## INF 01047 - Fundamentos de Computação Gráfica

## (2016/01)

*Prof. Marcelo Walter*

***Tiago Silveira Camilo  
201249***

**Objetivo:**

Desenvolver um jogo com a mecânica semelhante ao jogo Dig Dug II. Utilizando tecnologia OpenGL alinhado aos conhecimentos adquiridos no decorrer do semestre.

**Descrição Geral:**

O jogo possui 3 tipos de "visão": 2D, 1ª Pessoa e 3ª Pessoa.

Controles:

* W - Movimenta para frente.
* S - Movimenta para trás.
* A - Movimenta para Esquerda [2D e 3ª Pessoa] | Gira câmera 90º Esquerda [1ª Pessoa].
* D - Movimenta para Direita [2D e 3ª Pessoa] | Gira câmera 90º Direita [1ª Pessoa].
* V - Alterna modo de jogo entre 2D, 1ª Pessoa e 3ª Pessoa.
* Space - Cria rachadura, somente se o personagem estiver em cima de um buraco.

O personagem está em um mapa finito. Caso atravesse o limite do mapa, será caraterizada "derrota".

O mapa é composto por duas imagens BMP 20x20:

* A primeira caracteriza o mapa inferior, neste caso define onde será "chão da nave" e onde será "espaço".
* A segunda caracteriza o mapa superior, neste caso define a posição dos elementos listados abaixo.

No mapa existem os elementos:

* Esfera
  + Elemento 3D
  + Elemento de posição fixa, em caso de colisão o personagem não avança.
* Inimigo
  + Elemento 3D
  + Elemento de posição variável, em caso de colisão caracteriza "derrota".
  + O inimigo possui inteligência para detectar que está a menos de 4 casas de distância do personagem e, nesse caso, iniciar uma perseguição.
* Buraco
  + Textura modificada
  + O personagem pode andar sobre, os inimigos não.
  + Com o personagem em cima de um buraco será possível criar uma rachadura na direção para qual a visão do personagem aponta.
* Rachadura
  + Textura modificada
  + O personagem pode andar sobre, os inimigos não.
  + A rachadura será criada até o próximo buraco, ou o fim do mapa, caso não exista buraco em seu caminho.
  + Quando as rachaduras fecharem um "polígono", o menor polígono será "desmoronado".
    - Neste caso foi utilizado o algoritmo recursivo "FloodFill" para contar a quantidade de casas em cada polígono, identificando assim o menor.

Todo o projeto foi desenvolvido utilizando versionamento GIT, através do repositórios abaixo é possível acompanhar os commits e evolução do projeto.

<https://bitbucket.org/camilo_tiago/computacaografica/commits/all>