



**libGDX**

<https://github.com/libgdx/libgdx/wiki/>

**Rafael Vieira Coelho**

rafael.coelho@farroupilha.ifrs.edu.br





## Parte 6 - Jogo Completo

- Versão 1 - Construção do Projeto
- Versão 2 - Desenha icicle
- Versão 3 - Desenha Jogador
- Versão 4 - Controle de Teclado (setas)
- Versão 5 - Adiciona Icicles
- Versão 6 - Remove Icicles que somem da tela
- Versão 7 - Detecta Colisão
- Versão 8 - Adiciona o HUD
- **Versão 9 - Adiciona níveis de dificuldade**
- Versão 10 - Adiciona seleção de nível de dificuldade

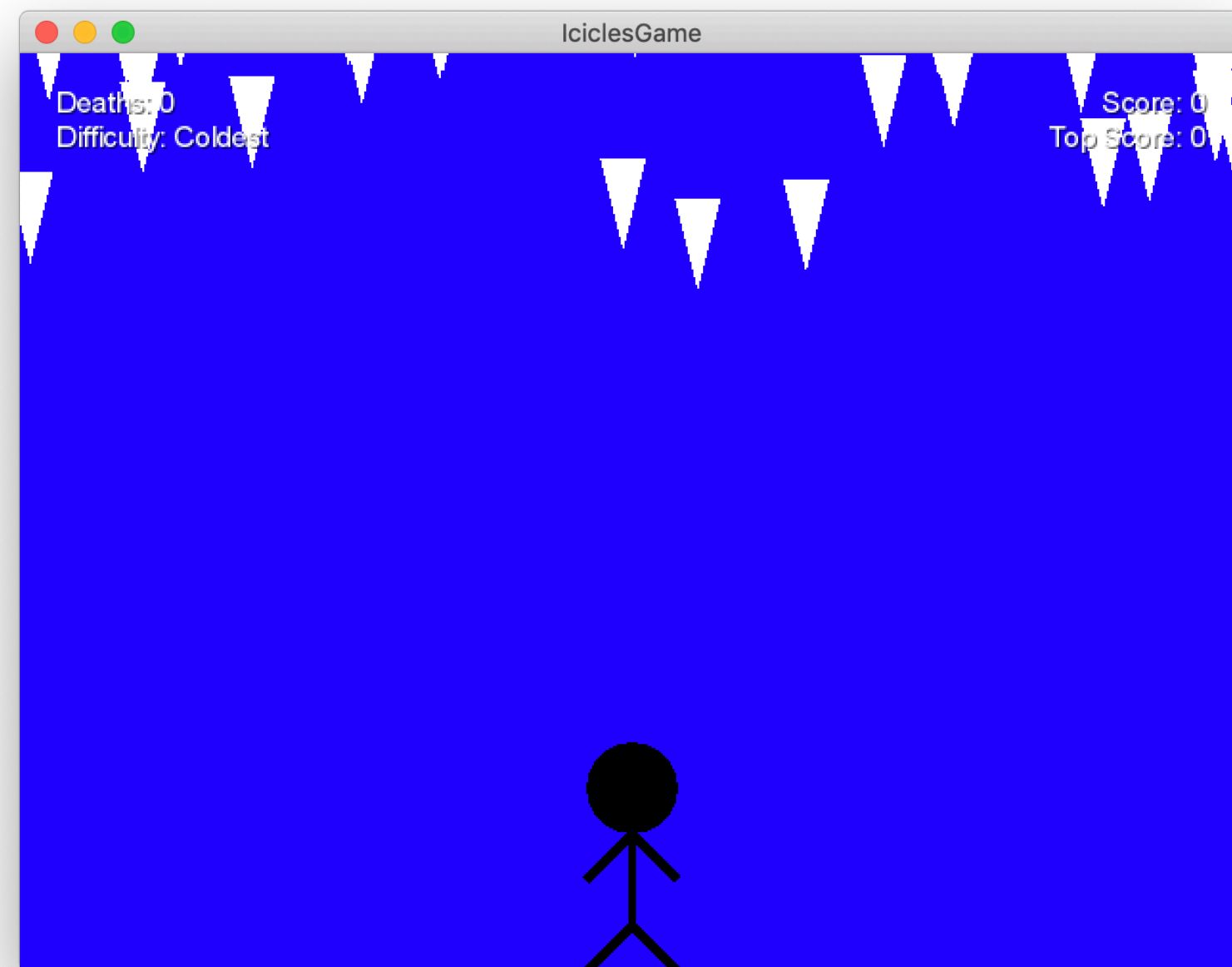
<https://github.com/libgdx/libgdx/wiki/>



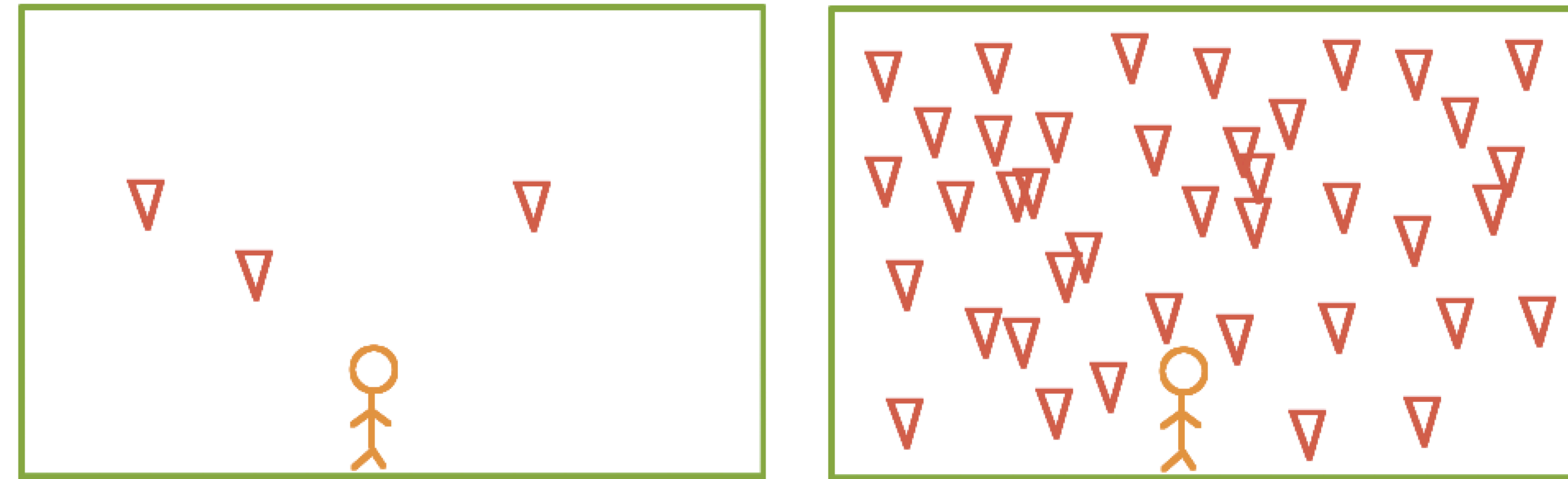


# Adicionando Níveis de Dificuldade

1. Precisamos criar uma enum que represente os diferentes níveis de dificuldade
2. Adicionar o nível de dificuldade ao construtor de IciclesScreen
3. Variar a velocidade com base no nível de dificuldade
5. Mostrar o nível atual no HUD



## Add Difficulty Levels



- ☐ Add Difficulty enum
- ☐ Add difficulty to IciclesScreen constructor
- ☐ Experiment with different icicle spawn rates

# Classe Constants.java

- Criamos as constantes para os 3 níveis
- E para as velocidades de criação de icicles com base no nível
- Por fim, a enum que representa a combinação do nível e de sua velocidade

```
Constants.java x Icicles.java x IciclesGame.java x IciclesScreen.java x
28
29 // TODO: Create constants for difficulty labels ("Cold", "Colder", "Coldest")
30 public static final String EASY_LABEL = "Cold";
31 public static final String MEDIUM_LABEL = "Colder";
32 public static final String HARD_LABEL = "Coldest";
33
34 // TODO: Create constants for the icicle spawn rates for the various difficulties
35 public static final float EASY_SPAWNS_PER_SECOND = 5;
36 public static final float MEDIUM_SPAWNS_PER_SECOND = 15;
37 public static final float HARD_SPAWNS_PER_SECOND = 25;
38
39 // TODO: Create Difficulty enum holding the spawn rate and label for each difficulty
40 public enum Difficulty {
41     EASY(EASY_SPAWNS_PER_SECOND, EASY_LABEL),
42     MEDIUM(MEDIUM_SPAWNS_PER_SECOND, MEDIUM_LABEL),
43     HARD(HARD_SPAWNS_PER_SECOND, HARD_LABEL);
44
45     float spawnRate;
46     String label;
47
48     Difficulty(float spawnRate, String label) {
49         this.spawnRate = spawnRate;
50         this.label = label;
51     }
52 }
53 }
```



# Classe IciclesScreen.java

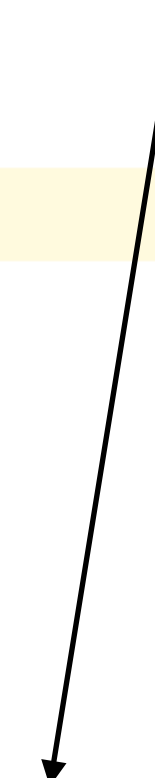
- Adicionamos o nível de dificuldade como atributo e recebemos no construtor a dificuldade inicial
- E no método **show()** passamos o nível de dificuldade atual para o construtor de **Icicles**.

```
IciclesScreen.java x Icicles.java x IciclesGame.java x
37 // TODO: Accept a Difficulty in the constructor
38 public IciclesScreen(Difficulty difficulty) {
39     // TODO: Set Difficulty
40     this.difficulty = difficulty;
41 }
42
43 @Override
44 public void show() {
45     iciclesViewport = new ExtendViewport(Constants.WORLD_SIZE,
46         Constants.WORLD_SIZE);
47     renderer = new ShapeRenderer();
48     renderer.setAutoShapeType(true);
49     hudViewport = new ScreenViewport();
50     batch = new SpriteBatch();
51     font = new BitmapFont();
52     font.getRegion().getTexture().setFilter(TextureFilter.Linear,
53         TextureFilter.Linear);
54     player = new Player(iciclesViewport);
55     // TODO: Initialize icicles with the difficulty
56     icicles = new Icicles(iciclesViewport, difficulty);
57     topScore = 0;
58 }
```

# Classe IciclesScreen.java

```
77  @Override
78  public void render(float delta) {
79      icicles.update(delta);
80      player.update(delta);
81      if (player.hitByIcicle(icicles)) {
82          icicles.init();
83      }
84
85      iciclesViewport.apply( centerCamera: true);
86      Gdx.gl.glClearColor(Constants.BACKGROUND_COLOR.r, Constants.BACKGROUND_COLOR.g, Constants.BACKGROUND_COLOR.b);
87      Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);
88
89      renderer.setProjectionMatrix(iciclesViewport.getCamera().combined);
90      renderer.begin(ShapeType.Filled);
91      icicles.render(renderer);
92      player.render(renderer);
93      renderer.end();
94
95      hudViewport.apply();
96      batch.setProjectionMatrix(hudViewport.getCamera().combined);
97      batch.begin();
98
99      topScore = Math.max(topScore, icicles.iciclesDodged);
100
101      // TODO: Show Difficulty level in the top left
102      font.draw(batch, str: "Deaths: " + player.deaths + "\nDifficulty: " + difficulty.label,
103                Constants.HUD_MARGIN, y: hudViewport.getWorldHeight() - Constants.HUD_MARGIN);
```

- Desenhamos o nível de dificuldade no HUD





# Classe Icicles.java

- Criamos o atributo do nível de dificuldade
- Recebemos um nível inicial no construtor
- E testamos a velocidade do nível de dificuldade na hora de criar novos icicles

```
Icicles.java x IciclesGame.java x
15 // TODO: Add a Difficulty
16 Difficulty difficulty;
17
18 int iciclesDodged;
19 DelayedRemovalArray<Icicle> icicleList;
20 Viewport viewport;
21
22 // TODO: Accept a difficulty in the constructor
23 public Icicles(Viewport viewport, Difficulty difficulty) {
24     // TODO: Set difficulty
25     this.difficulty = difficulty;
26     this.viewport = viewport;
27     init();
28 }
29
30 public void init() {
31     icicleList = new DelayedRemovalArray<Icicle>(ordered: false, capacity: 100);
32     iciclesDodged = 0;
33 }
34
35 public void update(float delta) {
36     // TODO: Use the difficulty's spawn rate
37     if (MathUtils.random() < delta * difficulty.spawnRate) {
38         Vector2 newIciclePosition = new Vector2(
39             x: MathUtils.random() * viewport.getWorldWidth(),
40             viewport.getWorldHeight()
41         );
42         Icicle newIcicle = new Icicle(newIciclePosition);
43         icicleList.add(newIcicle);
44     }
```

# Classe IciclesGame.java

- Definimos o nível de dificuldade inicial do jogo

```
IciclesGame.java x
1 package com.udacity.gamedev.icicles;
2
3 import ...
4
5
6
7 public class IciclesGame extends Game {
8
9     @Override
10    public void create() {
11        // TODO: Create IciclesScreen with a Difficulty
12        setScreen(new IciclesScreen(Difficulty.HARD));
13    }
14 }
```