

# Construção do Jogo

Programação de Jogos

Gravity Guy

Judson Santos Santiago

# Introdução

- ▶ O **Gravity Guy** é jogo infinito de corrida desenvolvido e publicado pela **MiniClip** em 2010 para dispositivos móveis



O jogador controla o Gravity Guy pelo pressionamento da tela (ou da barra de espaços na versão para PC) para trocar a gravidade. O objetivo é correr o mais longe possível enquanto evita os obstáculos que podem prender o personagem no cenário e fazer ele ser pego pelo oficial de polícia.

# Recursos do Jogo

## Plataformas Gravity Guy

Pequena



Média



Grande



Enter  
Enter  
Enter  
Enter  
Enter



## Camadas do Fundo



# Mecânicas do Jogo

- ▶ O personagem é mantido no **centro da tela**
  - Ele **cai com velocidade constante** para simular a gravidade
  - Ao colidir com uma plataforma sua posição é ajustada
    - Fica em constante colisão com as plataformas
  - O **pano de fundo** passa a sensação de movimento
    - Ele deve se mover no eixo x **com a velocidade do personagem**



# Mecânicas do Jogo

- ▶ A **inversão da gravidade** pode ser obtida invertendo a direção de movimento no eixo y

```
if (window->KeyPress(VK_SPACE))
{
    gravity = !gravity;
}

...

if (gravity == NORMAL)
    Translate(0, 300 * gameTime);
else
    Translate(0, -300 * gameTime);
```



As camadas do fundo usam velocidades ligeiramente diferentes para fornecer a **sensação de profundidade**.

# Mudança de Fases

- ▶ A Engine **executa** os métodos Init, Update, Draw e Finalize de um **objeto da classe Game**
  - Ela fornece o método Next para trocar de fase
  - A troca causa a liberação de todos os objetos da fase, **incluindo o jogador.**

```
if (window->KeyPress('N')) {  
    // passa para o nível 2  
    Engine::Next<Level2>();  
}
```

```
template<class T>  
static void Next()  
{  
    if (game)  
    {  
        game->Finalize();  
        delete game;  
        game = new T();  
        game->Init();  
    }  
};
```



# Mudança de Fases

- ▶ E se quisermos **manter o estado corrente do jogador** entre as várias fases do jogo?

- Podemos trazer a mudança de fases e o jogador para a classe principal
- É preciso **remover o jogador da cena** antes de mudar de fase para evitar que ele seja deletado

```
if (window->KeyPress('N')) {  
    // passa para o nível 2  
    scene->Remove(GravityGuy::player, MOVING);  
    GravityGuy::NextLevel<Level2>();  
}
```

```
template<class T>  
static void NextLevel()  
{  
    if (level)  
    {  
        level->Finalize();  
        delete level;  
        level = new T();  
        level->Init();  
    }  
};
```

# Resumo

- ▶ A Engine **executa um jogo** através dos métodos:
  - Game::Init()
  - Game::Update()
  - Game::Draw()
  - Game::Finalize()
- ▶ Ela executa qualquer objeto que seja **subclasse de Game**
  - Podemos **mudar de nível** criando uma nova instância de Game e alterando o “game” executado pelo motor
  - Ou podemos fazer nosso game executar outros, controlando assim a mudança de fases e **mantendo o jogador vivo**