

## Trabalho de Computação Gráfica – Fase Final

---

Ângela Rodrigues Ferreira RA: 552070

Charles David de Moraes RA: 489662

22 de Julho de 2016

### PROPOSTA GERAL DO TRABALHO

O objetivo deste trabalho é desenvolver um pequeno jogo utilizando WebGL explorando as transformações vistas em sala de aula.

O jogo implementado segue um estilo de *Doodle Jump* com elementos em 3D, ou seja, uma proposta de jogo 2.5D onde tem-se elementos em 3D no entanto, a jogabilidade é restringida para o plano 2D, onde o jogador não interage no eixo z.

Sendo essa a última fase do trabalho, o objetivo é ampliar a implementação apresentada nas fases anteriores para a finalização de todos os requisitos implementados com correções e melhorias, incluir textura em pelo menos um objeto e ter interações entre objetos com detecção de colisões.

### O JOGO

O jogo foi desenvolvido para Web utilizando a biblioteca `Three.js` e para executá-lo basta abrir a página `index.html` em um navegador de sua preferência.

Logo ao executar o jogo, o usuário encontra as instruções rápidas de manipulação da personagem e um botão "*Play*" para começar a jogar. Na parte superior direita da tela, há um menu com a opção de voltar para o início do jogo ou ir para uma tela de informações onde dados sobre os desenvolvedores e também sobre o jogo podem ser encontrados.

Esse jogo tem como personagem principal uma esfera branca que busca conseguir pontos através de moedas e/ou corações recolhidos ao decorrer do jogo evitando tocar em seu inimigo monstro.

O teclado é utilizado para controlar os movimentos da personagem principal, sendo a tecla "A" utilizada para movimentá-la para esquerda, "D" para movimentá-la para direita e "Espaço" para pular.

Quando a esfera entra em contato com alguma moeda ou algum coração em cena, ela ganha pontos. Porém, se ela entrar em contato com o inimigo, o jogo se encerra.

As animações desenvolvidas no jogo são o movimento de rotação da moeda e do coração, escalas dos objetos em cena, movimentação do inimigo e, a partir de interação do usuário, a movimentação da personagem principal (pular e andar para os lados).

## PRINCIPAIS DIFICULDADES

Nessa fase, uma das dificuldades encontradas foi a execução do "pulo" da personagem principal, primeiramente tentamos utilizando a biblioteca Tween.js e uma função seno, no entanto haviam algumas limitações em relação ao ponto de partida do pulo e futuramente na detecção de colisão, logo procuramos outra alternativa e conseguimos implementar o pulo do personagem utilizando a biblioteca física Physi.js ao adicionar gravidade à cena e transformar o personagem em um objeto físico com massa e aplicando um impulso nele no eixo Y, deixando com que a gravidade cuide de sua queda naturalmente.

Juntamente com a dificuldade supracitada, encontramos mais um problema: detectar colisões entre o personagem e os outros objetos. Uma primeira abordagem que tentamos foi utilizando Raycaster no entanto não obtivemos sucesso para detectar as intersecções dos objetos e consequentemente não conseguimos detectar as colisões entre o personagem e os outros objetos através dessa abordagem. Nossa segunda abordagem foi novamente utilizando a biblioteca Physi.js, no entanto com algumas ressalvas. Não conseguimos fazer com que a personagem consiga pular por baixo da plataforma, ele deve subir nas plataformas pelas laterais.

Apesar de conseguirmos detectar colisão do personagem com todos os outros objetos, não conseguimos manter os movimentos originais dos objetos, portanto optamos por deixar apenas o coração como um objeto físico que o objeto do personagem consiga detectar colisão. Dessa forma deixamos o jogo sem um mecanismo para se encerrar quando colidir com o inimigo, visto que quisermos manter sua movimentação da fase passada e também para não infringir um dos requisitos das fases anteriores.

## DIVISÃO DE TAREFAS

Ambos desenvolvedores testaram soluções para os problemas a serem resolvidos e por fim, cada um participou em todas as áreas.