

Classes:

A classe genérica de objeto deve ter:

- 4 métodos de movimentação: moveLeft, moveRight, moveUp, moveDown;
- 4 propriedades de aceleração: acelX, acelY, acelHoriz, acelVert;
 - acelHoriz e acelVert são valores fixos que são definidos na criação do objeto.
 - acelX e acelY são valores que variam de acordo com os valores de acelHoriz e acelVert. Exemplo:

```
function Entity:moveLeft()  
|   self.acelX = -self.acelHoriz  
end
```

- As imagens serão propriedade do objeto genérico, mas seu comportamento (renderização, atualização) será definido da classe herdeira;

A classe de player:

- Implementação de animação: A parte de input (usando isDown) além de chamar as funções de movimento (moveLeft, moveRight etc), também implementa a animação da imagem.
- O player possui como propriedade uma classe equivalente a sua arma. Haverá um botão no input que chama a propriedade de disparar a arma. A classe arma deve (se der tempo) “saber” quem a atirou para teste de colisão (friendly fire). Ao atirar deve-se dar um delay para poder atirar novamente (com temporizador).
- Os comandos do input deve ser parametrizado, por exemplo, no caso abaixo ao criar o player, definiu-se o left como a seta esquerda e right como a tecla D:

```
6  
7   self.left = "left"  
8   self.right = "D"  
9  
10  if love.keyboard.isDown(left) and not love.keyboard.isDown(right) then  
11  |   self.moveLeft()  
12  end
```

- O método acima garante que possa haver mais de um jogador e definir as teclas para tal.

A classe Gun:

- Implementação de animação;
- Movimento de zig-zag;
- Se der tempo, implementar explosão aumentando a largura e altura do tiro, de forma a simular uma explosão na colisão.