libGDX

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

Rafael Vieira Coelho

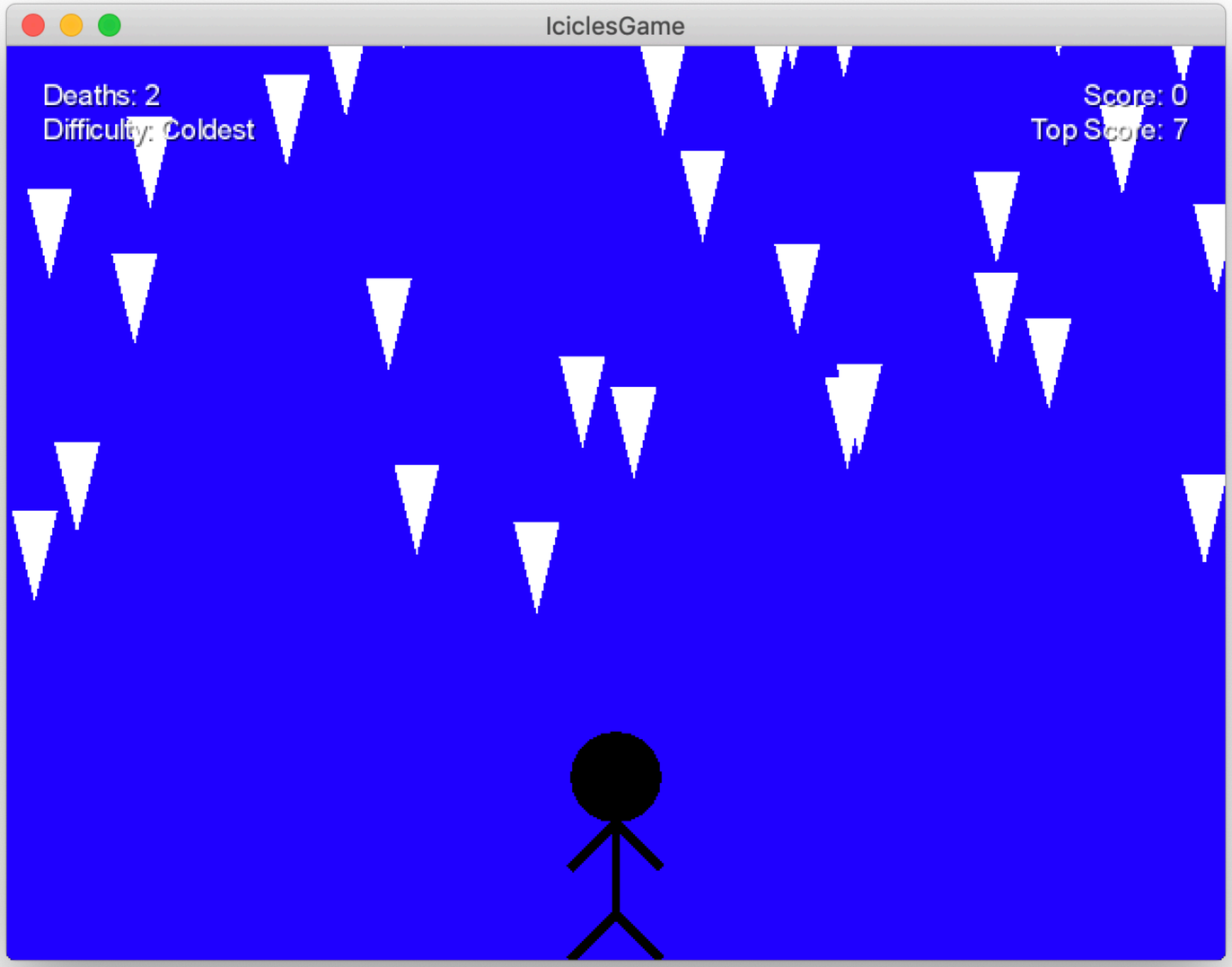
rafael.coelho@farroupilha.ifrs.edu.br

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

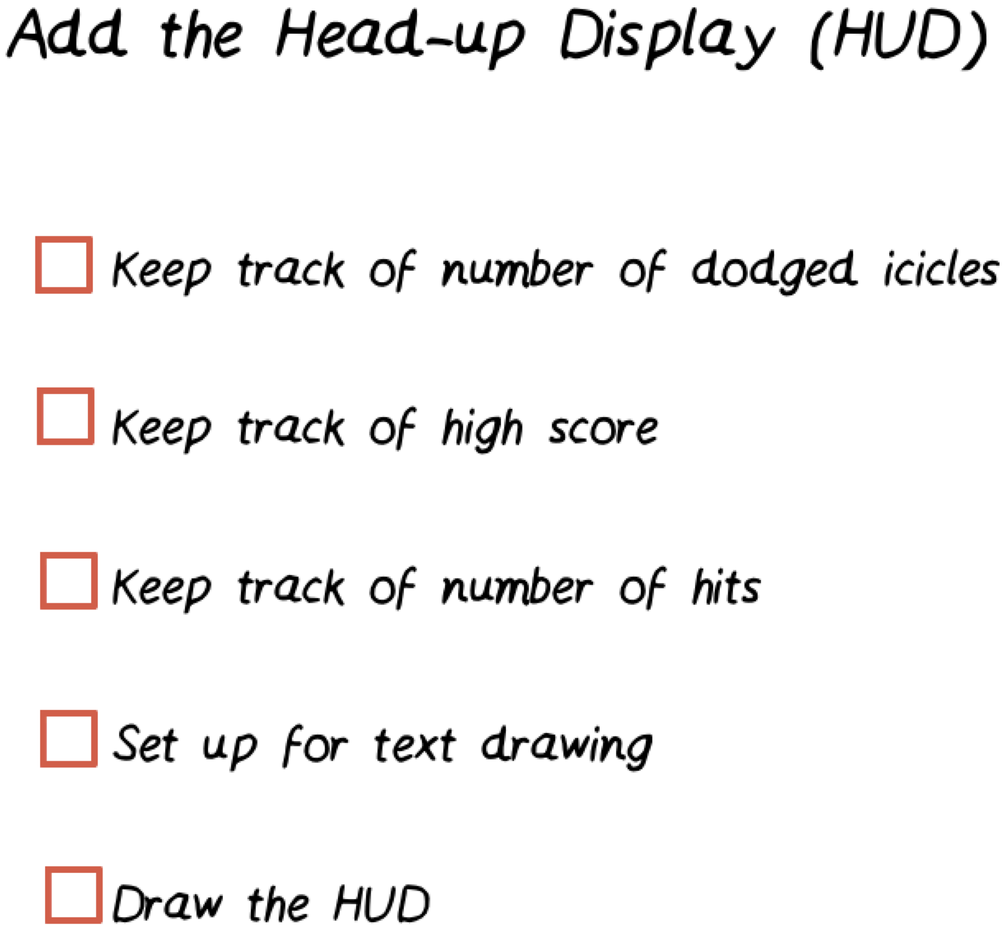
https://github.com/libgdx/libgdx/wiki/

Como Adiciono o HUD?



Como Adiciono o HUD?

- Manter um controle sobre o número de icicles desviados
- Manter um controle da maior pontuação
- Manter um controle do número de vezes que o jogador foi acertado por um icicle
- Configurar o texto para ser desenhado
- Desenhar o HUD (Head-up Display)



Classe Constants.java

- Devemos adicionar as constantes com o tamanho do HUD e a margem do mesmo.

Constants.java | IciclesScreen.java | Player.java | DifficultyScreen.java | Icicle.java

```
10
11      public static final float PLAYER_HEAD_RADIUS = 0.5f;
12      public static final float PLAYER_HEAD_HEIGHT = 4.0f * PLAYER_HEAD_RADIUS;
13      public static final float PLAYER_LIMB_WIDTH = 0.1f;
14      public static final int PLAYER_HEAD_SEGMENTS = 20;
15      public static final Color PLAYER_COLOR = Color.BLACK;
16      public static final float PLAYER_MOVEMENT_SPEED = 10.0f;
17
18      public static final float ACCELEROMETER_SENSITIVITY = 0.5f;
19      public static final float GRAVITATIONAL_ACCELERATION = 9.8f;
20
21      public static final float ICICLES_HEIGHT = 1.0f;
22      public static final float ICICLES_WIDTH = 0.5f;
23      public static final Vector2 ICICLES_ACCELERATION = new Vector2( x: 0,  y: -5.0f);
24      public static final Color ICICLE_COLOR = Color.WHITE;
25      public static final float ICICLE_SPAWNS_PER_SECOND = 10.0f;
26
27      // TODO: Add screen reference size for scaling the HUD (480 works well)
28      public static final float HUD_FONT_REFERENCE_SCREEN_SIZE = 480.0f;
29
30      // TODO: Add constant for the margin between the HUD and screen edge
31      public static final float HUD_MARGIN = 20.0f;
32  }
```

Classe Player.java

- Precisamos de um atributo que conte o número de mortes do jogador
- Inicializar este contador no construtor da classe
- E quando ocorre uma uma colisão com um icicle, somar 1 ao valor atual do contador

Player.java | Icicles.java | IciclesGame.java

```
18
19      // TODO: Add counter for number of deaths
20      int deaths;
21
22      public Player(Viewport viewport) {
23          this.viewport = viewport;
24          // TODO: Set number of deaths to zero
25          deaths = 0;
26          init();
27      }
28
29  @   public boolean hitByIcicle(Icicles icicles) {
30          boolean isHit = false;
31          for (Icicle icicle : icicles.icicleList) {
32              if (icicle.position.dst(position) < Constants.PLAYER_HEAD_RADIUS) {
33                  isHit = true;
34              }
35          }
36          // TODO: If the player was hit, increment death counter
37          if (isHit) {
38              deaths += 1;
39          }
40          return isHit;
41      }
```

Classe Icicles.java

- Precisamos ter um contador para o número de icicles desviados
- Inicializar o contador em zero
- E toda vez que um icicle passar do final da tela, adicionar um nesse contador

Icicles.java | IciclesGame.java

```
13
14      // TODO: Add counter for how many icicles have been dodged
15      int iciclesDodged;
16
17      public void init() {
18          icicleList = new DelayedRemovalArray<Icicle>( ordered: false,  capacity: 100);
19          // TODO: Set icicles dodged count to zero
20          iciclesDodged = 0;
21      }
22
23      public void update(float delta) {
24          if (MathUtils.random() < delta * Constants.ICICLE_SPAWNS_PER_SECOND) {
25              Vector2 newIciclePosition = new Vector2(
26                      x: MathUtils.random() * viewport.getWorldWidth(),
27                      viewport.getWorldHeight()
28              );
29              Icicle newIcicle = new Icicle(newIciclePosition);
30              icicleList.add(newIcicle);
31          }
32          for (Icicle icicle : icicleList) {
33              icicle.update(delta);
34          }
35          icicleList.begin();
36          for (int i = 0; i < icicleList.size; i++) {
37              if (icicleList.get(i).position.y < -Constants.ICICLES_HEIGHT) {
38                  // TODO: Increment count of icicles dodged
39                  iciclesDodged += 1;
40                  icicleList.removeIndex(i);
41              }
42          }
```