

Exercícios – Física para Jogos Digitais

Começando a Explorar a Box2D – Corpos Rígidos

Vamos começar relembando os conceitos que conversamos na Aula 2:

1. O que é um corpo rígido?

R: Um corpo que não pode ser deformado, somente transladado e rotacionado.

2. Pensando computacionalmente na simulação de um corpo rígido, quais são os atributos mais importantes para descrevê-lo? Você pode se basear na classe `b2Body` da Box2D.

R: Atributos e Fixtures.

3. Qual a diferença entre um objeto de corpo rígido declarado como cinemático (na Box2D, `b2_kinematicBody`) e dinâmico (`b2_dynamicBody`)?

R: O corpo cinemático não sofre alterações em seus atributos de posição e rotação por ação externa, ele deve ser movido manualmente em seus atributos (como um controlador de personagem por exemplo), já o dinâmico possui massa e portanto pode sofrer alteração por interações com outros objetos físicos. Por exemplo, digamos que a proposta é um personagem de FPS, como CS, em que só o jogador se move, ações de outros jogadores ou cenário não o afetam, mas quando ele morre, seu corpo pode ser movido por granadas e ações de outros jogadores. Então poderíamos utilizar para o personagem jogável um `kinematicBody` e quando ele chegar a zero pontos de vida, substituímos seu modelo pelo `Ragdoll`, um objeto `dynamicBody`.

4. O que é centro de massa?

R: É o ponto de equilíbrio do objeto, ou seja, um ponto em que se aplicada força, todo o objeto “sente” por igual. Em objetos homogêneos ele fica no centro geométrico, mas em objetos com formas desiguais ou diversos tipos de materiais em sua composição, ele irá se encontrar em um ponto diferente, no corpo humano ele fica próximo ao umbigo, em um martelo, fica mais próximo do centro da cabeça de metal do que da ponta do cabo de madeira.

5. Considere o seguinte cenário: você está simulando um corpo rígido em ambiente 2D que possui a forma de círculo. Como você mandaria desenhá-lo, utilizando OpenGL ou outra biblioteca gráfica? Descreva pelo menos uma estratégia.

R: a

Agora sim, a parte mais legal! Hora de programar:

6. Implemente funções (ou métodos de uma classe) que permitam a criação de caixas, círculos e linhas com atributos diferentes (posição inicial, tamanho, massa/densidade, coeficiente de atrito e de restituição).

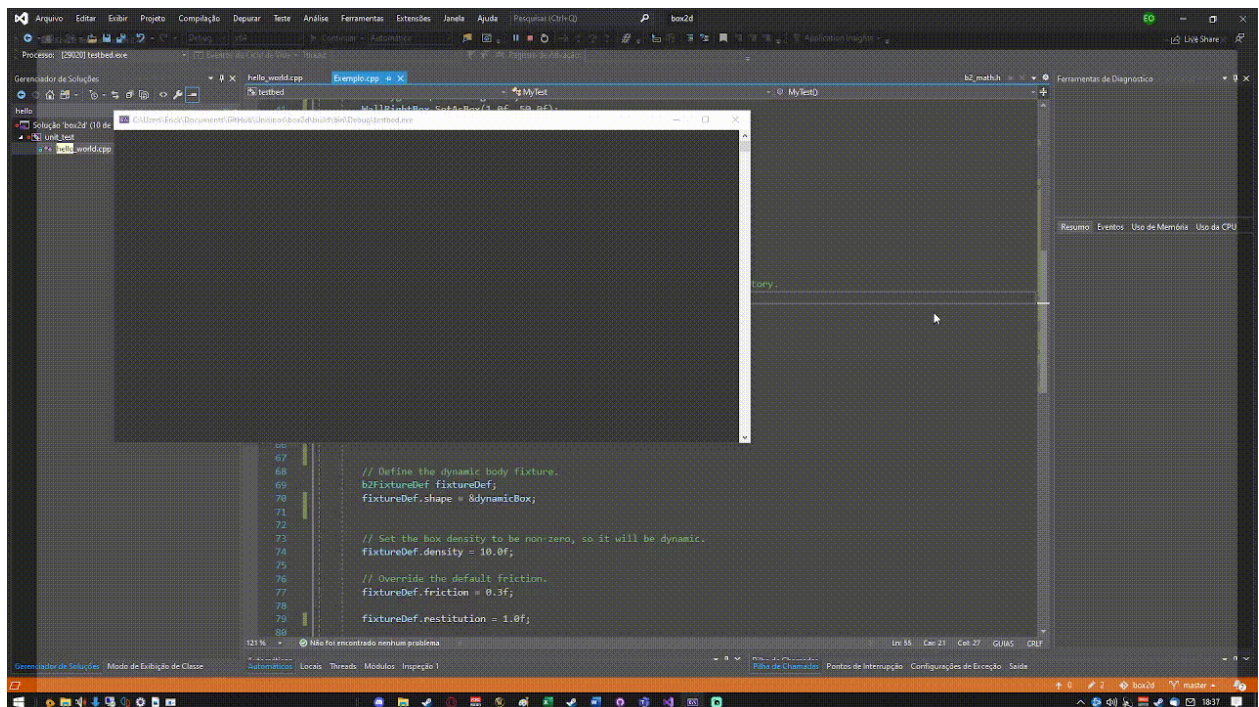
Resposta: Parte da resposta está no “void Keyboard()” e parte nas próprias classes: createBox, createCircle e createEdge.

7. Crie uma "borda" ao redor de sua cena (paredes, chão e teto), de maneira que os objetos criados não fujam da janela. Crie uma caixa ou esfera no centro da cena. Altere o vetor da aceleração da gravidade (intensidade, direção e sentido) e veja o que acontece.

Resposta: Eu fiz o exercício 7 antes do 6, então acabei fazendo uma primeira versão com cubos (gif abaixo) e depois refiz com edges, ambas estão marcadas como “//EXERCICIO 7” no .cpp.

Versão com cubos: linhas 7 a 34

Versão com edges: linhas 36 a 51



8. Faça um programa que crie caixas, círculos e linhas quando o usuário pressiona as teclas 'b', 'c' e 'l'. Os atributos dos objetos devem ser sorteados.

Resposta: A resposta está em “void Keyboard(int key) override”

9. Crie uma caixa no centro da cena, cada vez que o usuário pressiona a tecla 'b'. O coeficiente de restituição da primeira caixa deve ser 0, e a medida que novas caixas forem criadas, seus coeficientes de restituição devem ser maiores, até atingir 1.

Resposta: O código está no “void Keyboard”, eu coloquei como a tecla V, para ser uma nova função e não substituir a do exercício anterior.

10. Crie uma rampa (uma linha inclinada). Próximo a sua extremidade superior, crie uma caixa sempre que o usuário aperta a tecla 'b'. O coeficiente de atrito dessas caixas deve ir aumentando a cada caixa criada, começando em zero e indo até 1.

Resposta: O código está no “void Keyboard”, eu coloquei como a tecla G, para ser uma nova função e não substituir a do exercício anterior.

11. Crie 6 caixas empilhadas. Ao lado, crie uma pilha de 6 círculos.

Resposta: O código está no “void Keyboard”, eu coloquei como a tecla H, para ser uma nova função e não substituir a do exercício anterior.

12. Crie um "muro" de 10x10 caixas.

Resposta: O código está no “void Keyboard”, eu coloquei como a tecla J, para ser uma nova função e não substituir a do exercício anterior.

13. Crie um corpo rígido com mais de uma *fixture* (como por exemplo os patos abaixo).

Resposta: Parte está na função “createDiamond()” e está no “void Keyboard”. Segue abaixo também as anotações pra chegar no raciocínio do desenho.

