

Relatório de Computação Gráfica

Trabalho 2

Anderson Caio Santos Silva, 7972630

Luana Okino Sawada, 4589724

Robson Marques Pessoa, 8632563

Enunciado do trabalho: Faça uma aplicação interativa com a API OpenGL que represente um game do estilo “Space Invaders (Tomohiro Nishikado - 1978)”. Obs. 1: A aplicação deve tornar possível o controle de uma “nave” pelo usuário. De forma que sua movimentação seja ajustada de acordo com as setas do teclado (ou sistema AWSO). Além disso, a nave deve disparar um tiro quando o usuário pressionar determinada tecla (barra de espaço, por exemplo). Obs. 2: A aplicação também deve representar uma matriz de naves invasoras. Esta matriz deve ter, pelo menos, 5 linhas e 5 colunas. De forma que deve haver pelo menos três estilos (*designs*) diferentes de nave. Obs. 3: As naves invasoras também devem atirar eventualmente. Obs. 4: A matriz de naves invasoras deve se aproximar da base defensora eventualmente. Obs. 5: O jogo termina quando uma nave invasora tocar a base defensora, quando a nave defensora destruir todas as naves inimigas ou quando a nave defensora for atingida mais de três vezes por disparos inimigos.

Plataforma utilizada: O referido trabalho foi confeccionado no sistema operacional Linux. Não houve a utilização de IDE. Apenas foi utilizado um editor de texto e o compilador gcc pelo terminal, o qual faz uso da linguagem C e a biblioteca OpenGL.

Tecnologias externas: Para a confecção do referido trabalho foram utilizados os materiais de estudos obtidos no site da disciplina.

Principais dificuldades: O passo mais desafiador deste trabalho foi a tecnologia que envolvia a questão de detecção de colisão.

Divisão de tarefas: As tarefas deste trabalho foram mais dependentes uma das outras. Assim neste trabalho desenvolvemos tudo junto, um ajudando o outro, já que achamos mais fácil trabalhar assim do que paralelizar o trabalho.

Tempo: Para a realização deste trabalho foi preciso trabalhar por 16 horas.

Trecho de código que o grupo julga mais importante: O grupo acredita que o código a seguir foi a peça principal no referido trabalho.

```
280 // collision
281 for(int i = 0; i < 5; i++) {
282     for(int j = 0; j < 10; j++) {
283
284         // if the bullet is active, we have to move it as well
285         if(invaders[i][j].bullet.active) {
286             invaders[i][j].bullet.y -= (step+1)/0.5;
287
288             // Check if the bullet has decreased player's life
289             GLfloat bx = invaders[i][j].bullet.x;
290             GLfloat by = invaders[i][j].bullet.y;
291
292             if(bx >= base.x - 50.0f && bx <= base.x + 50.0f &&
293                by >= base.y - 35.0f && by <= base.y + 20.0f) {
294                 invaders[i][j].bullet.active = false;
295                 base.life_count--;
296             }
297
298             // has reached the bottom
299             if(invaders[i][j].bullet.y <= base.y-35)
300                 invaders[i][j].bullet.active = false;
301         }
302     }
303 }
```

Figura1: Trecho de código que o grupo julga mais importante

Demonstração de funcionalidades: Neste tópico são apresentados prints da cena gerada como resultado do trabalho, assim como um manual de uso, caso o aplicativo permita manipulação com o usuário.

Na figuras abaixo é ilustrado que a nave possa ser controlada com as setas do teclado. Além disso, a nave disparar um tiro quando o usuário pressionar a barra de espaço. Também representamos uma matriz, com 5 linhas e 5 colunas, de naves invasoras com diferentes estilos de nave. As naves invasoras também atiram com uma função randômica.

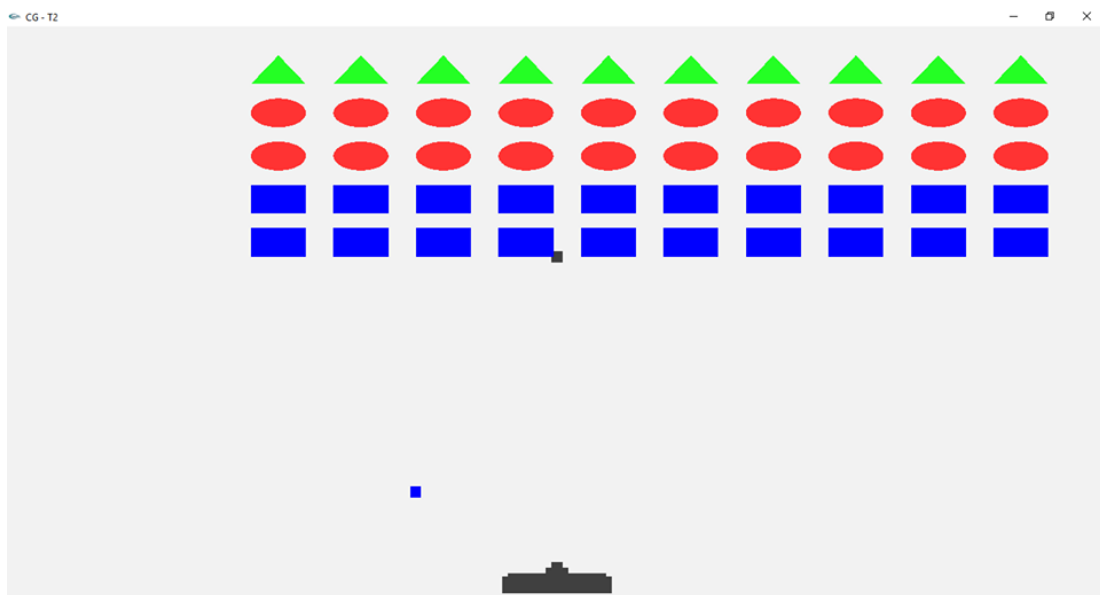


Figura 2 movimentação da nave com as setas do teclado e espaço para atirar

A nave possui 3 vidas. Quando um tiro de uma nave invasora acerta a nave, a mesma muda de cor para indicar que foi atingida e que perdeu uma vida e assim sucessivamente até perder as 3 vidas e o programa é fechado. Já as naves invasoras possuem diferentes estilos e cada estilo possui diferentes quantidades de vida. As naves quadradas possuem uma vida, já as em forma de círculos possuem 2 vidas e a triangular possui 3 vidas. A vida das naves invasoras também quando perdidas fazem as mesmas mudarem de cor.

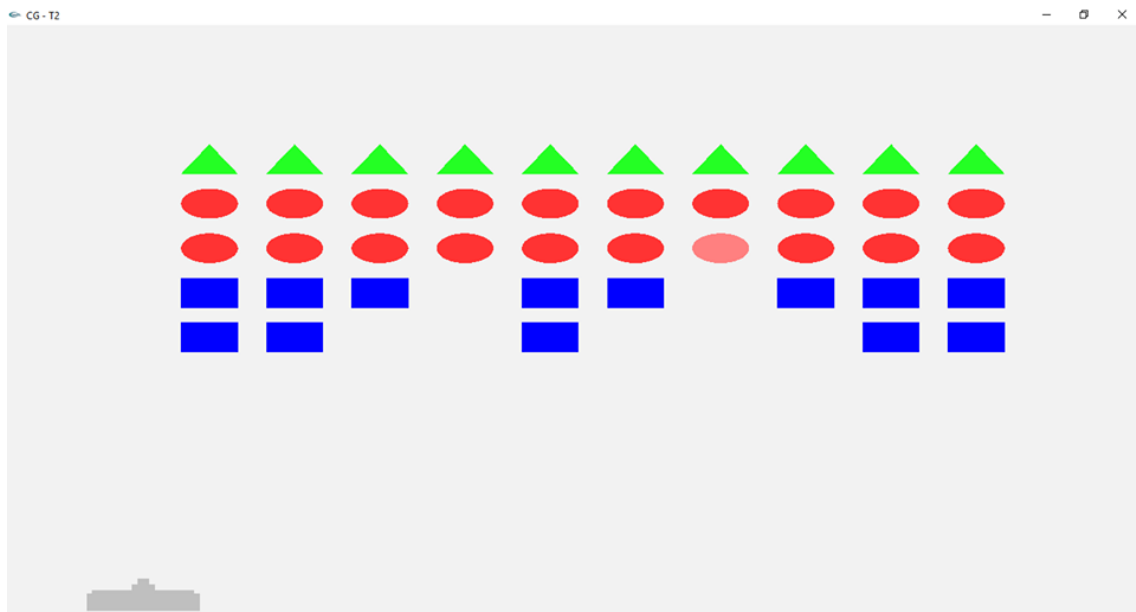


Figura 3 sistema de perda de vida

O jogo termina quando o jogador morre, uma nave invasora tocar a base defensora e quando todas as naves foram destruídas.