileModel;
4. import com.example.project.models.tiles.UpgradeTileModel;
5. import com.example.project.controllers.gameScreens.TooltipSetup;
6. import javafx.fxml.FXML;
7. import javafx.scene.Node;
8. import javafx.scene.control.Label;
9. import javafx.scene.image.Image;
10. import javafx.scene.image.ImageView;
11. import javafx.scene.layout.StackPane;
12.
13. import javafx.scene.shape.Rectangle;
14.
15. /**
16. * Controls the upgrade tile view fxml file.
17. */
18. public class UpgradeTileController extends TileController<UpgradeTileModel>
19. {
20.     @FXML
21.     protected StackPane root;
22.
23.     @FXML
24.     protected ImageView imageView;
25.
26.     @FXML
27.     protected Label countText;
28.
29.     public Node getRoot()
30.     {
31.         return root;
32.     }
33.
34.     public void bind(UpgradeTileModel tileModel)
35.     {
36.         var path = getClass().getResource(tileModel.getAbilityImagePath());
37.         if (path == null) { throw new RuntimeException("path null"); }
38.         var url = path.toExternalForm();
39.         var image = new Image(url);
40.         this.imageView.setImage(image);
41.
42.         Rectangle clip = new Rectangle();
43.         // set corner radii
44.         clip.setArcWidth(TileModel.CORNER_RADIUS);
45.         clip.setArcHeight(TileModel.CORNER_RADIUS);
46.
47.         clip.widthProperty().bind(imageView.fitWidthProperty());
48.         clip.heightProperty().bind(imageView.fitHeightProperty());
49.
50.         imageView.setClip(clip);
51.
52.         TooltipSetup tooltip = new TooltipSetup();
53.         tooltip.setupTooltip(this.root, TooltipSetup.Element.UPGRADE, model);
54.
55.         this.countText.setText("");
56.         this.countText.setVisible(false);
57.     }
58.
59.     /**
60.      * updates upgrade # subscript
61.      * @param count the number of duplicate upgrades
62.      */
63.     public void updateCount(int count)
64.     {
65.         if (count > 1)
66.         {
67.             this.countText.setText(String.valueOf(count));
68.             this.countText.setVisible(true);
69.         }
70.         else
71.         {
72.             this.countText.setText("");
73.             this.countText.setVisible(false);
74.         }
75.     }
76. }
```

com.example.project.controllers.tiles

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C UpgradeTileController | | 16% | | 25% | 4 | 6 | 21 | 24 | 2 | 4 | 0 | 1 |
| C LetterTileController | | 0% | | n/a | 4 | 4 | 13 | 13 | 4 | 4 | 1 | 1 |
| C TileControllerFactory | | 97% | | 100% | 1 | 15 | 0 | 38 | 1 | 12 | 0 | 1 |
| C EmptyTileSlotController | | 100% | | 100% | 0 | 8 | 0 | 14 | 0 | 5 | 0 | 1 |
| C TileController | | 100% | | n/a | 0 | 3 | 0 | 4 | 0 | 3 | 0 | 1 |
| Total | 142 of 380 | 62% | 3 of 14 | 78% | 9 | 36 | 34 | 93 | 7 | 28 | 1 | 5 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.controllers.tiles

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxt | Missed | Lines | Missed | Methods | Missed | Classes | |
|--|---------------------|------|-----------------|------|--------|-----|--------|-------|--------|---------|--------|---------|---|
| UpgradeTileController.java | | 16% | | 25% | 4 | 6 | 21 | 24 | 2 | 4 | 0 | 1 | |
| LetterTileController.java | | 0% | | n/a | 4 | 4 | 13 | 13 | 4 | 4 | 1 | 1 | |
| TileControllerFactory.java | | 97% | | 100% | 1 | 15 | 0 | 38 | 1 | 12 | 0 | 1 | |
| EmptyTileSlotController.java | | 100% | | 100% | 0 | 8 | 0 | 14 | 0 | 5 | 0 | 1 | |
| TileController.java | | 100% | | n/a | 0 | 3 | 0 | 4 | 0 | 3 | 0 | 1 | |
| Total | 142 of 380 | 62% | 3 of 14 | | 78% | 9 | 36 | 34 | 93 | 7 | 28 | 1 | 5 |

Created with JaCoCo 0.8.12.202403310830

DevButton

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|-----------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| DevButton() | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| Total | 6 of 6 | 0% | 0 of 0 | n/a | 1 | 1 | 3 | 3 | 1 | 1 |

Created with JaCoCo 0.8.12.202403310830

DevButton.java

```
1. package com.example.project.controllers;
2.
3. import com.example.project.Application;
4. import javafx.fxml.FXML;
5. import javafx.scene.control.Button;
6.
7. /**
8. * wrapper class for dev buttons to toggle off or on when we want.
9. */
10. public class DevButton extends Button
11. {
12.     /**
13.      * Constructor.
14.     */
15.     public DevButton() {
16.         super();
17.         this.setVisible(Application.DEV_BUTTONS_ON);
18.         // this.setManaged(Application.DEV_BUTTONS_ON);
19.     }
20. }
```

RootLayoutController

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| toggleSound() | | 0% | | 0% | 3 | 3 | 5 | 5 | 1 | 1 |
| setContent(Parent) | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| initialize() | | 0% | | n/a | 1 | 1 | 4 | 4 | 1 | 1 |
| RootLayoutController() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| skipLogin() | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| setSceneManager(SceneManager) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| Total | 58 of 58 | 0% | 4 of 4 | 0% | 8 | 8 | 18 | 18 | 6 | 6 |

Created with JaCoCo 0.8.12.202403310830

RootLayoutController.java

```
1. package com.example.project.controllers;
2.
3. import com.example.project.services.GameScene;
4. import com.example.project.services.SceneManager;
5. import com.example.project.services.sound.GameMusicPlayer;
6. import com.example.project.services.SVGIIcons;
7. import javafx.fxml.FXML;
8. import javafx.scene.Parent;
9. import javafx.scene.control.Button;
10. import javafx.scene.layout.BorderPane;
11. import javafx.scene.layout.StackPane;
12.
13.
14. /**
15. * Root layout that remains the root scene of the application to switch scenes within.
16. */
17. public class RootLayoutController
18. {
19.     @FXML
20.     private BorderPane rootPane;
21.
22.     @FXML
23.     private StackPane contentPane;
24.
25.     @FXML
26.     private StackPane headerPane;
27.
28.     @FXML
29.     Button soundToggle;
30.
31.     private final GameMusicPlayer musicPlayer = new GameMusicPlayer();
32.
33.     private boolean isPlaying;
34.
35.     /**
36.      * @param page Set page content to a game scene.
37.      */
38.     public void setContent(Parent page) {
39.         contentPane.getChildren().clear();
40.         contentPane.getChildren().add(page);
41.     }
42.
43.     @FXML
44.     void initialize()
45.     {
46.         soundToggle.setGraphic(SVGIIcons.getCogIcon());
47.         musicPlayer.playGameMusicLoop();
48.         this.isPlaying = true;
49.     }
50.
51.     @FXML
52.     private void toggleSound()
53.     {
54.         if (isPlaying){
55.             musicPlayer.mute();
56.         }
57.         else{
58.             musicPlayer.unMute();
59.         }
60.
61.         isPlaying = !isPlaying;
62.     }
63.
64.     /**
65.      * Set scene manager to use.
66.      * @param sceneManager scene manager.
67.      */
68.     public void setSceneManager(SceneManager sceneManager)
69.     {
70.         this.sceneManager = sceneManager;
71.     }
72.
73.     private SceneManager sceneManager;
74.
75.     @FXML
76.     private void skipLogin()
77.     {
78.         sceneManager.switchScene(GameScene.LEVEL);
```

79. }

80. }

com.example.project.controllers

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--|--|--|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C RootLayoutController |  0% |  0% | 8 | 8 | 18 | 18 | 6 | 6 | 1 | 1 | 1 | 1 |
| C DevButton |  0% | n/a | 1 | 1 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 64 of 64 | 0% | 4 of 4 | 0% | 9 | 9 | 21 | 21 | 7 | 7 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.controllers

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| RootLayoutController.java | | 0% | | 0% | 8 | 8 | 18 | 18 | 6 | 6 | 1 | 1 |
| DevButton.java | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 | 1 | 1 |
| Total | 64 of 64 | 0% | 4 of 4 | 0% | 9 | 9 | 21 | 21 | 7 | 7 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

LevelScoreSequence

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| • LevelScoreSequence(List, LevelModel, Label, Label) | | 0% | | 0% | 2 | 2 | 16 | 16 | 1 | 1 |
| • lambda\$new\$0(LevelModel, LetterTileController, ActionEvent) | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| Total | 94 of 94 | 0% | 2 of 2 | 0% | 3 | 3 | 19 | 19 | 2 | 2 |

Created with JaCoCo 0.8.12.202403310830

DailyRewardModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| applyReward(DailyRewardType) | | 0% | | 0% | 3 | 3 | 5 | 5 | 1 | 1 |
| DailyRewardModel(Session, SceneManager) | | 0% | | n/a | 1 | 1 | 4 | 4 | 1 | 1 |
| rollReward() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 47 of 47 | 0% | 3 of 3 | 0% | 5 | 5 | 10 | 10 | 3 | 3 |

Created with JaCoCo 0.8.12.202403310830

DailyRewardModel.java

```
1. package com.example.project.models.gameScreens;
2.
3. import com.example.project.services.SceneManager;
4. import com.example.project.services.Session;
5.
6. import java.time.LocalDate;
7. import java.util.List;
8. import java.util.Random;
9.
10. /**
11.  * Model for handling daily reward logic and state.
12. */
13. public class DailyRewardModel extends GameScreenModel
14. {
15.     /**
16.      * Constructor.
17.      * @param session session.
18.      * @param sceneManager scene manager.
19.      */
20.     public DailyRewardModel(Session session, SceneManager sceneManager)
21.     {
22.         super(session, sceneManager);
23.     }
24.
25.     private final List<DailyRewardType> possibleRewards = List.of(
26.         DailyRewardType.Daily_Reward_Won_1Dollar,
27.         DailyRewardType.Daily_Reward_Won_5Dollars,
28.         DailyRewardType.Daily_Reward_Won_NOTHING
29.     );
30.
31.     private final Random random = new Random();
32.
33.     /**
34.      * Randomly selects a daily reward.
35.      * @return a randomly selected reward
36.      */
37.     public DailyRewardType rollReward() {
38.         return possibleRewards.get(random.nextInt(possibleRewards.size()));
39.     }
40.
41.     /**
42.      * Applies the effect of the given reward to the session and marks it as claimed.
43.      * @param reward the reward to apply
44.      */
45.     public void applyReward(DailyRewardType reward)
46.     {
47.         switch (reward) {
48.             case Daily_Reward_Won_1Dollar -> session.modifyMoney(1);
49.             case Daily_Reward_Won_5Dollars -> session.modifyMoney(5);
50.             case Daily_Reward_Won_NOTHING -> {
51.                 // no-op
52.             }
53.         }
54.
55.         session.setLastRewardDate(LocalDate.now());
56.     }
57. }
```

DailyRewardType

| Element | Missed Instructions Cov. | Missed Branches Cov. | Missed Cxty | Missed Lines | Missed Methods |
|--|--------------------------|----------------------|-------------|--------------|----------------|
|  static {...} | 0% | n/a | 1 1 | 4 4 | 1 1 |
| Total | 21 of 21 | 0% 0 of 0 | n/a | 1 1 | 4 4 |

Created with JaCoCo 0.8.12.202403310830

DailyRewardType.java

```
1. package com.example.project.models.gameScreens;
2.
3. /**
4.  * Enum representing the possible daily reward outcomes.
5. */
6. public enum DailyRewardType {
7.     /**
8.      * Player wins $1
9.     */
10.    Daily_Reward_Won_1Dollar,
11.
12.    /**
13.     * Player wins $5
14.     */
15.    Daily_Reward_Won_5Dollars,
16.
17.    /**
18.     * Player wins nothing
19.     */
20.    Daily_Reward_Won_NOTHING
21. }
```

GameScreenModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| • getSession() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| • getSceneManager() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| • GameScreenModel(Session, SceneManager) | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| Total | 6 of 20 | 70% | 0 of 0 | n/a | 2 | 3 | 2 | 7 | 2 | 3 |

Created with JaCoCo 0.8.12.202403310830

GameScreenModel.java

```
1. package com.example.project.models.gameScreens;
2.
3. import com.example.project.services.Logger;
4. import com.example.project.services.SceneManager;
5. import com.example.project.services.Session;
6.
7. /**
8. * Game Screen Model. extended by loginModel, shopModel, levelModel.
9. */
10. public abstract class GameScreenModel
11. {
12.     protected Logger logger = new Logger();
13.
14.     protected final Session session;
15.
16.     protected final SceneManager sceneManager;
17.
18.     /**
19.      * Get session.
20.      * @return session.
21.     */
22.     public Session getSession() { return session; }
23.
24.     /**
25.      * get scene manager
26.      * @return scene manager
27.     */
28.     public SceneManager getSceneManager() { return sceneManager; }
29.
30.     protected GameScreenModel (Session session, SceneManager sceneManager)
31.     {
32.         this.sceneManager = sceneManager;
33.         this.session = session;
34.     }
35. }
```

LevelModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| calcTotalWordScore() | | 0% | | 0% | 2 | 2 | 4 | 4 | 1 | 1 |
| addToCombo(LetterTileModel) | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| refillTileTack() | | 66% | | 50% | 1 | 2 | 1 | 5 | 0 | 1 |
| resetCombo() | | 0% | | n/a | 1 | 1 | 3 | 3 | 1 | 1 |
| setWordPoints(int) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| setWordMulti(int) | | 0% | | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| getPlayersTotalPoints() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| wordPointsProperty() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| wordMultiProperty() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getIsRedrawActive() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| getTileScoreSoundPlayer() | | 0% | | n/a | 1 | 1 | 1 | 1 | 1 | 1 |
| LevelModel(Session, Logger, SceneManager) | | 100% | | n/a | 0 | 1 | 0 | 13 | 0 | 1 |
| LevelModel(Session, SceneManager) | | 100% | | n/a | 0 | 1 | 0 | 13 | 0 | 1 |
| tryMoveTile(LetterTileModel) | | 100% | | 100% | 0 | 6 | 0 | 11 | 0 | 1 |
| onWonLevel() | | 100% | | 50% | 1 | 2 | 0 | 9 | 0 | 1 |
| tryMoveTileToRedrawArea(LetterTileModel) | | 100% | | 66% | 2 | 4 | 0 | 5 | 0 | 1 |
| tryMoveToTileRack(LetterTileModel) | | 100% | | 75% | 1 | 3 | 0 | 7 | 0 | 1 |
| tryMoveTileToWordArea(LetterTileModel) | | 100% | | 50% | 2 | 3 | 0 | 4 | 0 | 1 |
| getCurrentWord() | | 100% | | 100% | 0 | 2 | 0 | 5 | 0 | 1 |
| generateLetters() | | 100% | | 100% | 0 | 2 | 0 | 4 | 0 | 1 |
| returnRedrawTilesToTheRack() | | 100% | | 100% | 0 | 2 | 0 | 3 | 0 | 1 |
| redrawTiles() | | 100% | | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| playTiles() | | 100% | | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| setupNewLevel() | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| hasWon() | | 100% | | 100% | 0 | 2 | 0 | 1 | 0 | 1 |
| onLostLevel() | | 100% | | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| hasLost() | | 100% | | 100% | 0 | 2 | 0 | 1 | 0 | 1 |
| resetLevelVariables() | | 100% | | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| getWordWindowTilesProperty() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getTileRackTilesProperty() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getRedrawWindowTilesProperty() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| addTileToWordWindow(LetterTileModel) | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| addTileToRack(LetterTileModel) | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| addTileToRedrawWindow(LetterTileModel) | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| isCurrentWordValid() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| setIsRedrawActive(boolean) | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| setTotalScore(int) | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| getPlayersCurrentPoints() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 93 of 627 | 85% | 9 of 40 | 77% | 18 | 58 | 20 | 125 | 10 | 38 |

LevelModel.java

```
1. package com.example.project.models.gameScreens;
2.
3. import com.example.project.models.tiles.LetterTileModel;
4. import com.example.project.models.tiles.UpgradeTileModel;
5. import com.example.project.services.GameScene;
6. import com.example.project.services.Logger;
7. import com.example.project.services.SceneManager;
8. import com.example.project.services.Session;
9. import com.example.project.models.tiles.ScrabbleTileProvider;
10. import com.example.project.services.sqlite.DAOs.DictionaryDAO;
11. import javafx.beans.property.*;
12. import javafx.collections.FXCollections;
13. import javafx.collections.ObservableList;
14.
15.
16. /**
17. * Represents the level model.
18. */
19. public class LevelModel extends GameScreenModel
20. {
21.     private final ObservableList<LetterTileModel> wordWindowTiles = FXCollections.observableArrayList();
22.     private final ObservableList<LetterTileModel> tileRackTiles = FXCollections.observableArrayList();
23.     private final ObservableList<LetterTileModel> redrawWindowTiles = FXCollections.observableArrayList();
24.     private final ReadOnlyBooleanWrapper isRedrawActive = new ReadOnlyBooleanWrapper(false);
25.     private final ReadOnlyIntegerWrapper wordPoints = new ReadOnlyIntegerWrapper(0);
26.     private final ReadOnlyIntegerWrapper wordMulti = new ReadOnlyIntegerWrapper(0);
27.     private final ReadOnlyIntegerWrapper playersTotalPoints = new ReadOnlyIntegerWrapper(0);
28.     private final DictionaryDAO dictionary = new DictionaryDAO();
29.
30.     private final ScoreChimePlayer tileScoreSoundPlayer = new ScoreChimePlayer();
31.     private final ScrabbleTileProvider scrabbleLettersBalancer = new ScrabbleTileProvider();
32.
33.     protected LevelModel(Session session, Logger logger, SceneManager sceneManager)
34.     {
35.         super(session, sceneManager);
36.         this.logger = logger;
37.     }
38.
39.     /**
40.      * Default constructor.
41.      * @param session session
42.      * @param sceneManager scene manager.
43.      */
44.     public LevelModel(Session session, SceneManager sceneManager)
45.     {
46.         super(session, sceneManager);
47.         generateLetters();
48.     }
49.
50.     /**
51.      * Gets the tile score sound effect player.
52.      * @return LevelTileScoreSoundPlayer.
53.      */
54.     public ScoreChimePlayer getTileScoreSoundPlayer() { return this.tileScoreSoundPlayer; }
55.
56.     /**
57.      * @return Read-only list of tiles currently in the word area
58.      */
59.     public ReadOnlyListProperty<LetterTileModel> getWordWindowTilesProperty() {
60.         return new ReadOnlyListWrapper<>(wordWindowTiles).getReadOnlyProperty();
61.     }
62.
63.     /**
64.      * @return Read-only list of tiles currently in the rack
65.      */
66.     public ReadOnlyListProperty<LetterTileModel> getTileRackTilesProperty() {
67.         return new ReadOnlyListWrapper<>(tileRackTiles).getReadOnlyProperty();
68.     }
69.
70.     /**
71.      * @return Read-only list of tiles currently in the redraw window.
72.      */
73.     public ReadOnlyListProperty<LetterTileModel> getRedrawWindowTilesProperty() {
74.         return new ReadOnlyListWrapper<>(redrawWindowTiles).getReadOnlyProperty();
75.     }
76.     /**
77.      * @return the total points property to observe.
78.      */
```

```

79.     public ReadOnlyIntegerProperty getPlayersTotalPoints() {
80.         return playersTotalPoints.getReadOnlyProperty();
81.     }
82.
83.     /**
84.      * @return the sum combo points property to observe.
85.     */
86.     public ReadOnlyIntegerProperty wordPointsProperty() {
87.         return wordPoints.getReadOnlyProperty();
88.     }
89.
90.     /**
91.      * @return the players current level points property to observe.
92.     */
93.     public ReadOnlyIntegerProperty getPlayersCurrentPoints() { return playersTotalPoints.getReadOnlyProperty(); }
94.
95.     /**
96.      * word multiplier.
97.      * @return multiplier.
98.     */
99.     public ReadOnlyIntegerProperty wordMultiProperty() {
100.        return wordMulti.getReadOnlyProperty();
101.    }
102.
103.    /**
104.     * gets the redraw property.
105.     * @return returns indication if redraw active.
106.    */
107.    public ReadOnlyBooleanProperty getIsRedrawActive()
108.    {
109.        return isRedrawActive.getReadOnlyProperty();
110.    }
111.
112.    /**
113.     * set redraw active.
114.     * @param newValue get if redraw window is on screen.
115.    */
116.    public void setIsRedrawActive(boolean newValue)
117.    {
118.        this.isRedrawActive.set(newValue);
119.    }
120.
121.    protected void addTileToWordWindow(LetterTileModel tile)
122.    {
123.        this.wordWindowTiles.add(tile);
124.    }
125.
126.    protected void addTileToRack(LetterTileModel tile)
127.    {
128.        this.tileRackTiles.add(tile);
129.    }
130.
131.    protected void addTileToRedrawWindow(LetterTileModel tile)
132.    {
133.        this.redrawWindowTiles.add(tile);
134.    }
135.
136.    /**
137.     * Called when the level has been lost.
138.     * resets the players session info and logs back out to the login screen.
139.    */
140.    public void onLostLevel()
141.    {
142.        this.resetLevelVariables();
143.        this.session.resetGame();
144.        this.sceneManager.switchScene(GameScene.MAINMENU);
145.    }
146.
147.    /**
148.     * Called when level has been won.
149.     * reset the per level info: redraws plays. Goes to shop window.
150.    */
151.    public void onWonLevel()
152.    {
153.        //award money equal to remaining plays
154.        int remainingPlays = this.session.getCurrentPlays().get();
155.        if (remainingPlays > 0)
156.        {
157.            session.getMoneyProperty().set(session.getMoneyProperty().get() + remainingPlays);
158.            this.logger.logMessage(String.format("You Won! Awarded $%d for %d remaining plays",
159.                remainingPlays, remainingPlays));
160.        }
161.
```

```

162.     this.resetLevelVariables();
163.     this.session.updateLevelInfo();
164.     this.sceneManager.switchScene(GameScene.SHOP);
165. }
166.
167. /**
168.  * returns true if player has won.
169.  * @return value indicating if player has won.
170. */
171. public boolean hasWon()
172. {
173.     return (this.session.getLevelRequirement().get() <= this.playersTotalPoints.get());
174. }
175.
176. /**
177.  * true if player has lost
178.  * @return value indicating if player has lost.
179. */
180. public boolean hasLost()
181. {
182.     return this.session.getCurrentPlays().get() == 0;
183. }
184.
185. /**
186.  * Attempts to move a tile from rack to word area
187.  * @param tile The tile to move
188.  * @return true if move was successful, false otherwise
189. */
190. private boolean tryMoveTileToWordArea(LetterTileModel tile) {
191.     if (tileRackTiles.contains(tile) && wordWindowTiles.size() < session.getWordWindowSize()) {
192.         tileRackTiles.remove(tile);
193.         wordWindowTiles.add(tile);
194.     }
195.     return wordWindowTiles.contains(tile);
196. }
197.
198. /**
199.  * Attempts to move a tile from rack to redraw area
200.  * @param tile The tile to move
201.  * @return true if move was successful, false otherwise
202. */
203. private boolean tryMoveTileToRedrawArea(LetterTileModel tile)
204. {
205.     if (redrawWindowTiles.size() < session.getRedrawWindowSize() && !redrawWindowTiles.contains(tile))
206.     {
207.         var rowItsIn = tileRackTiles.contains(tile) ? tileRackTiles : wordWindowTiles;
208.         rowItsIn.remove(tile);
209.         redrawWindowTiles.add(tile);
210.     }
211.
212.     return redrawWindowTiles.contains(tile);
213. }
214.
215. /**
216.  * Attempts to move a tile from word area or redraw area to rack
217.  * @param tile The tile to move
218.  * @return true if move was successful, false otherwise
219. */
220. private boolean tryMoveToTileRack(LetterTileModel tile) {
221.     if (wordWindowTiles.contains(tile)) {
222.         wordWindowTiles.remove(tile);
223.         tileRackTiles.add(tile);
224.     } else if (redrawWindowTiles.contains(tile)) {
225.         redrawWindowTiles.remove(tile);
226.         tileRackTiles.add(tile);
227.     }
228.
229.     return tileRackTiles.contains(tile);
230. }
231.
232. /**
233.  * determines where tile should go and moves it
234.  * @param tile The tile to move
235.  * @return true if move was successful, false otherwise
236. */
237. public boolean tryMoveTile(LetterTileModel tile) {
238.     boolean moved = false;
239.
240.     // When no redraw window is open.
241.     if (!isRedrawActive.get()) {
242.         if (tileRackTiles.contains((tile))) {
243.             moved = tryMoveTileToWordArea(tile);
244.         }

```

```

245.         else if (wordWindowTiles.contains(tile)) {
246.             moved = tryMoveToTileRack(tile);
247.         }
248.     }
249.     // when redraw window is open.
250.     else
251.     {
252.         if (!redrawWindowTiles.contains(tile)) {
253.             moved = tryMoveTileToRedrawArea(tile);
254.         }
255.         else {
256.             moved = tryMoveToTileRack(tile);
257.         }
258.     }
259.
260.     if (moved) { tile.getClackSoundPlayer().replay(); }
261.     return moved;
262. }
263.
264. /**
265. * Gets the current word formed by tiles in the word area
266. * @return returns current word string.
267. */
268. public String getCurrentWord() {
269.     StringBuilder word = new StringBuilder();
270.     for (LetterTileModel tile : wordWindowTiles) {
271.         word.append(tile.getLetter());
272.     }
273.     return word.toString();
274. }
275.
276. /**
277. * returns true if word is in dictionary.
278. * @return value indicating if word is valid.
279. */
280. public boolean isCurrentWordValid(){
281.     return dictionary.isWordInDictionary(this.getCurrentWord());
282. }
283.
284. /**
285. * redraws tiles into the tile rack and removes from redraw window.
286. */
287. public void redrawTiles()
288. {
289.     this.session.getCurrentRedraws().set(this.session.getCurrentRedraws().get() - 1);
290.     this.redrawWindowTiles.clear();
291.     refillTileTack();
292. }
293.
294. /**
295. * add combo sum and multiCombo
296. * @param tile tile.
297. */
298. public void addToCombo(LetterTileModel tile) {
299.     this.wordPoints.set(this.wordPoints.get() + tile.getValue());
300.     this.wordMulti.set(this.wordMulti.get() + 1);
301. }
302.
303. /**
304. * @return total score int
305. */
306. public int calcTotalWordScore() {
307.     for (UpgradeTileModel upgrade : session.getPlayersUpgradesProperty())
308.     {
309.         upgrade.runUpgradeEffect(this);
310.     }
311.
312.     return this.wordPoints.get() * this.wordMulti.get();
313. }
314.
315. /**
316. * sets the current word points before multipliers
317. * @param newWordPoints the new word points value
318. */
319. public void setWordPoints(int newWordPoints) {
320.     this.wordPoints.set(newWordPoints);
321. }
322.
323. /**
324. * sets the current word multiplier
325. * @param newMulti the new multiplier value
326. */
327. public void setWordMulti(int newMulti) {

```

```

328.     this.wordMulti.set(newMulti);
329. }
330.
331. /**
332. * @param totalScore from calcTotalScore
333. * sets Total Score
334. */
335. public void setTotalScore(int totalScore){
336.     this.playersTotalPoints.set(totalScore);
337. }
338.
339. /**
340. * clears the word row tiles. and refills the tile rack. and decreases the plays left.
341. */
342. public void playTiles()
343. {
344.     this.wordWindowTiles.clear();
345.     this.refillTileTack();
346.     this.session.getCurrentPlays().set(this.session.getCurrentPlays().get() - 1);
347. }
348.
349. /**
350. * sends the selected redraw tiles back to the rack
351. */
352. public void returnRedrawTilesToTheRack() {
353.     for (int i = 0; i < redrawWindowTiles.size(); i++) {
354.         tryMoveToTileRack(redrawWindowTiles.get(i));
355.     }
356. }
357.
358. /**
359. * Initialise new level. Clears word row, redraw rack. draws new tiles for the player's tile rack.
360. */
361. public void setupNewLevel()
362. {
363.     this.wordWindowTiles.clear();
364.     this.returnRedrawTilesToTheRack();
365.     isRedrawActive.set(false);
366.     this.session.resetPlaysRedraws();
367. }
368.
369. /**
370. * resets counts for sum and multi in combo
371. */
372. public void resetCombo()
373. {
374.     this.wordPoints.set(0);
375.     this.wordMulti.set(0);
376. }
377.
378. private void resetLevelVariables()
379. {
380.     this.playersTotalPoints.set(0);
381.     this.session.resetPlaysRedraws();
382. }
383.
384. private void generateLetters() {
385.     for (int i = 0; i < session.getHandSize(); i++) {
386.         var newLetter = new LetterTileModel(this.scrabbleLettersBalancer.drawRandomTile());
387.         this.tileRackTiles.add(newLetter); // Start all tiles in rack
388.     }
389. }
390.
391. /**
392. * Refills tile rack. Only up to the hand size allowed this level.
393. */
394. private void refillTileTack()
395. {
396.     var tilesPlayerHas = tileRackTiles.size() + wordWindowTiles.size() + redrawWindowTiles.size();
397.     var tilesToReplace = (this.session.getHandSize() - tilesPlayerHas);
398.     for (int i = 0; i < tilesToReplace; i++){
399.         tileRackTiles.add(new LetterTileModel(this.scrabbleLettersBalancer.drawRandomTile()));
400.     }
401. }
402. }

```

LoginModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| ● isInputLegal(String, String) | 100% | 100% | 91% | 91% | 1 | 7 | 0 | 13 | 0 | 1 |
| ● onLoginClicked(String, String) | 100% | 100% | 100% | 100% | 0 | 3 | 0 | 10 | 0 | 1 |
| ● onSignUpClicked(String, String) | 100% | 100% | 100% | 100% | 0 | 3 | 0 | 8 | 0 | 1 |
| ● LoginModel(Session, SceneManager, UsersDAO, PasswordHasher) | 100% | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| ● doesPasswordMatch(String, String) | 100% | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| ● getWelcomeTextProperty() | 100% | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 144 | 100% | 1 of 20 | 95% | 1 | 16 | 0 | 39 | 0 | 6 |

Created with JaCoCo 0.8.12.202403310830

LoginModel.java

```
1. package com.example.project.models.gameScreens;
2.
3. import com.example.project.models.User;
4. import com.example.project.services.PasswordHasher;
5. import com.example.project.services.Session;
6. import com.example.project.services.sqlite.dAOs.UsersDAO;
7. import com.example.project.services.GameScene;
8. import com.example.project.services.SceneManager;
9. import javafx.beans.property.ReadOnlyStringProperty;
10. import javafx.beans.property.ReadOnlyStringWrapper;
11.
12. /**
13.  * Login model class.
14. */
15. public class LoginModel extends GameScreenModel
16. {
17.     private final UsersDAO usersDAO;
18.     private final PasswordHasher passwordHasher;
19.
20.     private final ReadOnlyStringWrapper infoText = new ReadOnlyStringWrapper("");
21.
22.     /**
23.      * Gets the info text property.
24.      * @return ReadonlyStringProperty.
25.     */
26.     public ReadOnlyStringProperty getWelcomeTextProperty(){
27.         return this.infoText;
28.     }
29.
30.     /**
31.      * constructor.
32.      * @param session session to use for the game.
33.      * @param sceneManager scenes.
34.      * @param dao users database implementation.
35.      * @param passwordHasher password hasher for this login
36.     */
37.     public LoginModel(Session session, SceneManager sceneManager, UsersDAO dao, PasswordHasher passwordHasher)
38.     {
39.         super(session, sceneManager);
40.         this.usersDAO = dao;
41.         this.passwordHasher = passwordHasher;
42.     }
43.
44.     /**
45.      * @param username username
46.      * @param password password
47.      * @return returns value indicating if login is valid.
48.     */
49.     private boolean doesPasswordMatch(String username, String password)
50.     {
51.         var user = this.usersDAO.getUser(username);
52.         return this.passwordHasher.doesPasswordMatch(password, user.getPassword());
53.     }
54.
55.     /**
56.      * @param username username.
57.      * @param password password.
58.     */
59.     public void onLoginClicked(String username, String password)
60.     {
61.         if (!this.usersDAO.doesUserExist(username)) {
62.             this.infoText.set("No account found. Sign up first.");
63.             return;
64.         }
65.
66.         if (!this.doesPasswordMatch(username, password)) {
67.             this.infoText.set("Incorrect password.");
68.             return;
69.         }
70.
71.         var user = this.usersDAO.getUser(username);
72.         this.session.setUser(user);
73.         this.sceneManager.switchScene(GameScene.MAINMENU);
74.     }
75.
76.     /**
77.      * returns value indicating if the input is legal. Illegal if blank.
78.      * @param username username.
```

```

79.     * @param password password.
80.     * @return value indicating if the input is legal.
81.     */
82.     private boolean isInputLegal(String username, String password)
83.     {
84.         if (username.isBlank() || password.isBlank()) {
85.             this.infoText.set("Fields cannot be empty.");
86.             return false;
87.         }
88.
89.         if (username.length() < 3 || username.length() > 30) {
90.             this.infoText.set("Username must be between 3-30 characters long.");
91.             return false;
92.         }
93.
94.         if (password.length() < 8) {
95.             this.infoText.set("Password must be at least 8 characters.");
96.             return false;
97.         }
98.
99.         if (password.equals(username)) {
100.             this.infoText.set("Password must not match username.");
101.             return false;
102.         }
103.
104.         return true;
105.     }
106.
107. /**
108. * Adds user to database.
109. * @param username username
110. * @param password password
111. */
112. public void onSignUpClicked(String username, String password)
113. {
114.     if (!isInputLegal(username, password)){
115.         return;
116.     }
117.
118.     if (this.usersDAO.doesUserExist(username))
119.     {
120.         this.infoText.set("Already signed up. Click login.");
121.         return;
122.     }
123.
124.     this.usersDAO.addUser(new User(username, password, 0));
125.     this.infoText.set("User added successfully.");
126. }
127. }

```

LevelScoreSequence.java

```
1. package com.example.project.controllers.gameScreens.animations;
2.
3. import com.example.project.controllers.tiles.LetterTileController;
4. import com.example.project.models.gameScreens.LevelModel;
5. import javafx.animation.*;
6. import javafx.scene.control.Label;
7. import javafx.scene.paint.Color;
8. import javafx.util.Duration;
9.
10. import java.util.List;
11.
12. /**
13. * Animates each level tile popping up and the score text increasing for every tile in the word row.
14. */
15. public class LevelScoreSequence extends AnimationWrapper
16. {
17.     /**
18.      * Constructor.
19.      * @param wordTileControllers tiles to animate.
20.      * @param levelModel level model.
21.      * @param comboCountLabel the text label to add the sum to.
22.      * @param multiplierLabel the text label to add the multi to.
23.      */
24.     public LevelScoreSequence(List<LetterTileController> wordTileControllers, LevelModel levelModel, Label comboCountLabel, Label multiplierLabel)
25.     {
26.         super();
27.
28.         for (LetterTileController control : wordTileControllers)
29.         {
30.             var translateUp = new TranslateTransition(Duration.seconds(0.1), control.getRoot());
31.             translateUp.setByY(-10);
32.             translateUp.setOnFinished(e ->
33.             {
34.                 levelModel.addToCombo(control.getModel());
35.                 levelModel.getTileScoreSoundPlayer().playNextNote();
36.             });
37.
38.             this.sequentialAnimation.getChildren().add(translateUp);
39.             var pointsTextColour = multiplierLabel.getTextFill();
40.             var multiplierTextColour = comboCountLabel.getTextFill();
41.             TextEmphasisAnimation sumComboSequence = new TextEmphasisAnimation(comboCountLabel, Color.GREEN, pointsTextColour, Duration.seconds(0));
42.             TextEmphasisAnimation multiComboSequence = new TextEmphasisAnimation(multiplierLabel, Color.GREEN, multiplierTextColour, Duration.seconds(0));
43.             this.sequentialAnimation.getChildren().addAll(sumComboSequence.getChildren());
44.             this.sequentialAnimation.getChildren().addAll(multiComboSequence.getChildren());
45.         }
46.
47.         // After all tiles
48.         var timeDelay = new PauseTransition(Duration.seconds(1));
49.         this.sequentialAnimation.getChildren().add(timeDelay);
50.     }
51. }
```

MainMenuModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| MainMenuModel(Session, SceneManager) | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| onStartClicked() | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| onLogoutClicked() | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 | |
| Total | 0 of 15 | 100% | 0 of 0 | n/a | 0 | 3 | 0 | 6 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

MainMenuModel.java

```
1. package com.example.project.models.gameScreens;
2.
3. import com.example.project.services.GameScene;
4. import com.example.project.services.SceneManager;
5. import com.example.project.services.Session;
6.
7. /**
8. * main menu model class
9. */
10. public class MainMenuModel extends GameScreenModel
11. {
12.     /**
13.      * constructor.
14.      * @param session session to use for the game.
15.      * @param sceneManager scenes.
16.      */
17.     public MainMenuModel(Session session, SceneManager sceneManager)
18.     {
19.         super(session, sceneManager);
20.     }
21.
22.     /**
23.      * switches to the level screen
24.      */
25.     public void onStartClicked()
26.     {
27.         this.sceneManager.switchScene(GameScene.LEVEL);
28.     }
29.
30.     /**
31.      * switches to the login screen
32.      */
33.     public void onLogoutClicked()
34.     {
35.         this.sceneManager.switchScene(GameScene.LOGIN);
36.     }
37. }
38. }
```

ScoreChimePlayer

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| ● playNextNote() | | 0% | | 0% | 2 | 2 | 5 | 5 | 1 | 1 |
| ● reset() | | 0% | n/a | n/a | 1 | 1 | 2 | 2 | 1 | 1 |
| ● ScoreChimePlayer() | | 100% | n/a | n/a | 0 | 1 | 0 | 12 | 0 | 1 |
| Total | 26 of 107 | 75% | 2 of 2 | 0% | 3 | 4 | 7 | 19 | 2 | 3 |

Created with JaCoCo 0.8.12.202403310830

ScoreChimePlayer.java

```
1. package com.example.project.models.gameScreens;
2.
3.
4. import com.example.project.services.sound.GameSoundPlayer;
5.
6. import java.util.List;
7.
8. /**
9. * plays a note from the F major scale ascending in pitch each time for a nice score combo sound.
10.*/
11. public class ScoreChimePlayer
12. {
13.     private final GameSoundPlayer F_4 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/F_4.wav");
14.     private final GameSoundPlayer G_4 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/G_4.wav");
15.     private final GameSoundPlayer A_4 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/A_4.wav");
16.     private final GameSoundPlayer Bb_4 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/Bb_4.wav");
17.     private final GameSoundPlayer C_4 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/C_4.wav");
18.     private final GameSoundPlayer D_4 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/D_5.wav");
19.     private final GameSoundPlayer E_4 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/E_5.wav");
20.     private final GameSoundPlayer F_5 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/F_5.wav");
21.     private final GameSoundPlayer G_5 = new GameSoundPlayer("/com/example/project/Sounds/scoreSounds/G_5.wav");
22.
23.     private final List<GameSoundPlayer> theCMajorScale = List.of(F_4, G_4, A_4, Bb_4, C_4, D_4, E_4, F_5, G_5);
24.
25.     private int currentNote = 0;
26.
27.     /**
28.      * resets back to 0.
29.     */
30.     public void reset(){
31.         currentNote = 0;
32.     }
33.
34.     /**
35.      * plays the note then increments the int index.
36.     */
37.     public void playNextNote()
38.     {
39.         if (theCMajorScale.size() == currentNote){
40.             reset();
41.         }
42.
43.         theCMajorScale.get(currentNote).replay();
44.         currentNote++;
45.     }
46. }
```

ShopModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| tryPurchase(UpgradeTileModel) | | 100% | | 100% | 0 | 3 | 0 | 14 | 0 | 1 |
| ShopModel(Session, SceneManager) | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| reroll() | | 100% | | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| regenerateShopItems() | | 100% | | 100% | 0 | 2 | 0 | 4 | 0 | 1 |
| ShopModel(Session, SceneManager, Logger) | | 100% | | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| onNextLevelPressed() | | 100% | | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| getCurrentShopItemsProperty() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getRerollCostProperty() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 150 | 100% | 0 of 6 | 100% | 0 | 11 | 0 | 34 | 0 | 8 |

Created with JaCoCo 0.8.12.202403310830

ShopModel.java

```
1. package com.example.project.models.gameScreens;
2.
3. import com.example.project.models.tiles.UpgradeTileModel;
4. import com.example.project.services.GameScene;
5. import com.example.project.services.Logger;
6. import com.example.project.services.SceneManager;
7. import com.example.project.services.Session;
8. import com.example.project.services.shopItems.UpgradeTiles;
9. import javafx.beans.property.IntegerProperty;
10. import javafx.beans.property.ListProperty;
11. import javafx.beans.property.ReadOnlyIntegerWrapper;
12. import javafx.beans.property.SimpleListProperty;
13. import javafx.collections.FXCollections;
14.
15. /**
16. * Shop Model.
17. */
18. public class ShopModel extends GameScreenModel
19. {
20.     protected final int numberOfShopItems = 3;
21.
22.     private final ListProperty<UpgradeTileModel> currentInShop = new SimpleListProperty<>(FXCollections.observableArrayList());
23.
24.     /**
25.      * Current items in the shop row that you can buy.
26.      * @return shop items.
27.     */
28.     public ListProperty<UpgradeTileModel> getCurrentShopItemsProperty(){
29.         return currentInShop;
30.     }
31.
32.     /**
33.      * Constructor for tests.
34.      * @param session session.
35.      * @param logger logger to use.
36.     */
37.     protected ShopModel(Session session, SceneManager sceneManager, Logger logger)
38.     {
39.         this(session, sceneManager);
40.         this.logger = logger;
41.     }
42.
43.     /**
44.      * Constructor
45.      * @param session game session.
46.      * @param sceneManager scenes.
47.     */
48.     public ShopModel(Session session, SceneManager sceneManager)
49.     {
50.         super(session, sceneManager);
51.     }
52.
53.     /**
54.      * create new shop items.
55.     */
56.     public void regenerateShopItems()
57.     {
58.         currentInShop.clear();
59.         ◆◆◆ for (int i = 0; i < numberOfShopItems; i++)
60.         {
61.             currentInShop.add(UpgradeTiles.getRandomUpgradeTile());
62.         }
63.     }
64.
65.     /**
66.      * reroll.
67.     */
68.     public void reroll()
69.     {
70.         this.regenerateShopItems();
71.         this.session.modifyMoney(-double) this.reRollCost.get());
72.         this.reRollCost.set(reRollCost.get() * 2);
73.     }
74.
75.     private final IntegerProperty reRollCost = new ReadOnlyIntegerWrapper(3);
76.
77.     /**
78.      * Get reroll cost.
79.      * @return integer property.
80.     */
81.     public IntegerProperty getRerollCostProperty(){
82.         return reRollCost;
```

```

83. }
84.
85. /**
86. * This should attempt to purchase an upgrade tile from the shop.
87. * Verifies the player has sufficient funds, deducts the cost, removes the item
88. * from the shop, and logs the transaction.
89. * @param tileClickedOn the upgrade tile the player is trying to purchase
90. * @throws IllegalArgumentException if tileClickedOn is empty (null)
91. * @see Session#addUpgrade(UpgradeTileModel) for adding to player's collection
92. */
93. public void tryPurchase(UpgradeTileModel tileClickedOn)
94. {
95.     if (tileClickedOn == null) {
96.         throw new IllegalArgumentException("Cannot purchase null tile");
97.     }
98.
99.     if (session.getMoneyProperty().get() >= tileClickedOn.getCost())
100.    {
101.        currentInShop.remove(tileClickedOn);
102.        session.addUpgrade(tileClickedOn);
103.        session.modifyMoney(-tileClickedOn.getCost());
104.
105.        this.logger.logMessage(String.format("Purchased %s for $%.2f",
106.            tileClickedOn.getName(),
107.            tileClickedOn.getCost()));
108.    }
109.    else
110.    {
111.        // TODO: some indication that they can't afford. money sfx or red currency emphasis.
112.        this.logger.logMessage(String.format("Cannot afford %s (costs $%.2f, have $%f)",
113.            tileClickedOn.getName(),
114.            tileClickedOn.getCost(),
115.            session.getMoneyProperty().get()));
116.    }
117. }
118.
119. /**
120. * exists shop and increments level requirement for the next level.
121. */
122. public void onNextLevelPressed()
123. {
124.     sceneManager.switchScene(GameScene.LEVEL);
125. }
126. }

```

com.example.project.models.gameScreens

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C LevelModel | | 85% | | 77% | 18 | 58 | 20 | 125 | 10 | 38 | 0 | 1 |
| C DailyRewardModel | | 0% | | 0% | 5 | 5 | 10 | 10 | 3 | 3 | 1 | 1 |
| C ScoreChimePlayer | | 75% | | 0% | 3 | 4 | 7 | 19 | 2 | 3 | 0 | 1 |
| C DailyRewardType | | 0% | | n/a | 1 | 1 | 4 | 4 | 1 | 1 | 1 | 1 |
| C GameScreenModel | | 70% | | n/a | 2 | 3 | 2 | 7 | 2 | 3 | 0 | 1 |
| C ShopModel | | 100% | | 100% | 0 | 11 | 0 | 34 | 0 | 8 | 0 | 1 |
| C LoginModel | | 100% | | 95% | 1 | 16 | 0 | 39 | 0 | 6 | 0 | 1 |
| C MainMenuModel | | 100% | | n/a | 0 | 3 | 0 | 6 | 0 | 3 | 0 | 1 |
| Total | 193 of 1,131 | 82% | 15 of 71 | 78% | 30 | 101 | 43 | 244 | 18 | 65 | 2 | 8 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.models.gameScreens

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| LevelModel.java | 85% | 85% | 77% | 77% | 18 | 58 | 20 | 125 | 10 | 38 | 0 | 1 |
| DailyRewardModel.java | 0% | 0% | 0% | 0% | 5 | 5 | 10 | 10 | 3 | 3 | 1 | 1 |
| ScoreChimePlayer.java | 75% | 75% | 0% | 0% | 3 | 4 | 7 | 19 | 2 | 3 | 0 | 1 |
| DailyRewardType.java | 0% | 0% | | n/a | 1 | 1 | 4 | 4 | 1 | 1 | 1 | 1 |
| GameScreenModel.java | 70% | 70% | | n/a | 2 | 3 | 2 | 7 | 2 | 3 | 0 | 1 |
| ShopModel.java | 100% | 100% | 100% | 100% | 0 | 11 | 0 | 34 | 0 | 8 | 0 | 1 |
| LoginModel.java | 100% | 100% | 95% | 95% | 1 | 16 | 0 | 39 | 0 | 6 | 0 | 1 |
| MainMenuModel.java | 100% | 100% | | n/a | 0 | 3 | 0 | 6 | 0 | 3 | 0 | 1 |
| Total | 193 of 1,131 | 82% | 15 of 71 | 78% | 30 | 101 | 43 | 244 | 18 | 65 | 2 | 8 |

Created with JaCoCo 0.8.12.202403310830

DefinitionPopup

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| setIsDefinitionActive(boolean) | | 85% | | 75% | 1 | 3 | 1 | 5 | 0 | 1 |
| DefinitionPopup() | | 100% | | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| setPopup(String) | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| definitionProperty() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| wordProperty() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getIsDefinitionActive() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| getFXMLPath() | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 3 of 73 | 95% | 1 of 4 | 75% | 1 | 9 | 1 | 18 | 0 | 7 |

Created with JaCoCo 0.8.12.202403310830

DefinitionPopup.java

```

1. package com.example.project.models.popups;
2. import com.example.project.services.sqlite.dAOs.DictionaryDAO;
3. import javafx.beans.property.*;
4.
5. /**
6. * Represents Definition window called when a word is played
7. */
8. public class DefinitionPopup extends PopupModel {
9.
10.    private final ReadOnlyStringWrapper definition = new ReadOnlyStringWrapper("lorem ipsum");
11.
12.    private final ReadOnlyStringWrapper word = new ReadOnlyStringWrapper("lorem");
13.
14.    private final ReadOnlyBooleanWrapper isDefinitionActive = new ReadOnlyBooleanWrapper(false);
15.
16.    /**
17.     * @return returns property of definition
18.     */
19.    public ReadOnlyStringProperty definitionProperty(){
20.        return definition.getReadOnlyProperty();
21.    }
22.
23.    /**
24.     * @return returns property of word
25.     */
26.    public ReadOnlyStringProperty wordProperty(){
27.        return word.getReadOnlyProperty();
28.    }
29.
30.    /**
31.     * @return returns boolean
32.     */
33.    public ReadOnlyBooleanProperty getIsDefinitionActive(){
34.    {
35.        return isDefinitionActive.getReadOnlyProperty();
36.    }
37.
38.    /**
39.     * toggle isDefinitionActive
40.     * @param newState current state of definition window
41.     */
42.    public void setIsDefinitionActive(boolean newState)
43.    {
44.        if(this.isDefinitionActive.get() == newState) { return; }
45.        if (newState){getPaperSoundPlayer().replay();}
46.        else {getReversePaperSoundPlayer().replay();}
47.        this.isDefinitionActive.set(newState);
48.    }
49.
50.    /**
51.     * @param currentWord Word, when OnPlayButton is passed
52.     */
53.    public void setPopup(String currentWord){
54.        word.set(currentWord);
55.        DictionaryDAO dictionary = new DictionaryDAO();
56.        currentWord = currentWord.toLowerCase();
57.        definition.set(dictionary.getWordDefinition(currentWord));
58.    }
59.
60.    // TODO this is only used here are there other classes with popups? Otherwise can remove parent Popup class and just have a single concrete DefinitionPopup model and controller.
61.    @Override
62.    public String getFXMLPath() {
63.        return "/com/example/project/PopUps/DefinitionView.fxml";
64.    }
65. }
```

ScoreTimeline

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| animateTotalScore(int, int, Label, long) | | 0% | n/a | | 1 | 1 | 10 | 10 | 1 | 1 |
| lambda\$animateTotalScore\$0(IntegerProperty, int, Label, ActionEvent) | | 0% | | 0% | 2 | 2 | 4 | 4 | 1 | 1 |
| lambda\$animateTotalScore\$1(Label, int, ActionEvent) | | 0% | n/a | | 1 | 1 | 2 | 2 | 1 | 1 |
| ScoreTimeline() | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| Total | 70 of 70 | 0% | 2 of 2 | 0% | 5 | 5 | 17 | 17 | 4 | 4 |

Created with JaCoCo 0.8.12 202403310830

PopupModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cqty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| PopupModel() | 100% | | n/a | | 0 | 1 | 0 | 3 | 0 | 1 |
| getPaperSoundPlayer() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| getReversePaperSoundPlayer() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 21 | 100% | 0 of 0 | n/a | 0 | 3 | 0 | 5 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

PopupModel.java

```
1. package com.example.project.models.popups;
2.
3. import com.example.project.services.sound.GameSoundPlayer;
4.
5. /**
6.  * Parent class for popup windows
7. */
8. public abstract class PopupModel {
9.
10.    /**
11.     * gets FXML path
12.     * @return string path
13.    */
14.    public abstract String getFXMLPath();
15.
16.    protected final GameSoundPlayer paperSoundPlayer = new GameSoundPlayer("/com/example/project/Sounds/PaperFlip.wav");
17.
18.    protected final GameSoundPlayer reversePaperSoundPlayer = new GameSoundPlayer("/com/example/project/Sounds/PaperFlipReversed.wav");
19.    /**
20.     * gets paper sound player
21.     * @return sound player
22.    */
23.    public GameSoundPlayer getPaperSoundPlayer() {return paperSoundPlayer;}
24.
25.    /**
26.     * gets reversed paper sound player
27.     * @return sound player
28.    */
29.    public GameSoundPlayer getReversePaperSoundPlayer() {return reversePaperSoundPlayer;}
30. }
```

com.example.project.models.popups

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|-----------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C DefinitionPopup | | 95% | | 75% | 1 | 9 | 1 | 18 | 0 | 7 | 0 | 1 |
| C PopupModel | | 100% | | n/a | 0 | 3 | 0 | 5 | 0 | 3 | 0 | 1 |
| Total | 3 of 94 | 96% | 1 of 4 | 75% | 1 | 12 | 1 | 23 | 0 | 10 | 0 | 2 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.models.popups

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|--------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| DefinitionPopup.java | | 95% | | 75% | 1 | 9 | 1 | 18 | 0 | 7 | 0 | 1 |
| PopupModel.java | | 100% | | n/a | 0 | 3 | 0 | 5 | 0 | 3 | 0 | 1 |
| Total | 3 of 94 | 96% | 1 of 4 | 75% | 1 | 12 | 1 | 23 | 0 | 10 | 0 | 2 |

Created with JaCoCo 0.8.12.202403310830

LetterTileGroup

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| updateVisuals() | | 100% | | 100% | 0 | 3 | 0 | 6 | 0 | 1 |
| LetterTileGroup(int, Pane, ReadOnlyListProperty, Consumer, List, TileControllerFactory) | | 100% | | n/a | 0 | 1 | 0 | 7 | 0 | 1 |
| createEmptySlots() | | 100% | | 100% | 0 | 2 | 0 | 5 | 0 | 1 |
| loadEmptySlotIntoContainer() | | 100% | | n/a | 0 | 1 | 0 | 4 | 0 | 1 |
| LetterTileGroup(int, Pane, ReadOnlyListProperty, Consumer) | | 100% | | n/a | 0 | 1 | 0 | 5 | 0 | 1 |
| LetterTileGroup(int, Pane, ReadOnlyListProperty, Consumer, List) | | 100% | | n/a | 0 | 1 | 0 | 3 | 0 | 1 |
| lambda\$new\$1(List, ObservableValue, ObservableList, ObservableList) | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| lambda\$new\$0(List, ObservableValue, ObservableList, ObservableList) | | 100% | | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 136 | 100% | 0 of 6 | 100% | 0 | 11 | 0 | 29 | 0 | 8 |

Created with JaCoCo 0.8.12.202403310830

LetterTileGroup.java

```
1. package com.example.project.models.tileGroups;
2.
3. import com.example.project.controllers.tiles.EmptyTileSlotController;
4. import com.example.project.controllers.tiles.LetterTileController;
5. import com.example.project.models.tiles.EmptyTileSlotModel;
6. import com.example.project.models.tiles.LetterTileModel;
7. import com.example.project.controllers.tiles.TileControllerFactory;
8. import javafx.beans.property.ReadOnlyListProperty;
9. import javafx.scene.layout.Pane;
10. import java.util.ArrayList;
11. import java.util.List;
12. import java.util.function.Consumer;
13.
14. /**
15. * tile group that observes an observable list and updates the ui Tile Controller's nodes into their EmptyTileSlotController nodes.
16. */
17. public class LetterTileGroup extends TileGroup<LetterTileModel, LetterTileController>
18. {
19.     private final List<EmptyTileSlotController> tileSlots = new ArrayList<>();
20.     private final int numberofEmptyTileSlots;
21.
22.     protected LetterTileGroup(int numberofEmptyTileSlots, Pane container,
23.                               ReadOnlyListProperty<LetterTileModel> observedList,
24.                               Consumer<LetterTileController> onClickHandler,
25.                               List<Runnable> afterSyncActions,
26.                               TileControllerFactory factory)
27.     {
28.         super(container, onClickHandler, LetterTileController.class, observedList);
29.         this.tileControllerFactory = factory;
30.         observedList.addListener((obs, oldVal, newVal) -> afterSyncActions.forEach(Runnable::run));
31.         this.numberofEmptyTileSlots = numberofEmptyTileSlots;
32.         createEmptySlots();
33.     }
34.
35.     /**
36.      * Constructor
37.      * @param numberofEmptyTileSlots number of max tiles in group (empty slots)
38.      * @param container container to place all in.
39.      * @param observedList the observed list.
40.      * @param onClickHandler On tile click action.
41.      * @param afterSyncActions additional synchronisation actions that need to happen when this observed list changes.
42.      */
43.     public LetterTileGroup(int numberofEmptyTileSlots, Pane container,
44.                           ReadOnlyListProperty<LetterTileModel> observedList,
45.                           Consumer<LetterTileController> onClickHandler,
46.                           List<Runnable> afterSyncActions)
47.     {
48.         this(numberofEmptyTileSlots, container, observedList, onClickHandler);
49.         observedList.addListener((obs, oldVal, newVal) -> afterSyncActions.forEach(Runnable::run));
50.     }
51.
52.     /**
53.      * Constructor
54.      * @param numberofEmptyTileSlots number of max tiles in group (empty slots)
55.      * @param container container to place all in.
56.      * @param observedList the observed list.
57.      * @param onClickAction On tile click action.
58.      */
59.     public LetterTileGroup(int numberofEmptyTileSlots, Pane container,
60.                           ReadOnlyListProperty<LetterTileModel> observedList,
61.                           Consumer<LetterTileController> onClickAction)
62.     {
63.         super(container, onClickAction, LetterTileController.class, observedList);
64.         this.numberofEmptyTileSlots = numberofEmptyTileSlots;
65.         createEmptySlots();
66.     }
67.
68.     /**
69.      * Create empty slots for both word area and tile rack
70.      */
71.     private void createEmptySlots()
72.     {
73.         tileSlots.clear();
74.
75.         for (var i = 0; i < numberofEmptyTileSlots; i++)
76.         {
77.             var emptyTileController = loadEmptySlotIntoContainer();
78.             tileSlots.add(emptyTileController);
79.         }
80.     }
81.
82.     /**
83.      * Load a new empty slot into a container
84.      */
```

```
85.     private EmptyTileSlotController loadEmptySlotIntoContainer()
86.     {
87.         // TODO: this model should store empty tiles and the controller of this should observe those.
88.         var emptyTile = new EmptyTileSlotModel();
89.         EmptyTileSlotController controller = tileControllerFactory.createTileController(emptyTile, EmptyTileSlotController.class);
90.         container.getChildren().add(controller.getRoot());
91.         return controller;
92.     }
93.
94.     @Override
95.     protected void updateVisuals()
96.     {
97.         // update visuals
98.         for (EmptyTileSlotController rowsEmptyTile : tileSlots) {
99.             rowsEmptyTile.setLetter(null);
100.        }
101.
102.         for (int i = 0; i < tileControllers.size(); i++){
103.             tileSlots.get(i).setLetter(tileControllers.get(i));
104.         }
105.     }
106. }
```

Created with JaCoCo 0.8.12.202403310830

TileGroup

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| ● lambda\$recreateControllers\$1(TileController, MouseEvent) | | 0% | n/a | | 1 | 1 | 1 | 1 | 1 | 1 |
| ● recreateControllers() | | 100% | | 75% | 1 | 3 | 0 | 8 | 0 | 1 |
| ● TileGroup(Pane, Class, ReadOnlyListProperty) | | 100% | n/a | | 0 | 1 | 0 | 9 | 0 | 1 |
| ● TileGroup(Pane, Consumer, Class, ReadOnlyListProperty) | | 100% | n/a | | 0 | 1 | 0 | 3 | 0 | 1 |
| ● syncTiles() | | 100% | n/a | | 0 | 1 | 0 | 3 | 0 | 1 |
| ● getControllers() | | 100% | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| ● lambda\$new\$0(ObservableValue, ObservableList, ObservableList) | | 100% | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 5 of 92 | 94% | 1 of 4 | 75% | 2 | 9 | 0 | 24 | 1 | 7 |

Created with JaCoCo 0.8.12.202403310830

TileGroup.java

```
1. package com.example.project.models.tileGroups;
2.
3. import com.example.project.controllers.tiles.TileController;
4. import com.example.project.models.tiles.TileModel;
5. import com.example.project.controllers.tiles.TileControllerFactory;
6. import javafx.beans.property.ReadOnlyListProperty;
7. import javafx.scene.layout.Pane;
8.
9. import java.util.ArrayList;
10. import java.util.List;
11. import java.util.function.Consumer;
12.
13. /**
14. * Tile Group abstract class holds a row or column of tiles.
15. * @param <modelType> tile model type.
16. * @param <controllerType> controller type.
17. */
18. public abstract class TileGroup<modelType extends TileModel, controllerType extends TileController<modelType>>
19. {
20.     protected final Pane container;
21.     protected final List<controllerType> tileControllers = new ArrayList<>();
22.     protected TileControllerFactory tileControllerFactory = new TileControllerFactory();
23.     protected Consumer<controllerType> onClickAction = null;
24.     private final Class<controllerType> controllerClass;
25.     private final ReadOnlyListProperty<modelType> observedModels;
26.
27.     TileGroup(Pane container, Consumer<controllerType> onClickAction, Class<controllerType> tileControllerClass, ReadOnlyListProperty<modelType> observedList)
28.     {
29.         this(container, tileControllerClass, observedList);
30.         this.onClickAction = onClickAction;
31.     }
32.
33.     TileGroup(Pane container, Class<controllerType> tileControllerClass, ReadOnlyListProperty<modelType> observedList)
34.     {
35.         this.container = container;
36.         this.controllerClass = tileControllerClass;
37.         this.observedModels = observedList;
38.         this.observedModels.addListener((obs, oldVal, newVal) -> syncTiles());
39.     }
40.
41.     private void recreateControllers()
42.     {
43.         tileControllers.clear();
44.         for (modelType tile : this.observedModels)
45.         {
46.             controllerType controller = tileControllerFactory.createTileController(tile, controllerClass);
47.             tileControllers.add(controller);
48.
49.             if (onClickAction != null){
50.                 controller.getRoot().setOnMouseClicked(e -> onClickAction.accept(controller));
51.             }
52.         }
53.     }
54.
55.     protected abstract void updateVisuals();
56.
57. /**
58. * Sync tiles in this tile groups controller.
59. */
60. public void syncTiles()
61. {
62.     recreateControllers();
63.     updateVisuals();
64. }
65.
66. /**
67. * Gets the groups tile controllers.
68. * @return the letter tile groups controllers
69. */
70. public List<controllerType> getControllers(){
71.     return tileControllers;
72. }
```

UpgradeTileGroup

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| UpgradeTileGroup(Pane, ReadOnlyListProperty) | | 0% | n/a | 1 | 1 | 2 | 2 | 1 | 1 | |
| lambda\$trackDuplicates\$0(Object, Integer) | | 60% | | 50% | 1 | 2 | 0 | 1 | 0 | 1 |
| updateVisuals() | | 98% | | 66% | 4 | 7 | 1 | 22 | 0 | 1 |
| trackDuplicates(List) | | 100% | | 100% | 0 | 2 | 0 | 5 | 0 | 1 |
| UpgradeTileGroup(Pane, ReadOnlyListProperty, Consumer) | | 100% | n/a | 0 | 1 | 0 | 2 | 0 | 0 | 1 |
| Total | 11 of 141 | 92% | 5 of 16 | 68% | 6 | 13 | 3 | 31 | 1 | 5 |

Created with JaCoCo 0.8.12.202403310830

UpgradeTileGroup.java

```

1. package com.example.project.models.tileGroups;
2.
3. import com.example.project.controllers.tiles.UpgradeTileController;
4. import com.example.project.models.tiles.UpgradeTileModel;
5. import javafx.beans.property.ReadOnlyListProperty;
6. import javafx.scene.layout.Pane;
7.
8. import java.util.*;
9. import java.util.function.Consumer;
10.
11. /**
12. * tile group that observes an observable list and updates the ui nodes for some game tiles.
13. */
14. public class UpgradeTileGroup extends TileGroup<UpgradeTileModel, UpgradeTileController>
15. {
16.     /**
17.      * Constructor.
18.      * @param container container all tiles are in.
19.      * @param observedList the observed list that when changes this tile group will sync.
20.      * @param onClickAction additional synchronisation actions that need to be called.
21.      */
22.     public UpgradeTileGroup(Pane container, ReadOnlyListProperty<UpgradeTileModel> observedList, Consumer<UpgradeTileController> onClickAction)
23.     {
24.         super(container, onClickAction, UpgradeTileController.class, observedList);
25.     }
26.
27.     /**
28.      * Constructor with no additional sync action parameter.
29.      * @param container container tiles are in.
30.      * @param observedList the observed list that when changes this tile group will sync.
31.      */
32.     public UpgradeTileGroup(Pane container, ReadOnlyListProperty<UpgradeTileModel> observedList)
33.     {
34.         super(container, UpgradeTileController.class, observedList);
35.     }
36.
37.     /**
38.      * Makes hashmap that tracks the count of each element in list
39.      * @param list the input list of elements
40.      * @param <T> the type of elements in the list
41.      * @return a map for # of dupes for each unique element
42.      */
43.     public static <T> Map<T, Integer> trackDuplicates(List<T> list){
44.         Map<T, Integer> countMap = new HashMap<>();
45.
46.         for (T element : list) {
47.             countMap.compute(element, (key, count) -> (count == null) ? 1 : count + 1);
48.         }
49.
50.         return countMap;
51.     }
52.
53.     @Override
54.     protected void updateVisuals()
55.     {
56.         List<UpgradeTileModel> allModels = new ArrayList<>();
57.         for (var controller : tileControllers) {
58.             allModels.add(controller.getModel());
59.         }
60.
61.         Map<UpgradeTileModel, Integer> countMap = trackDuplicates(allModels);
62.
63.         Set<UpgradeTileModel> uniqueModels = countMap.keySet();
64.
65.         Set<UpgradeTileController> usedControllers = new HashSet<>();
66.
67.
68.
69.         container.getChildren().clear();
70.
71.         for (var uniqueModel : uniqueModels)
72.         {
73.             UpgradeTileController controllerToUse = null;
74.             for (var controller : tileControllers) {
75.                 if (controller.getModel().equals(uniqueModel) && !usedControllers.contains(controller)) {
76.                     controllerToUse = controller;
77.                     usedControllers.add(controller); // Mark as used
78.                     break;
79.                 }
80.             }
81.
82.             if (controllerToUse != null) {
83.                 int count = countMap.get(uniqueModel);
84.                 controllerToUse.updateCount(count);
85.                 container.getChildren().add(controllerToUse.getRoot());
86.             }
87.         }
88.     }
89. }
```

ScoreTimeline.java

```
1. package com.example.project.controllers.gameScreens.animations;
2. import javafx.animation.KeyFrame;
3. import javafx.animation.Timeline;
4. import javafx.beans.property.IntegerProperty;
5. import javafx.beans.property.SimpleIntegerProperty;
6. import javafx.scene.control.Label;
7. import javafx.util.Duration;
8.
9. /**
10. * Class for total score counter animation
11. */
12. public class ScoreTimeline {
13.     /**
14.      * @param totalDurationMillis total duration
15.      * @param startScore int from current total score
16.      * @param endScore int from calculated current score
17.      * @param currentScoreLabel Label for total score
18.      * @return timeline of total score counter
19.     */
20.     public Timeline animateTotalScore(int startScore, int endScore, Label currentScoreLabel, long totalDurationMillis) {
21.         // Total scoring timeline
22.         Timeline scoreTimeline = new Timeline();
23.         IntegerProperty currentScore = new SimpleIntegerProperty(startScore);
24.
25.         int scoreDifference = endScore - startScore;
26.
27.         int steps = scoreDifference;
28.
29.         double durationPerStepMillis = (double) totalDurationMillis / steps;
30.
31.         scoreTimeline.getKeyFrames().add(
32.             new KeyFrame(Duration.millis(durationPerStepMillis), event -> {
33.                 if (currentScore.get() < endScore) {
34.                     currentScore.set(currentScore.get() + 1);
35.                     currentScoreLabel.setText(String.valueOf(currentScore.get()));
36.                 }
37.             })
38.         );
39.         scoreTimeline.setCycleCount(steps);
40.
41.         scoreTimeline.setOnFinished(event -> {
42.             currentScoreLabel.setText(String.valueOf(endScore));
43.         });
44.
45.         return scoreTimeline;
46.     }
47. }
```

com.example.project.models.tileGroups

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| C UpgradeTileGroup | | 92% | | 68% | 6 | 13 | 3 | 31 | 1 | 5 | 0 | 1 |
| C TileGroup | | 94% | | 75% | 2 | 9 | 0 | 24 | 1 | 7 | 0 | 1 |
| C LetterTileGroup | | 100% | | 100% | 0 | 11 | 0 | 29 | 0 | 8 | 0 | 1 |
| Total | 16 of 369 | 95% | 6 of 26 | 76% | 8 | 33 | 3 | 84 | 2 | 20 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

com.example.project.models.tileGroups

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods | Missed | Classes |
|---------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|--------|---------|
| UpgradeTileGroup.java | | 92% | | 68% | 6 | 13 | 3 | 31 | 1 | 5 | 0 | 1 |
| TileGroup.java | | 94% | | 75% | 2 | 9 | 0 | 24 | 1 | 7 | 0 | 1 |
| LetterTileGroup.java | | 100% | | 100% | 0 | 11 | 0 | 29 | 0 | 8 | 0 | 1 |
| Total | 16 of 369 | 95% | 6 of 26 | 76% | 8 | 33 | 3 | 84 | 2 | 20 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830

EmptyTileSlotModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| EmptyTileSlotModel() | 100% | | n/a | | 0 | 1 | 0 | 2 | 0 | 1 |
| getTile() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| setTile(LetterTileModel) | 100% | | n/a | | 0 | 1 | 0 | 2 | 0 | 1 |
| getFXMLPath() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 21 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 6 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

EmptyTileSlotModel.java

```
1. package com.example.project.models.tiles;
2.
3.
4. import javafx.beans.property.SimpleObjectProperty;
5.
6. /**
7. * An empty tile slot model. Empty tile slots store a letter tile.
8. */
9. public class EmptyTileSlotModel extends TileModel
10.
11.     private SimpleObjectProperty<LetterTileModel> currentTile = new SimpleObjectProperty<>(null);
12.
13.     /**
14.     * get letter tile in slot.
15.     * @return letter tile.
16.     */
17.     public LetterTileModel getTile(){
18.         return this.currentTile.get();
19.     }
20.
21.     /**
22.     * Set a new tile.
23.     * @param tile letter tile.
24.     */
25.     public void setTile(LetterTileModel tile)
26.     {
27.         this.currentTile.set(tile);
28.     }
29.
30.     @Override
31.     public String getFXMLPath() {
32.         return "/com/example/project/SingleTiles/emptyTileSlot.fxml";
33.     }
34. }
```

LetterTileModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|--|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| • LetterTileModel(Character) | 100% | | n/a | | 0 | 1 | 0 | 5 | 0 | 1 |
| • getLetter() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| • getValue() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| • getFXMLPath() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 28 | 100% | 0 of 0 | n/a | 0 | 4 | 0 | 8 | 0 | 4 |

Created with JaCoCo 0.8.12.202403310830

LetterTileModel.java

```
1. package com.example.project.models.tiles;
2.
3.
4. /**
5.  * Represents the letter tiles the player plays holds, redraws.
6. */
7. public class LetterTileModel extends TileModel
8. {
9.     private final Character letter;
10.
11.    private final Integer value;
12.
13.    /**
14.     * @param newLetter letter
15.     */
16.    public LetterTileModel(Character newLetter)
17.    {
18.        this.letter = Character.toUpperCase(newLetter);
19.        ScrabbleTileProvider provider = new ScrabbleTileProvider();
20.        this.value = provider.getValue(letter);
21.    }
22.
23.    /**
24.     * @return returns the tiles character.
25.     */
26.    public Character getLetter() {
27.        return letter;
28.    }
29.
30.    /**
31.     * @return base value of scrabble letter score.
32.     */
33.    public Integer getValue(){
34.        return this.value;
35.    }
36.
37.    @Override
38.    public String getFXMLPath() {
39.        return "/com/example/project/SingleTiles/letterTileView.fxml";
40.    }
41.}
```

ScrabbleTileProvider.ScrabbleTileData

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| ScrabbleTileProvider.ScrabbleTileData(int, int) | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 9 | 100% | 0 of 0 | n/a | 0 | 1 | 0 | 1 | 0 | 1 |

Created with JaCoCo 0.8.12.202403310830

ScrabbleTileProvider

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| ScrabbleTileProvider() | 100% | 100% | 100% | 100% | 0 | 2 | 0 | 38 | 0 | 1 |
| getValue(Character) | 100% | 100% | 100% | 100% | 0 | 2 | 0 | 2 | 0 | 1 |
| drawRandomTile() | 100% | 100% | n/a | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| static {...} | 100% | 100% | n/a | n/a | 0 | 1 | 0 | 1 | 0 | 1 |
| seedRandomNumberGenerator(long) | 100% | 100% | n/a | n/a | 0 | 1 | 0 | 2 | 0 | 1 |
| Total | 0 of 380 | 100% | 0 of 4 | 100% | 0 | 7 | 0 | 45 | 0 | 5 |

Created with JaCoCo 0.8.12.202403310830

ScrabbleTileProvider.java

```
1. package com.example.project.models.tiles;
2.
3. import java.util.Map;
4. import java.util.TreeMap;
5. import java.util.Random;
6.
7. /**
8. * Static class to store the scrabble base letter value.
9. */
10. public class ScrabbleTileProvider {
11.     /**
12.      * Constructor.
13.     */
14.     public ScrabbleTileProvider() {
15.         /**
16.          // Calculate total number of tiles by summing frequencies
17.          double totalTiles = this.baseLetterData.values().stream()
18.              .mapToInt(ScrabbleTileData::frequency)
19.              .sum();
20.
21.          double cumulativeFrequency = 0.0;
22.
23.          // Populate the cdfMap
24.          for (Map.Entry<Character, ScrabbleTileData> entry : baseLetterData.entrySet()) {
25.              cumulativeFrequency += entry.getValue().frequency();
26.              cdfMap.put(cumulativeFrequency / totalTiles, entry.getKey());
27.          }
28.      }
29. }
30.
31. /**
32. * @param value value of scrabble letter
33. * @param frequency population of tile
34. */
35. public record ScrabbleTileData(int value, int frequency)
36. {
37.
38.     private final TreeMap<Double, Character> cdfMap = new TreeMap<>();
39.     private static final Random random = new Random();
40.
41.     protected static void seedRandomNumberGenerator(long newSeed){
42.         random.setSeed(newSeed);
43.     }
44.
45.     private final Map<Character, ScrabbleTileData> baseLetterData = Map.ofEntries(
46.         Map.entry('a', new ScrabbleTileData(1, 9)),
47.         Map.entry('b', new ScrabbleTileData(3, 2)),
48.         Map.entry('c', new ScrabbleTileData(3, 2)),
49.         Map.entry('d', new ScrabbleTileData(2, 4)),
50.         Map.entry('e', new ScrabbleTileData(1, 12)),
51.         Map.entry('f', new ScrabbleTileData(4, 2)),
52.         Map.entry('g', new ScrabbleTileData(2, 3)),
53.         Map.entry('h', new ScrabbleTileData(4, 2)),
54.         Map.entry('i', new ScrabbleTileData(1, 9)),
55.         Map.entry('j', new ScrabbleTileData(8, 1)),
56.         Map.entry('k', new ScrabbleTileData(5, 1)),
57.         Map.entry('l', new ScrabbleTileData(1, 4)),
58.         Map.entry('m', new ScrabbleTileData(3, 2)),
59.         Map.entry('n', new ScrabbleTileData(1, 6)),
60.         Map.entry('o', new ScrabbleTileData(1, 8)),
61.         Map.entry('p', new ScrabbleTileData(3, 2)),
62.         Map.entry('q', new ScrabbleTileData(10, 1)),
63.         Map.entry('r', new ScrabbleTileData(1, 6)),
64.         Map.entry('s', new ScrabbleTileData(1, 4)),
65.         Map.entry('t', new ScrabbleTileData(1, 6)),
66.         Map.entry('u', new ScrabbleTileData(1, 4)),
67.         Map.entry('v', new ScrabbleTileData(4, 2)),
68.         Map.entry('w', new ScrabbleTileData(4, 2)),
69.         Map.entry('x', new ScrabbleTileData(8, 1)),
70.         Map.entry('y', new ScrabbleTileData(4, 2)),
71.         Map.entry('z', new ScrabbleTileData(10, 1))
72. );
73.
74. /**
75. * Returns Integer of the letter according to scrabble
76. * @param letter letter to get value of
77. * @return Integer value of the letter, or 0 if not found
78. */
```

```
79.     public int getValue(Character letter)
80.     {
81.         ScrabbleTileData data = baseLetterData.get(Character.toLowerCase(letter));
82.         return (data != null) ? data.value() : 0;
83.     }
84.
85.    /**
86.     * @return randomly selected tile based on CDF
87.     */
88.    public Character drawRandomTile() {
89.        // Generate a random float between 0.0 (inclusive) and 1.0 (exclusive)
90.        double randomFloat = random.nextDouble();
91.
92.        // Find the first entry in the CDF map whose key is greater than or equal to the random float.
93.        return cdfMap.tailMap(randomFloat, true).firstEntry().getValue();
94.    }
95.
96. }
```

Created with JaCoCo 0.8.12.202403310830

TileModel

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed | Cxty | Missed | Lines | Missed | Methods |
|---------------------------------------|---------------------|------|-----------------|------|--------|------|--------|-------|--------|---------|
| TileModel() | 100% | | n/a | | 0 | 1 | 0 | 3 | 0 | 1 |
| getHoverSoundPlayer() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| getClackSoundPlayer() | 100% | | n/a | | 0 | 1 | 0 | 1 | 0 | 1 |
| Total | 0 of 23 | 100% | 0 of 0 | n/a | 0 | 3 | 0 | 5 | 0 | 3 |

Created with JaCoCo 0.8.12.202403310830