

Exercício - Allegro

Professor: Isaac Franco Fernandes

Crie um *Game Engine* capaz de realizar os seguintes recursos:

Etapa 1)

- Tenha uma classe base chamada `GameObject` com “nome”, “posicao_x”, “posicao_y”, “velocidade_x”, “velocidade_y” e “atrito”.
- `GameObject` deve ter 4 métodos: `Start()`; `Update()`; `Render()`;
 - `Start()` roda somente uma vez quando o objeto é criado
 - `Update()` roda a cada frame e deve conter as mudanças físicas de movimento;
 - `Render()` também roda a cada frame e deve conter as rotinas de desenho do objeto na tela. No `GameObject Render()` não faz nada, deixando esse comportamento para suas subclasses
- Crie a subclasse `Circulo` que herda de `GameObject` e adiciona novos atributos necessários, além de implementar corretamente o método `Render()`;
- Crie a subclasse `Retangulo` que herda de `GameObject` e adiciona novos atributos necessários, além de implementar corretamente o método `Render()`;
- Crie uma classe chamada `GameObjectList` que deve implementar uma lista de ponteiros de `GameObject`, permitindo adicionar, remover e buscar um (`GameObject *`) por nome;
- Em `GameObjectList` crie os métodos `Update()` e `Render()` que percorre todos os elementos cadastrados executando o método `Update()` e `Render()`, respectivamente, de cada um deles.
- Crie um jogo que teste seu *Game Engine versão 0.1-alpha*

Etapa 2)

- Na classe `GameObject`, `Circulo` e `Retangulo` crie um método chamado `MouseDown(int x, int y)` que deve ser chamada quando se clicar em um dos Objetos, passando as coordenadas (x, y) onde ocorreu o clique.
- Na classe `GameObjectList` crie um método também chamado `MouseDown(int x, int y)`, onde (x, y) é o local onde ocorreu o clique. Esse método deve percorrer todos os elementos da lista verificando se o clique nas coordenadas (x, y) atingiu o elemento atual. Se atingir o método `MouseDown(x, y)` desse elemento deve ser invocado. Assim que o primeiro elemento for atingido a busca deve finalizar (um clique só atinge um elemento). Repare que cada figura geométrica tem sua forma de verificar se um ponto (x,y) se encontra dentro dela.
- Nas classes `GameObject` crie um método chamado `AplicarForca(float x, float y, float forca)` que modifica as velocidades_x e velocidades_y do elemento. Deconsidere a forma geométrica do elemento, considere que a força é aplicada no elemento por igual em todos os seus pontos.
- Crie um jogo que teste os novos recursos do seu *Game Engine versão 0.2-beta*

Etapa 3) – desafio

- Crie um sistema de detecção de colisão entre os elementos do jogo(`GameObject`). Esse sistema pode ser um método dentro de `GameObjectList` chamado `DetactaColisao()` que verifica se cada um dos elementos colidiu com cada um dos outros elementos. Se colidiu um método chamado `ColisaoEncontrada(GameObject *objeto1, GameObject *objeto2)` é chamada e deve decidir o que fazer. Inicialmente o que pode ser feito é reposicionar os objetos para que eles não colidam. Uma estratégia é voltar os objetos para sua posição no frame anterior (que precisa ser armazenada). Uma outra estratégia é utilizar a física para definir a nova posição dos dois objetos que colidiram de acordo com as forças aplicadas.