Aquário

Simulação gráfica em Unreal Engine 4 com ligação a CUDA, Inteligência Artificial e Base de Dados Cloud

Pedro Miguel Medinas Fresco

Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Beja, Portugal

Introdução

Simulação gráfica de vários aquários, possibilitando a movimentação dos peixes de forma realista e a entrada do utilizador dentro do aquário. São utilizadas várias tecnologias, com o objetivo de juntar diversos recursos proporcionados por cada uma delas.

Foi criado um mapa, onde se pode percorrer uma floresta visualizando o voo dos pássaros e o edifício dos aquários.

Tecnologias Usadas

1. Motor de Jogo, Unreal Engine 4
2. Plataforma de computação paralela e modelo de programação CUDA
3. Linguagem de programação, C++
4. Linguagem de programação, Python
5. Controlador de versões, Github
6. Website com Modelos 3D, sketchfab
7. Programa para modelagem, animação, texturização e composição, blender
8. Base de Dados, Firebase
9. IDE, Visual Studio 2019

Objetivos

. Uso de um Motor de Jogo para criação de objetos 3D
. Utilização CUDA em funções que são paralelizáveis
. Fazer ligação entre várias tecnologias.
. Criar vários aquários virtuais e peixes.
. Possibilitar a movimentação dos peixes dentro do aquário.
. Impedir os peixes de saírem do aquário.
. Inserir inteligência artificial (IA) nos peixes de modo a criar percursos realistas.

Ligação UE4-CUDA-Python

Devido a inexistência de compatibilidade nativa do UE4 em relação a CUDA e Python, descobriu-se a possibilidade de criar uma biblioteca estática de C++ com as funções de CUDA e os scripts de Python através de CPython, possibilitando chamar estas funções dentro do UE4.

Para se conseguir fazer esta ligação é necessario guardar a biblioteca e as importações de outras bibliotecas na pasta do projeto UE4.
Modificou-se o ficheiro "Nome Do Projeto".Build.cs dentro do projeto do UE4 e acrescentou-se aos PublicAdditionalLibraries e PublicIncludePaths o percurso para biblioteca e as importações de outras bibliotecas respetivamente, e acrescentou-se o cudart.lib e python.lib e as suas respetivas bibliotecas importadas nos dois. No caso do Python é também necessário acrescentar os scripts nas diretorias de trabalho do UE4 e do projeto compilado.

Unreal Engine 4

O UE4 é à base de objetos, sendo os tipos de objetos utilizados os actors, que são objetos colocados estaticamente no mapa sem movimento, os objetos do tipo pawns que são capazes de realizarem movimento e os objetos do tipo characters que são controlados pelo o utilizador.

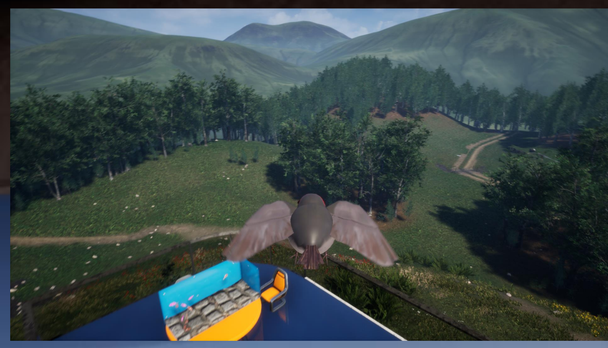
Os objetos são compostos por vários componentes que tem variados objetivos, o StaticMesh/SkeletalMesh, onde se encontra o modelo 3D do objeto, e os componentes baseados em formas geométricas usados normalmente para verificar sobreposição entre objetos de modo a chamar um trigger, função que é chamada em casos específicos.

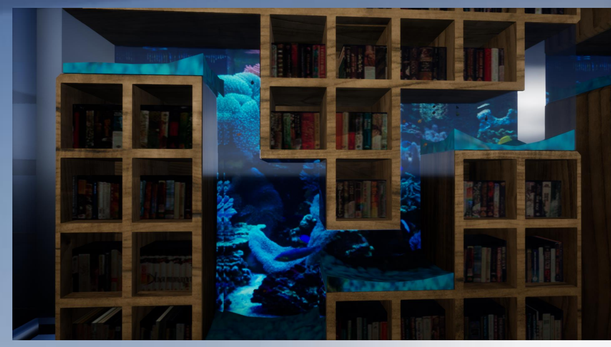
Criando-se 3 objetos, com objetivo de serem os aquários e os peixes, sendo estes:

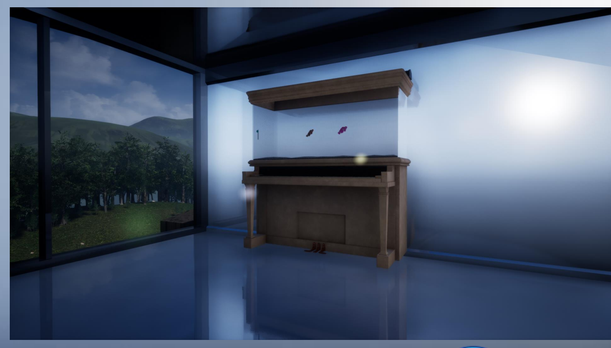
1. **FishTarget**, actor, que vai servir de alvo, com um componente esférico para ocorrer sobreposição com os peixes com o objetivo de dar um volume de sucesso onde o peixe deve seguir para o alvo seguinte. Sem este componente o peixe só seguiria para o alvo seguinte quando chega-se a posição x,y,z exatas onde o alvo esta colocado. Devido a utilizar uma matriz de adjacência para definir os percursos possíveis entre vários alvos e a ordem de como o UE4 vai buscar os alvos ser inconsistente, acrescentou-se a variavel Id.

2. **FishTank**, actor, tem apenas um componente retangular com o simples objetivo de ir buscar todos os FishTarget que estão dentro do volume e organiza por Id. Dentro do editor de unreal quando é criado um herdeiro deste objeto é acrescentado um staticmesh para inserir o modelo do aquário. Também têm variáveis que definem a velocidade que o peixe poderá ter dentro do aquário (CurrentForwardSpeed, Acceleration, TurnSpeed, MaxSpeed, MinSpeed). Este objeto foi criado com o intuito de limitar a busca dos alvos na área do aquário, indispensável ao objetivo de ter mais do que um aquário no mapa.

3. **Fish**, pawn, tem dois componentes, um SkeletalMesh, para colocar o modelo do peixe, e um componente esférico com o objetivo de fazer trigger à sobreposição com o FishTarget para seguir para o próximo FishTarget. Este objeto move-se usando o movimento 3D referido anteriormente. Este objeto está ligado a um aquário, onde foi buscar a velocidade e o array de FishTargets desse mesmo aquário.
Notar que o modo de controle para que alvo se deve dirigir é feito através da criação de um array de inteiros com dimensão 5000 que contem o percurso, o identificador do FishTarget, e deve-se percorre-los por ordem do array, tendo um contador que vai buscar esse identificador para ser usado para indexar o array de FishTarget de modo a encontrar o FishTarget respetivo.







Percurso dos Peixes

O percurso do objeto do tipo peixe, é executado em Python e segue baseado na seguinte logica :

Os vários alvos estão localizados sempre na mesma posição no mapa, converteu-se os vários grafos, um por cada aquário, tendo correspondência entre o número do ponto e o ID do alvo, para várias matrizes de adjacência. Por fim, criou-se uma matriz de tipo, onde se verifica o tipo de alvo (exemplo: tipo A, tipo B, ou tipo C).
Cria-se uma matriz de pesos onde se mostram os pesos de andar entre pontos de vários tipos, sendo esta matriz assimétrica (exemplo: ir de um alvo do tipo A para um do tipo B têm peso 5, e voltar tem peso 10). Usando-se as três matrizes criadas anteriormente, criou-se uma matriz de adjacência com pesos, sendo usado o CUDA Kernel com o algoritmo de Dijkstra para descobrir o percurso mais curto entro dois alvos.
Executando o algoritmo de Dijkstra entre todos os alvos e criando um modelo de inteligência artificial com os resultados do Dijikstra.

Resultados

Existe neste projeto:
. Objeto do tipo MainCharacter, objeto que o utilizador controla
. Objeto do tipo FishTarget, usado como alvo para o movimento
. Objeto do tipo FishTank, o aquário
. Objeto do tipo Fish, o peixe
. 5 modelos 3D para os aquários
. 3 modelos 3D para os peixes
. 12 texturas para os peixes
. 1 floresta virtual
. 1 casa virtual
. 1 biblioteca estática de C++ capaz de compilar e correr CUDA e Python.
. Projeto de Firebase, com Cloud Firestore (base de dados com as matrizes dos aquários) e Storage (onde os ficheiros com os modelos de IA estão guardados)
. Uma função para fazer a conexão à base de dados Firebase
. Script de Python para criar um aquário na base de dados
. Script de Python para criar os modelos de IA com os percursos, usando três CUDA kernels para criação das matrizes necessárias e o algoritmo de Dijkstra
. Script para criar o percursos do objeto do tipo peixe