**Anotações Game Maker Studio 2 – GML 2024  
Por Victor Theodoro**

O **GameMaker Studio 2** é uma plataforma de **desenvolvimento de jogos** amplamente utilizada, conhecida por sua acessibilidade e poderosa capacidade de criar **jogos 2D**. Ele oferece uma interface visual para a criação de jogos, mas também permite o uso da Linguagem de Script do **GameMaker (GML)** para maior flexibilidade e controle.

-

O Game Maker é 100% baseado no **conceito de programação orientada a objetos**, tendo pronto em sua plataforma uma área com várias divisões de Objetos, Sprites, Áudios, Scripts, dentre outras coisas que serão necessárias em seu projeto.

Uma imagem contendo Diagrama

Descrição gerada automaticamente

**Room** basicamente é a **área principal do jogo**, onde utilizamos para **montar todo o design e unir os objetos** para que eles possam interagir um com o outro.  
 **Cada room é uma “sala” do jogo**, então por exemplo, podemos definir que na ‘ room\_menu’, terá um botão de “Iniciar”, quando clicado, levara o jogador para ‘room\_level01’.

**Objeto** é um componente que representa **elementos do jogo**, como personagens ou itens, e possui comportamento e interações definidos por código GML.

Dentro de um objeto, você **pode definir diferentes eventos para controlar seu comportamento**. Alguns dos principais eventos incluem:

**- Create:** Este evento **ocorre quando o objeto é criado no jogo**. É usado para inicializar variáveis, configurar propriedades e executar qualquer código necessário para a **preparação do objeto**.

**- Step:** O evento Step é acionado **a cada quadro do jogo**. É aqui que você coloca o código que precisa ser **executado continuamente**, como atualizações de movimento, verificação de colisões ou qualquer **lógica que precise ser verificada repetidamente**.

**- Draw:** Este evento é responsável por **desenhar o objeto na tela**. Aqui, você pode definir **como o objeto deve ser visualizado**, desenhando sprites, formas ou **outros elementos gráficos**.

**- Collision:** Quando o objeto **colide com outro objeto ou com uma determinada área do jogo**, este evento é acionado. É útil para lidar com interações de **colisão específicas, como reações a colisões com paredes ou com outros objetos**.

**- Destroy:** Este evento ocorre quando o **objeto é destruído no jogo**, seja por meio de código ou devido a certas condições no jogo. Pode ser usado para liberar recursos ou executar **ações finais antes que o objeto seja removido**.

-

**Funções:**

As funções são usadas para **executar ações específicas e realizar operações dentro do jogo**, permitindo que os desenvolvedores **controlem o comportamento e a lógica do jogo de maneira programática**.  
**Aqui estão algumas funções principais do GameMaker Studio 2:**

* **keyboard\_check(key):** Verifica se uma tecla do teclado está atualmente pressionada. Você fornece o código da tecla como argumento para esta função.
* **mouse\_check\_button(button):** Verifica se um botão do mouse está atualmente pressionado. O argumento button especifica qual botão do mouse será verificado.
* **instance\_create(x, y, obj):** Cria uma nova instância de um objeto na posição (x, y)  
  especificada.
* **instance\_destroy():** Destrói a instância atual do objeto em que a função é chamada.
* **draw\_sprite(sprite, subimage, x, y):** Desenha um sprite na posição (x, y) especificada na tela.
* **draw\_text(x, y, string):** Desenha um texto na posição (x, y) especificada na tela.
* **audio\_play\_sound(sound, priority, loop):** Reproduz um som especificado no jogo. Você pode especificar a prioridade e se o som deve ser reproduzido em loop.
* **room\_goto(room):** Muda para a Room especificada no argumento room.
* **collision\_rectangle(x1, y1, x2, y2, obj, prec, notme):** Verifica colisões entre instâncias de objetos dentro de um retângulo especificado pelas coordenadas (x1, y1) e (x2, y2).
* **random\_range(min, max):** Gera um número aleatório entre min e max.

-

**A verificação do teclado** é uma das principais funções e é feita através de **funções específicas que detectam se determinadas teclas estão sendo pressionadas** ou não. **Para fazer um personagem andar usando o teclado**, você pode seguir este exemplo básico em código GML:

**// Define as variáveis de movimento do personagem  
var moveSpeed = 4;  
var hSpeed = 0;**

**// Verifica as teclas pressionadas  
if (keyboard\_check(vk\_left)) {   
 hSpeed = -moveSpeed; // Define a velocidade horizontal para a esquerda  
}  
else if (keyboard\_check(vk\_right)) {  
 hSpeed = moveSpeed; // Define a velocidade horizontal para a direita  
}  
else {  
 hSpeed = 0;   
// Se nenhuma tecla de movimento horizontal estiver sendo pressionada  
a velocidade horizontal é 0  
}**

**// Aplica o movimento ao personagem no eixo X (horizontal)  
x += hSpeed;**

* **O GameMaker Studio 2 fornece uma lista de códigos como esse para várias teclas comuns, como setas direcionais, teclas de função e teclas alfanuméricas.**

**https://manual.gamemaker.io/monthly/br/GameMaker\_Language/GML\_Reference/Game\_Input/Keyboard\_Input/Keyboard\_Input.htm**

-