libGDX

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

Rafael Vieira Coelho

rafael.coelho@farroupilha.ifrs.edu.br

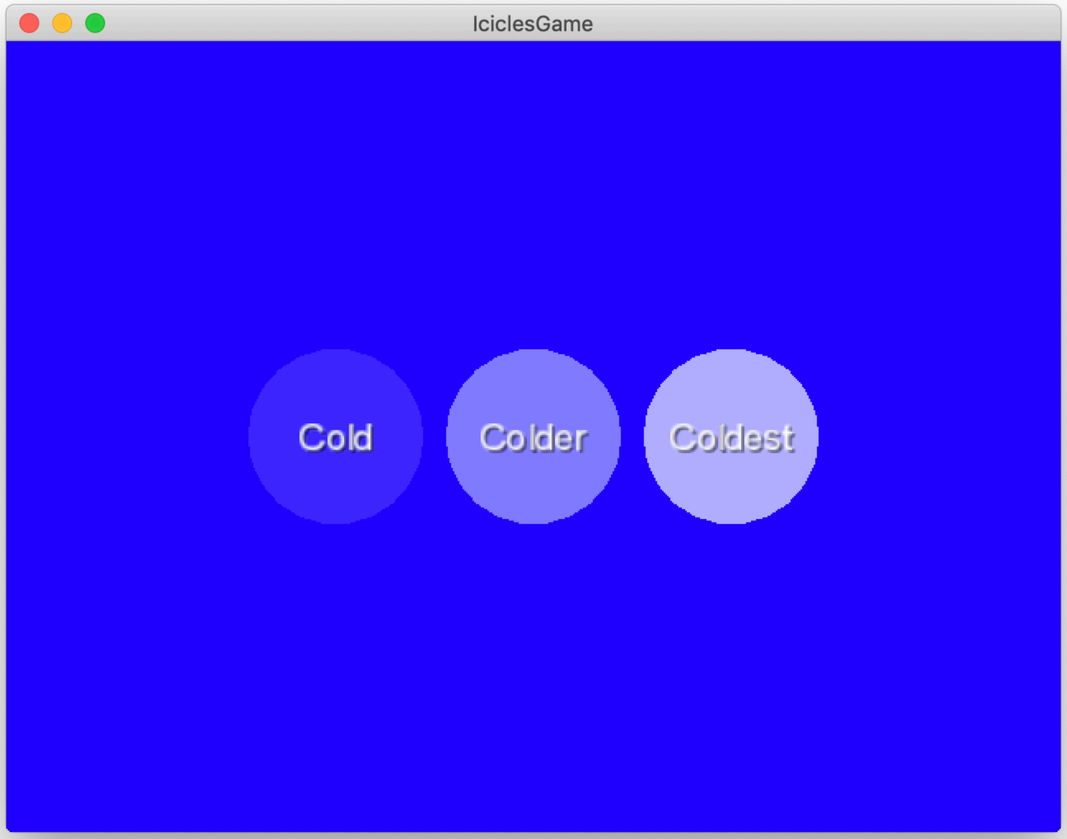
Parte 6 - Jogo Completo

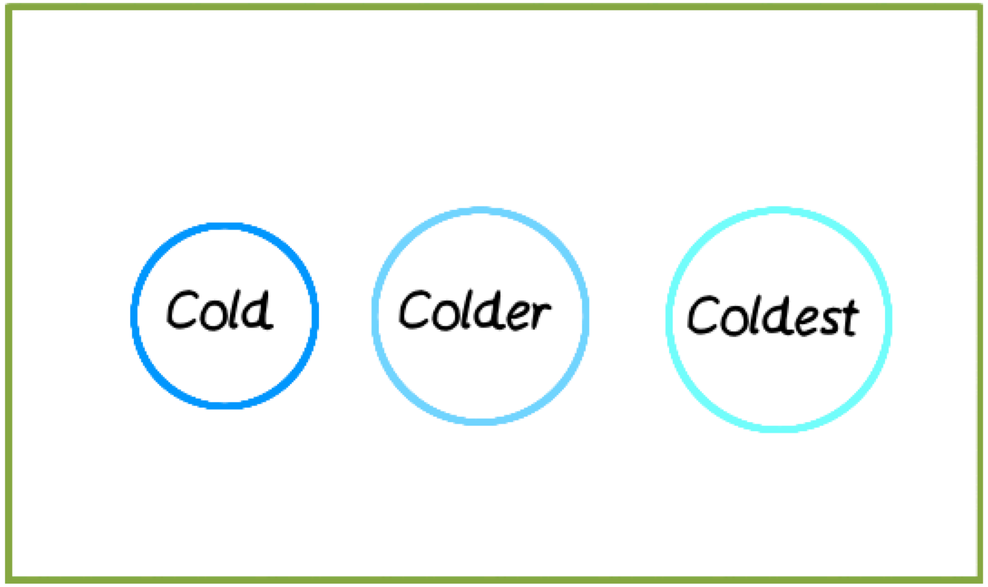
- Versão 1 - Construção do Projeto
- Versão 2 - Desenha icicle
- Versão 3 - Desenha Jogador
- Versão 4 - Controle de Teclado (setas)
- Versão 5 - Adiciona Icicles
- Versão 6 - Remove Icicles que somem da tela
- Versão 7 - Detecta Colisão
- Versão 8 - Adiciona o HUD
- Versão 9 - Adiciona níveis de dificuldade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

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

Como Adiciono Níveis de Dificuldade?

- Criamos a tela de seleção
- Adicionamos 3 botões (fácil, médio e difícil)
- Fazemos as ações de navegação entre telas





- [ ] Add DifficultyScreen
- [ ] Add difficulty select "buttons"
- [ ] Add navigation between screens

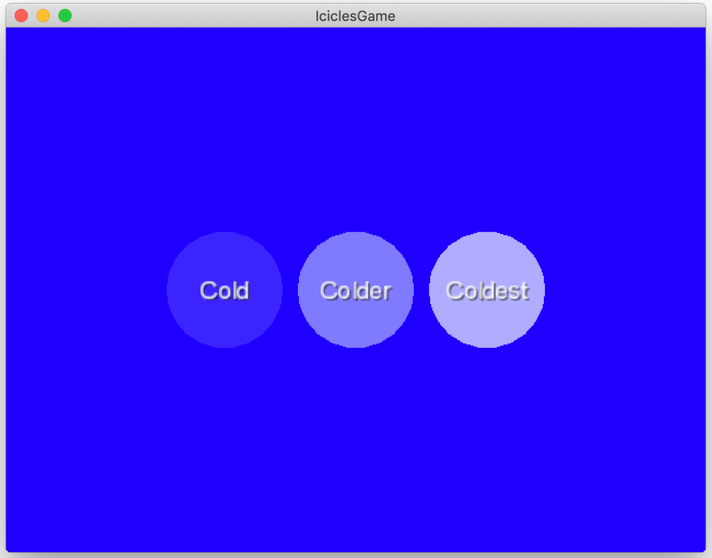
- Adicionar as diferentes cores dos níveis de dificuldade, tamanho do mundo difícil, raio dos botões, escala do label e posições dos 3 botões para escolher o nível.

Constants.java | DifficultyScreen.java | IciclesGame.java | IciclesScreen.java | Icicle.java

```
35      public static final float HARD_SPAWNS_PER_SECOND = 25;
36
37      // TODO: Add constants for the color of each difficulty select circle
38      public static final Color EASY_COLOR = new Color( r: 0.2f,  g: 0.2f,  b: 1,  a: 1);
39      public static final Color MEDIUM_COLOR = new Color( r: 0.5f,  g: 0.5f,  b: 1,  a: 1);
40      public static final Color HARD_COLOR = new Color( r: 0.7f,  g: 0.7f,  b: 1,  a: 1);
41
42      // TODO: Add constant for the size of the difficulty world
43      public static final float DIFFICULTY_WORLD_SIZE = 480.0f;
44
45      // TODO: Add constant for the radius of the difficulty select "buttons"
46      public static final float DIFFICULTY_BUBBLE_RADIUS = DIFFICULTY_WORLD_SIZE / 9;
47
48      // TODO: Add constant for the scale of the difficulty labels (1.5 works well)
49      public static final float DIFFICULTY_LABEL_SCALE = 1.5f;
50
51      // TODO: Add Vector2 constants for the centers of the difficulty select buttons
52      public static final Vector2 EASY_CENTER = new Vector2( x: DIFFICULTY_WORLD_SIZE / 4,
53              y: DIFFICULTY_WORLD_SIZE / 2);
54      public static final Vector2 MEDIUM_CENTER = new Vector2( x: DIFFICULTY_WORLD_SIZE / 2,
55              y: DIFFICULTY_WORLD_SIZE / 2);
56      public static final Vector2 HARD_CENTER = new Vector2( x: DIFFICULTY_WORLD_SIZE * 3 / 4,
57              y: DIFFICULTY_WORLD_SIZE / 2);
```

Classe DifficultyScreen.java

- Devemos criar a classe que representa a tela inicial para a escolha do nível de dificuldade.



DifficultyScreen.java | IciclesGame.java | IciclesScreen.java | Icicle.java

```
1   package com.udacity.gamedev.icicles;
2
3   import ...
17
18
19  public class DifficultyScreen extends InputAdapter implements Screen {
20
21      public static final String TAG = DifficultyScreen.class.getName();
22
23      IciclesGame game;
24
25      ShapeRenderer renderer;
26      SpriteBatch batch;
27      FitViewport viewport;
28
29      BitmapFont font;
30
31      public DifficultyScreen(IciclesGame game) { this.game = game; }
```

- Instanciamos o FitViewPort com o tamanho do mundo
- E definimos a escala de aumento da fonte do texto

DifficultyScreen.java | IciclesGame.java | IciclesScreen.java | Icicle.java | Icicles.java

```
35      @Override
36      public void show() {
37          renderer = new ShapeRenderer();
38          batch = new SpriteBatch();
39
40          // TODO: Initialize a FitViewport with the difficulty world size constant
41          viewport = new FitViewport(Constants.DIFFICULTY_WORLD_SIZE, Constants.DIFFICULTY_WORLD_SIZE);
42          Gdx.input.setInputProcessor(this);
43
44          font = new BitmapFont();
45          // TODO: Set the font scale using the constant we defined
46          font.getData().setScale(Constants.DIFFICULTY_LABEL_SCALE);
47          font.getRegion().getTexture().setFilter(TextureFilter.Linear, TextureFilter.Linear);
48      }
```

- Aplicar a FitViewPort instanciada (método apply())
- Definir a matriz de projeção a partir da camera
- Desenhar os botões com o ShapeRenderer

DifficultyScreen.java | IciclesGame.java | IciclesScreen.java | Icicle.java | Icicles.java

```
50      @Override
51      public void render(float delta) {
52          // TODO: Apply the viewport
53          viewport.apply();
54          Gdx.gl.glClearColor(Constants.BACKGROUND_COLOR.r, Constants.BACKGROUND_COLOR.g, Constants.BACKGROUND_COLOR.
55          Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);
56
57          // TODO: Set the ShapeRenderer's projection matrix
58          renderer.setProjectionMatrix(viewport.getCamera().combined);
59
60          // TODO: Use ShapeRenderer to draw the buttons
61          renderer.begin(ShapeType.Filled);
62          renderer.setColor(Constants.EASY_COLOR);
63          renderer.circle(Constants.EASY_CENTER.x, Constants.EASY_CENTER.y, Constants.DIFFICULTY_BUBBLE_RADIUS);
64          renderer.setColor(Constants.MEDIUM_COLOR);
65          renderer.circle(Constants.MEDIUM_CENTER.x, Constants.MEDIUM_CENTER.y, Constants.DIFFICULTY_BUBBLE_RADIUS);
66          renderer.setColor(Constants.HARD_COLOR);
67          renderer.circle(Constants.HARD_CENTER.x, Constants.HARD_CENTER.y, Constants.DIFFICULTY_BUBBLE_RADIUS);
68          renderer.end();
```

- Desenhar os labels (rótulos) dos botões

DifficultyScreen.java | IciclesGame.java | IciclesScreen.java | Icicle.java | Icicles.java

```
69
70          // TODO: Set the SpriteBatche's projection matrix
71          batch.setProjectionMatrix(viewport.getCamera().combined);
72          // TODO: Use SpriteBatch to draw the labels on the buttons
73          // HINT: Use GlyphLayout to get vertical centering
74          batch.begin();
75
76          final GlyphLayout easyLayout = new GlyphLayout(font, Constants.EASY_LABEL);
77          font.draw(batch, Constants.EASY_LABEL, Constants.EASY_CENTER.x,  y: Constants.EASY_CENTER.y + easyLayout.height / 2,
78
79          final GlyphLayout mediumLayout = new GlyphLayout(font, Constants.MEDIUM_LABEL);
80          font.draw(batch, Constants.MEDIUM_LABEL, Constants.MEDIUM_CENTER.x,  y: Constants.MEDIUM_CENTER.y + mediumLayout.hei
81
82          final GlyphLayout hardLayout = new GlyphLayout(font, Constants.HARD_LABEL);
83          font.draw(batch, Constants.HARD_LABEL, Constants.HARD_CENTER.x,  y: Constants.HARD_CENTER.y + hardLayout.height / 2,
84
85          batch.end();
86      }
```

Classe DifficultyScreen.java

- Atualizar o FitViewPort
- Quando o usuário usar o touchDown, parar de projetar e verificar se houve um clique no botão e mostrar a janela dos icicles

DifficultyScreen.java | IciclesGame.java | IciclesScreen.java | Icicle.java | Icicles.java

```
89  public void resize(int width, int height) {
90      // TODO: Update the viewport
91      viewport.update(width, height, centerCamera: true);
92  }
93
94  @Override
95  public void hide() {
96      batch.dispose();
97      font.dispose();
98      renderer.dispose();
99  }
100
101 @Override
102 public boolean touchDown(int screenX, int screenY, int pointer, int button) {
103     // TODO: Unproject the touch from the screen to the world
104     Vector2 worldTouch = viewport.unproject(new Vector2(screenX, screenY));
105
106     // TODO: Check if the touch was inside a button, and launch the icicles screen with
107     //  the appropriate difficulty
108     if (worldTouch.dst(Constants.EASY_CENTER) < Constants.DIFFICULTY_BUBBLE_RADIUS) {
109         game.showIciclesScreen(Difficulty.EASY);
110     }
111     if (worldTouch.dst(Constants.MEDIUM_CENTER) < Constants.DIFFICULTY_BUBBLE_RADIUS) {
112         game.showIciclesScreen(Difficulty.MEDIUM);
113     }
114     if (worldTouch.dst(Constants.HARD_CENTER) < Constants.DIFFICULTY_BUBBLE_RADIUS) {
115         game.showIciclesScreen(Difficulty.HARD);
116     }
117     return true;
```

Classe IciclesGame.java

- Definir quais objetos devem ser instanciados para cada janela

IciclesGame.java | IciclesScreen.java | Icicle.java | Icicles.java

```
1   package com.udacity.gamedev.icicles;
2
3   import ...
5
6
7   public class IciclesGame extends Game {
8
9       @Override
10      public void create() { showDifficultyScreen(); }
13
14
15      public void showDifficultyScreen() {
16          // TODO: Show the difficulty screen
17          setScreen(new DifficultyScreen( game: this));
18      }
19
20      public void showIciclesScreen(Difficulty difficulty) {
21          // TODO: Show the Icicles screen with the appropriate difficulty
22          setScreen(new IciclesScreen( game: this, difficulty));
23      }
24  }
```

- Adicionar um atributo IciclesGame em IciclesScreen
- E caso seja clicado, mostrar a janela de seleção de nível

IciclesScreen.java | Icicle.java | Icicles.java

```
33
34      // TODO: Add IciclesGame member variable
35      IciclesGame game;
36      Difficulty difficulty;
37
38      // TODO: Accept IciclesGame in the constructor
39      public IciclesScreen(IciclesGame game, Difficulty difficulty) {
40          // TODO: Save the IciclesGame
41          this.game = game;
42          this.difficulty = difficulty;
43      }
44
45      @Override
46      public boolean touchDown(int screenX, int screenY, int pointer, int button) {
47          // TODO: Tell IciclesGame to show the difficulty screen
48          game.showDifficultyScreen();
49          return true;
50      }
```