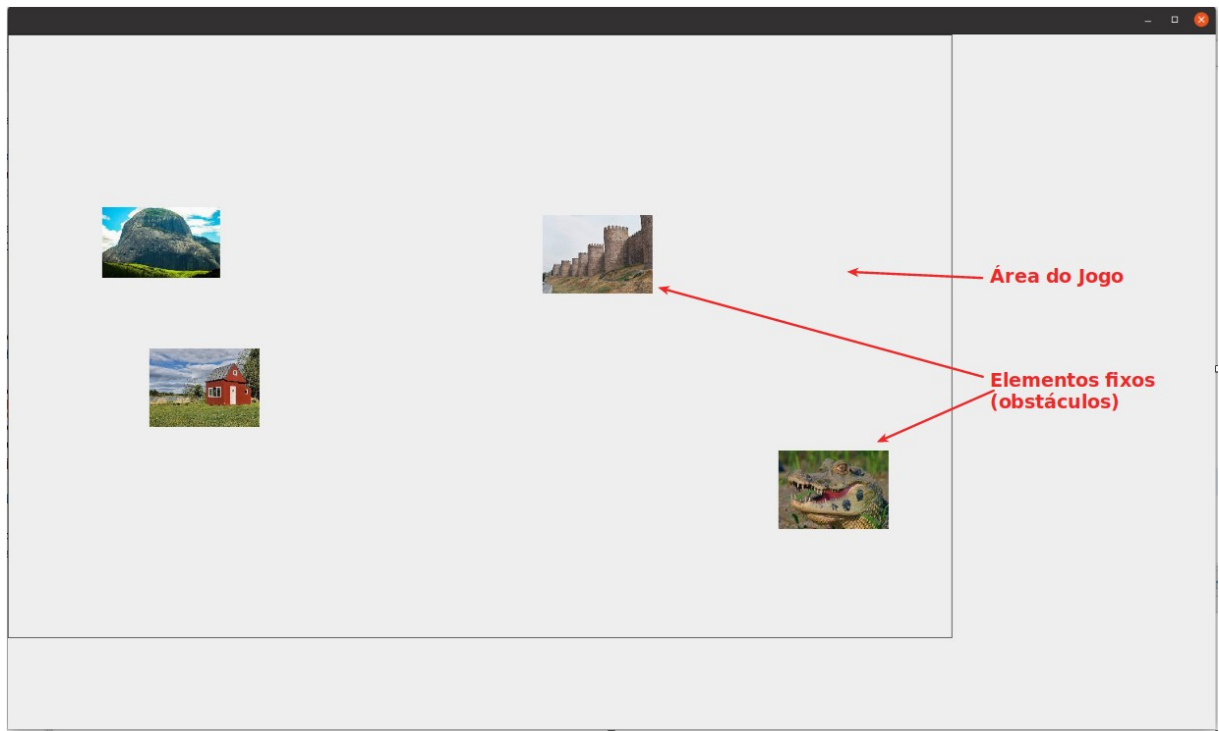


Tema: Criar um jogo simples com interface gráfica e usando os conceitos de POO. O jogo consistirá de uma área de jogo (1200x768) na qual serão dispostos diversos objetos, devendo estes ser de 3 grupos distintos (objetos fixos, objetos móveis que podem ser guerreiros ou projéteis).

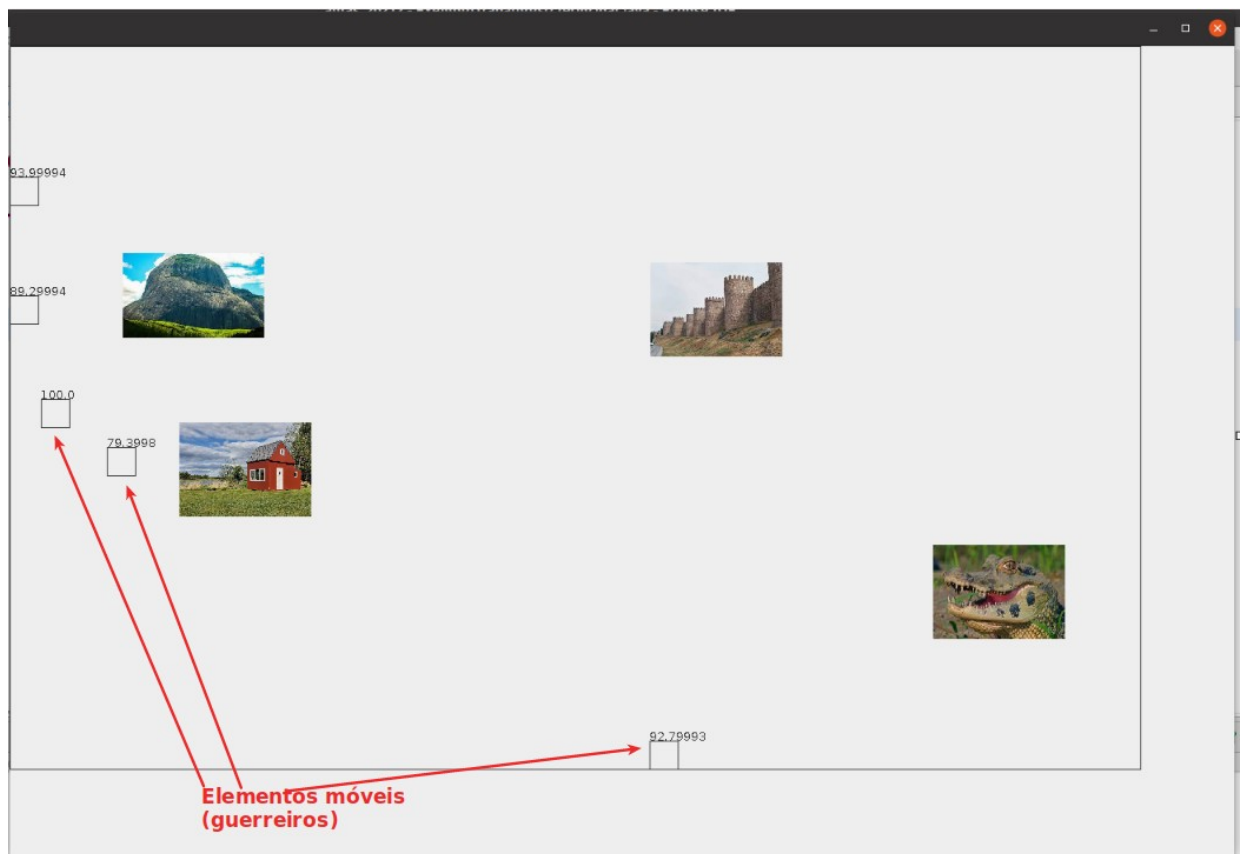
Etapa 1:

Desenhar a tela do jogo e os elementos fixos.



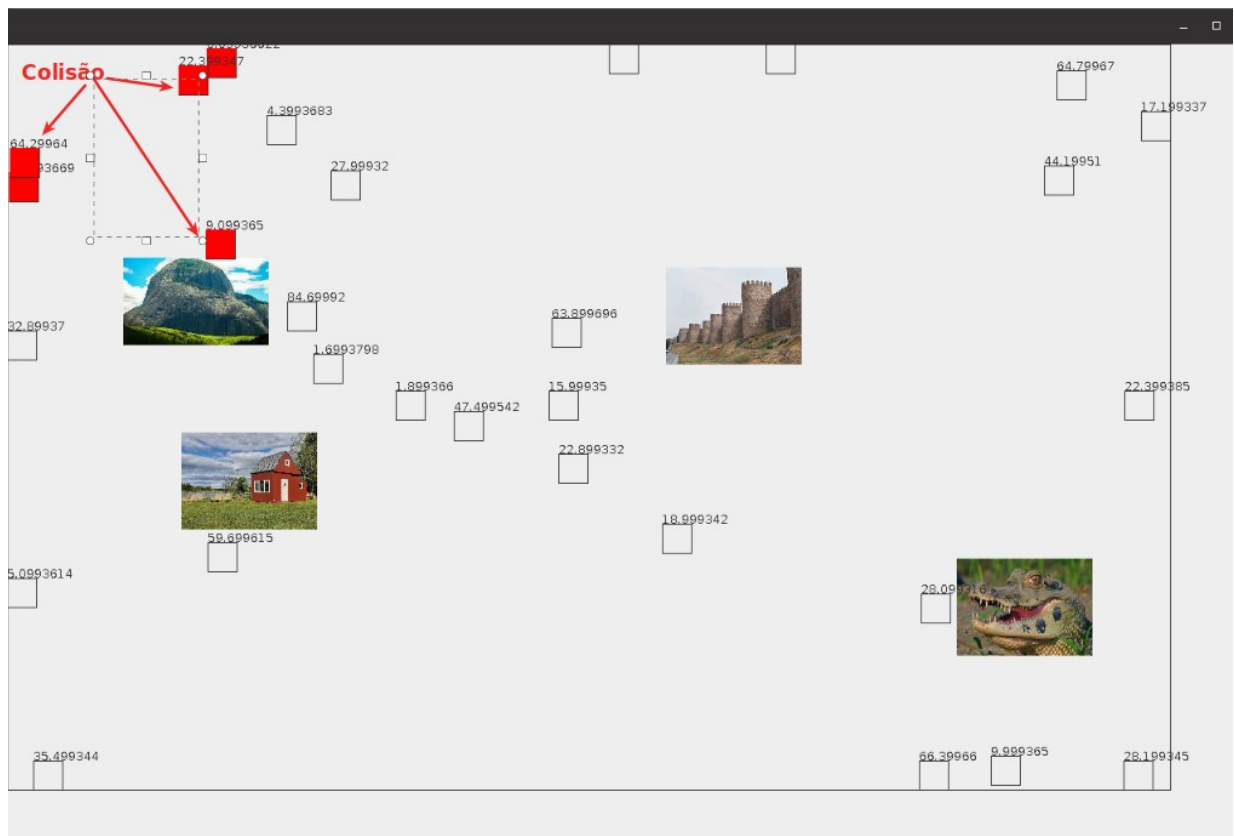
Etapa 2:

Adicionar os elementos que irão se movimentar na área do jogo.



Etapa 3:

Programar os movimentos e identificar colisão entre os próprios elementos móveis e entre os elementos móveis e os fixos.



Etapa 4:

Definir uma política de energia para cada elemento móvel e uma penalização para as colisões entre os objetos.

Veja o vídeo: [Aqui](#)

Etapa 5:

Disparo de projétil.

Veja o vídeo: [Aqui](#)

Dados Gerais:

Cada objeto móvel deve conter um indicador de energia que é reduzido em cada colisão. Na implementação de demonstração, utilizei valores para calcular a energia dos objetos após uma colisão: energia atual (energia), resistência do objeto (resistência) e capacidade de dano (ataque). Em caso de colisão entre objetos móveis, o cálculo usado foi (this é o objeto atual e outro é o objeto no qual ele colidiu):

```
this.energia = this.energia - (1 * outro.ataque) + (1 * this.resistencia);
```

Em caso de colisão com objetos fixos:

```
this.energia = this.energia - (1 + (1 * his.resistencia));
```

Os movimentos dos objetos não têm restrição, eles podem se movimentar em qualquer uma das quatro direções (UP, RIGHT, RIGHT e LEFT). Quando o objeto atinge uma borda do espaço de jogo, se ele é um guerreiro ele não sofre penalização e inicia aleatoriamente o movimento em outro sentido, se ele for um projétil ele é eliminado no jogo. Quando um guerreiro atinge um objeto fixo ele perde energia (conforme acima). Um projétil que atinge um objeto fixo é eliminado.

Funcionalidades:

A composição da nota final se dará da seguinte forma:

50% pelo funcionamento do programa e 50% pelo uso de orientação a objetos corretamente para resolver o problema. É requisito que apareçam nas classes definidas o uso dos conceitos de classe/objeto, encapsulamento, herança, interface e polimorfismo.

Importante: A implementação de referência aqui apresentada, é apenas um exemplo, você pode desenvolver uma interface gráfica com características diferentes desde que apresente as funcionalidades solicitadas e os conceitos de POO. Mais detalhes e explicações serão adicionados a este documento de acordo com as dúvidas que forem apresentadas.

Trabalho em equipe: 2 alunos por equipe.

Entrega: 29/11/23 – Com apresentação individual para o professor durante a aula.