



Universidade Federal de Roraima
Departamento de Computação
Disciplina: DCC 703 - Computação Gráfica
Alunos: Joshua Kook Ho Pereira & Rodrigo de Andrade
Rolim Bem
Matrícula: 2017009284 & 2017009480

Relatório - Projeto Final

1. Introdução

Para o trabalho final da disciplina, foi desenvolvido uma variante do jogo Tetris utilizando a engine Unity v.2019.4.16f1. O jogo possui as funcionalidades clássicas, como mover e rotacionar peças, bem como funcionalidades de prever a próxima peça e outra para segurar/trocar a peça atual.

2. Roteiro

O jogo é semelhante à sua contraparte original: o jogador irá encaixar peças de formatos diferentes de modo a completar uma linha. Linhas completas irão desaparecer e aumentar a pontuação do jogador. O jogo irá continuar até que ele se encontre em uma situação onde não é mais possível encaixar nenhuma peça, condição que marca a derrota do jogador e o fim do jogo. Juntamente ao aviso de fim de jogo está disponível a opção de jogar novamente.

3. Elementos do Jogo

Os elementos principais do jogo são as peças e o espaço onde elas estarão situadas. Foram criados 7 *prefabs*, uma para cada peça, chamadas tetrominos, de formatos e cores diferentes:



Figura 1 - Tetrominos

O script dos tetrominos são encarregados de detectar a entrada do teclado do jogador e realizar o movimento correspondente, sempre checando se a ação é válida, seja de movimentação ou de rotação. Ao detectar que não é mais possível descer, o tetromino assume que ele chegou no fundo ou que há uma peça embaixo dele e que o jogador não poderá mais mexer nela. Neste momento o tetromino irá requisitar que seja gerada a próxima peça e que seja prevista a seguinte. Caso o tetromino se encontre em uma posição inválida no momento em que ele é gerado, é gerada a condição de fim de jogo. Os tetrominos foram montados a partir de 7 quadrados com bordas de cores diferentes, que foram preenchidos posteriormente.

O *Spawner* é o *GameObject* encarregado de gerar as peças no topo do espaço de jogo. Ele possui um vetor de objetos com 7 objetos, que são os tetrominos. O script possui 3 funções:

- **spawnNext:** irá gerar o próximo tetromino, definido pelo *GameObject Previewer*.
- **spawnFirst:** chamado apenas 1 vez no início da cena do jogo para gerar o primeiro tetromino.
- **switchT:** irá gerar o tetromino que estava sendo segurado.

O *Previewer* é um objeto que irá, aleatoriamente, decidir qual será o próximo tetromino a ser gerado, e irá mostrá-lo ao jogador. Ele também é responsável por atualizar a pontuação, a qual é baseada no número de linhas deletadas, e também por exibir a mensagem de fim de jogo, juntamente com o botão de reiniciar o jogo.

O script *Restart* do botão de reiniciar irá apenas recarregar a cena do jogo, reiniciando todas as variáveis.

O script *Playfield* é o responsável por gerenciar o espaço de jogo. Ele possui as funções de:

- Detectar linhas completas;
- Deletar linhas completas;
- Reposicionar as linhas que estavam acima da linha deletada para uma linha abaixo;
- Armazenar o número de linhas deletadas, a pontuação e o tetromino atual;
- Variar a velocidade com que os tetrominos caem.

As funções relativas ao reposicionamento e remoção de linhas faz uso de uma matriz interna para auxiliar o processo.

Por fim, o *GameObject Holder* é o objeto responsável por gerenciar a funcionalidade de segurar um tetromino, ou trocar o atual com o que se estava segurando. Para tal ele dispõe de duas variáveis, uma para verificar se ele já está segurando algo e outra para verificar qual o tetromino que está sendo segurado. Toda vez que um tetromino for segurado, o *Holder* irá requisitar que o *Spawner* gere o tetromino que ele estava segurando anteriormente, se houver algum, e irá armazenar o tetromino que estava na tela, efetivamente trocando o tetromino atual com o que estava guardado.

A cena do jogo foi feita com um fundo cinza com lacunas em preto, nas quais ficam posicionadas o campo de jogo, a peça segurada e a previsão da próxima peça. Foram posicionados dois *sprites* de linha reta nas laterais do campo de jogo,

os quais foram estendidos verticalmente para cobrir toda a tela, simbolizando os limites do espaço de jogo.

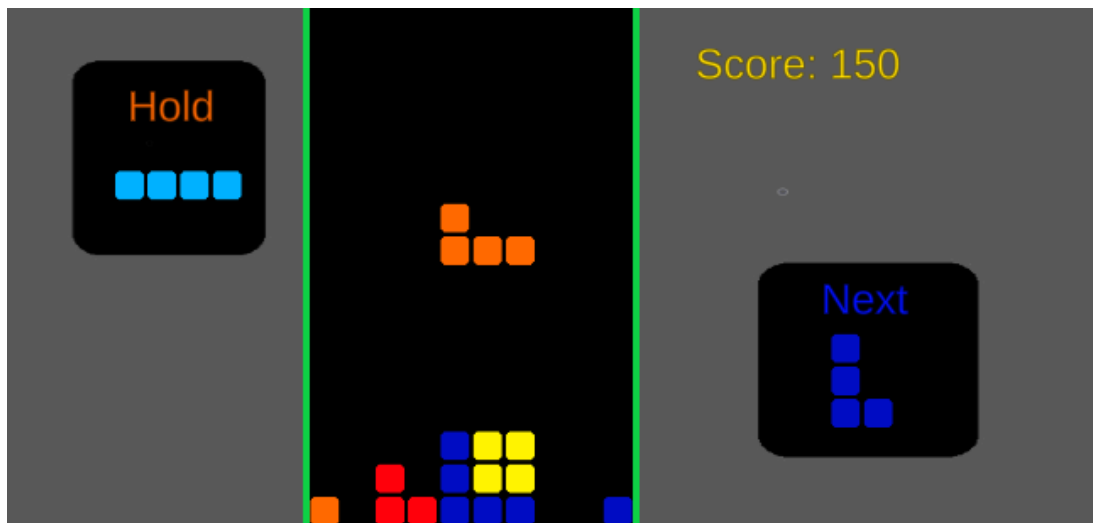


Figura 2 - Cena do Jogo

Os textos foram feitos utilizando o canvas disposto pelo próprio Unity.

Por fim, foi adicionado uma música de fundo, escolhida a gosto do desenvolvedor.

4. Ferramentas

O jogo e suas funcionalidades foram desenvolvidos usando a engine Unity v.2019.4.16f1. Os sprites para os planos de fundo, os blocos, o botão, os tetrominos e as bordas foram feitos utilizando o Paint3D, devido à sua funcionalidade de manter um fundo transparente.

5. Principais Técnicas de Computação Gráfica

As técnicas mais utilizadas foram as de transformações espaciais, em especial a translação e a rotação.

A Translação fica bastante explícita na função de movimentação do tetromino. O Unity aplica essa técnica facilmente quando se soma ao vetor de posição do tetromino um novo vetor com o deslocamento nos eixos X e Y.

```
transform.position += new Vector3(1, 0, 0);
```

Figura 3 - Aplicação da Translação

Já a Rotação é realizada quando aplicamos a função interna *Rotate* de um objeto do tipo *transform*, especificando um vetor com os valores de rotação em graus em cada eixo.

```
transform.Rotate(0, 0, -90);
```

Figura 4 - Aplicação da Rotação

Outra técnica utilizada também é a de escala, utilizada nas bordas. Inicialmente, as bordas eram apenas uma linha reta, mas através do Unity é possível aumentar a escala dos seus eixos. Dessa forma a escala vertical dos sprites das bordas foi multiplicada por 40 e a horizontal por 3.

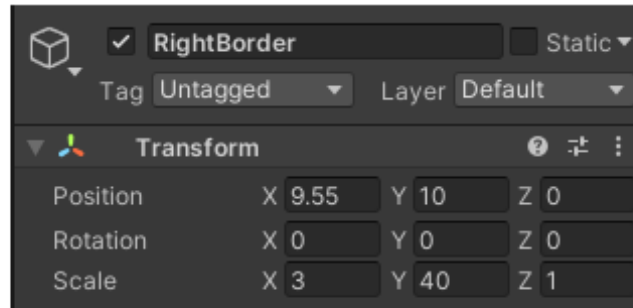


Figura 5 - Configuração de uma das bordas

Outras técnicas utilizadas foram as de Rasterização de Linhas e de Preenchimento, as quais o software Paint3D aplica automaticamente.

6. Conclusões

Antes de realizar este trabalho, bem como no decorrer na disciplina de Computação Gráfica, tomei um ponto de vista diferente de antigamente. Apesar de ter experiência não só como Unity, mas também com diversos outros softwares que utilizam técnicas de C.G., nunca parei para me perguntar como esses softwares funcionavam internamente. Após aprender sobre o assunto, enxergo melhor como as coisas funcionam por trás dos panos, e como eles, apesar de ter apenas um comando simples, fazem todo um processo matemático para executar o que queremos. Foi realmente uma experiência nova e interessante.(Joshua)

Todo o aprendizado no decorrer da disciplina, foi de grande importância para entender o funcionamento das ferramentas e técnicas empregadas nos softwares e jogos, assim como esclarecer o mundo da computação gráfica, que faz coisas incríveis, porém o background é cheio de cálculos e lógica matemática, coisas que não eram tão claras em minha mente.(Rodrigo)