Pontifícia Universidade Católica de Minas Gerais

Algoritmos e Estrutura de Dados

-

Lista Prática - Módulo I

1. Quando organizamos a hierarquia de GameObjects dentro da Unity, também definimos as relações de parentesco entre diferentes objetos. Ocasionalmente torna-se interessante encontrar, dentre os filhos de um objeto, quais possuem um determinado componente. A função ***'GetComponentsInChildren<C>()'*** retorna, para um determinado Transform, um Array com referências para todos os componentes do tipo ***C*** dentre seus filhos (e filhos de seus filhos.)

   Implemente sua própria versão desta função que, para um tipo de componente de sua escolha (e.g.: Rigidbody, Collider, SpriteRenderer, etc.), a função recursivamente constrói uma Lista de componentes deste tipo utilizando 'GetComponent<>()'. Esta lista retornada deve conter todas as referências para componentes deste tipo em objetos da hierarquia de um mesmo Transform.

2. Implemente uma função que para uma determinada lista de classes de sua escolha, esta função embaralha os elementos dessa lista em uma ordem aleatória (dica: pesquise sobre a classe 'Random').

3. Construa uma cena na Unity que represente uma fila de clientes em um banco (Utilize quaisqueres abstrações que desejar para ilustrar a cena.) Cada cliente é representado por um GameObject próprio que possui uma classe 'Cliente'. Esta classe deve conter:

CPF do cliente
Tempo mínimo que este cliente gasta para realizar transações no banco em minutos.
Tempo maximo que este cliente gasta para realizar transações no banco em minutos.

   Implemente uma interface gráfica com três botões que realizam as seguintes operações para uma fila de clientes aleatória e sem repetições. O resultado deve ser mostrado em texto na tela:

   a. Chama o próximo cliente da fila, calculando um tempo aleatório gasto pelo cliente em suas operações que deve estar entre os limites de máximo e mínimo.
   b. Mostra na tela quantos clientes restam na fila, e o tempo total gasto por todos os clientes que já foram processados.
   c. Imprime um Debug.Log com o CPF de cada cliente restante na fila (se estiver zerada, informar.)

4. Para cada cliente, adicione uma variável do tipo string contendo uma expressão matemática de números inteiros e símbolos de adição (+) e subtração (-). Esta expressão representa as operações feitas pelo cliente em sua conta.

   Quando o botão de chamar um cliente na fila for pressionado, o programa deve armazenar cada termo desta expressão em uma **pilha**, e resolver esta expressão por desempilhar cada termo. O resultado deve ser demonstrado como outra caixa de texto na tela.