Relatório – Lista 4

Aluno André Mazal Krauss, 1410386

Disciplina INF2609 – PUC-Rio

Professores Bruno Feijó e Antônio Furtado

Instruções de Execução

1. Executar Build/Build.exe

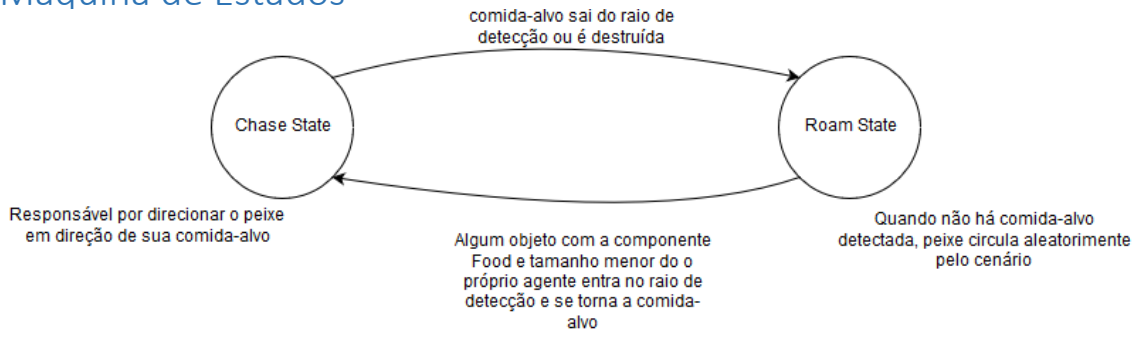
O projeto do Unity também está incluído e deve ser aberto selecionando-se a pasta GameAI_Lista4.

No jogo, diversos peixes competem entre si para comer e crescer de tamanho. Para isso, os peixes procuram comer a comida espalhada pelo Level, e também comer outros peixes que sejam menores do que si. Os peixes vermelhos são controlados por uma máquina de estados, enquanto que os peixes verdes usam uma *behaviour tree.* Implementei tanto a Máquina de Estados quanto a b*ehaviour tree* diretamente por meio de scripts do Unity: a máquina de estados é implementada pelos scripts Fish.cs, ChaseState.cs e RoamState.cs; já a BT é implementada em Fish_Tree.cs. Ambos os scripts usam um raio de detecção para obter uma comida-alvo, que pode ser uma comida ou um peixe menor do que si próprio.

Considerações

Para ambas as implementações, marcante é o fato de que o comportamento desejado é muito simples. Vejo isso, somado ao fato de eu não ter usado nenhuma ferramenta especializada, como um dos motivos para ter preferido enormemente a máquina de estado. Minha implementação “caseira” da máquina de estados não só foi simples e rápida de fazer, como utiliza uma componente no Unity para cada estado, o que é mais modular, fácil de debugar e bem refletido visualmente no inspetor do Unity. Como o comportamento era simplista, foi implementado em somente dois estados. Com uma behaviour tree, a maior dificuldade foi implementar a estrutura da árvore, e não o comportamento em si. Minha solução “caseira” usa co-rotinas, aglutina tudo no mesmo arquivo e representa cada nó sequencial com um loop for. Ou seja, pouco organizado e escalável. Teria que ter um trabalho inicial mais robusto para realmente aproveitar o potencial das *behaviour trees.*

Máquina de Estados



Behaviour Tree