

ANÁLISE DO USO DE ANIMAÇÃO COMPORTAMENTAL COM O MOTOR DE JOGOS UNITY

Aluno(a): João Marcos Estevão

Orientador: Dalton Solano dos Reis

Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Requisitos Funcionais e Não Funcionais
- Implementação
- Resultados
- Conclusões e Sugestões

Introdução

- Uso de simuladores no âmbito educacional.
 - Facilitar aprendizado
 - Permitir reprodução de processos lentos ou perigosos
- Simulação de seres vivos

Objetivos

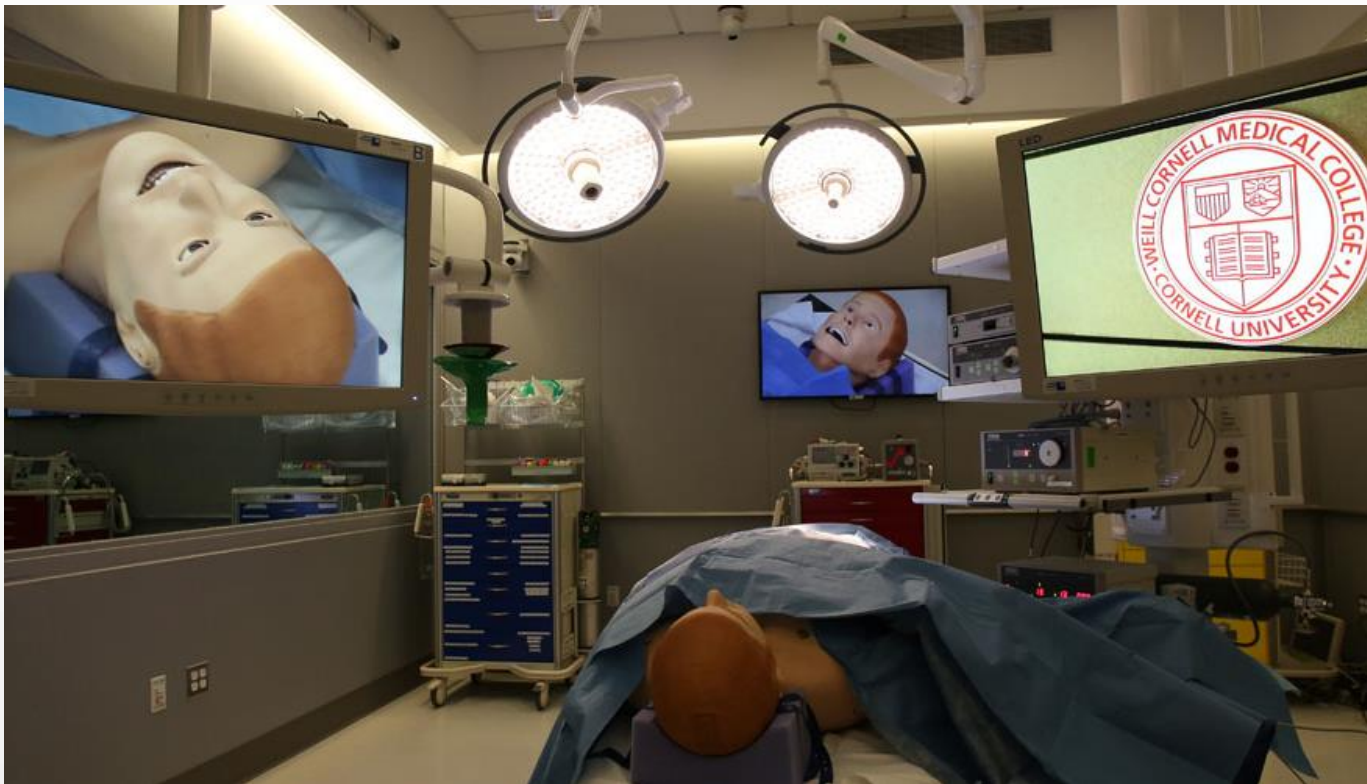
- Extensão do trabalho do EcosAR através da adição de animais
- Desenvolvimento de inteligência artificial utilizando aprendizado de máquina por meio da biblioteca Unity ML-Agents
- Aplicação da inteligência artificial aos animais

Fundamentação Teórica

- Simuladores:
 - Ambiente virtual executado de forma mais próxima possível a do mundo real
 - Seguro e de baixo custo
 - Manipulação de condições para estudo
 - Uso frequente nas áreas de saúde, negócios e militar

Fundamentação Teórica

Simulador do departamento de Anestesiologia do Weill Cornell Medical Center



Fundamentação Teórica

- Animação comportamental:
 - Conceito de controle de ações de objetos em ambientes computacionais de acordo com um comportamento dotado de inteligência artificial.
 - Percepção, Raciocínio e Ação

Fundamentação Teórica

Simulador de multidões Massive



Fundamentação Teórica

- Unity Machine Learning Agents:
 - Permite que jogos e simulações sirvam como ambientes para treinar agentes inteligentes
 - Sensor, Agent e Academy
 - Sistema de recompensas

Trabalhos Correlatos

- VisEdu – Aquário Virtual: Simulador de ecossistema utilizando animação comportamental
 - Kevin Eduardo Piske (2015)
 - Ecossistema marinho de aquário
 - Cadeia alimentar de três componentes
 - Javascript e HTML5

Trabalhos Correlatos

- VisEdu-SIMULA 1.0: Visualizador de material educacional, módulo de animação comportamental
 - Gustavo Rufino Feltrin (2014)
 - Extensão do trabalho de Harbs (2013)
 - Simulador 2D para geração de animação comportamental
 - Java e HTML5

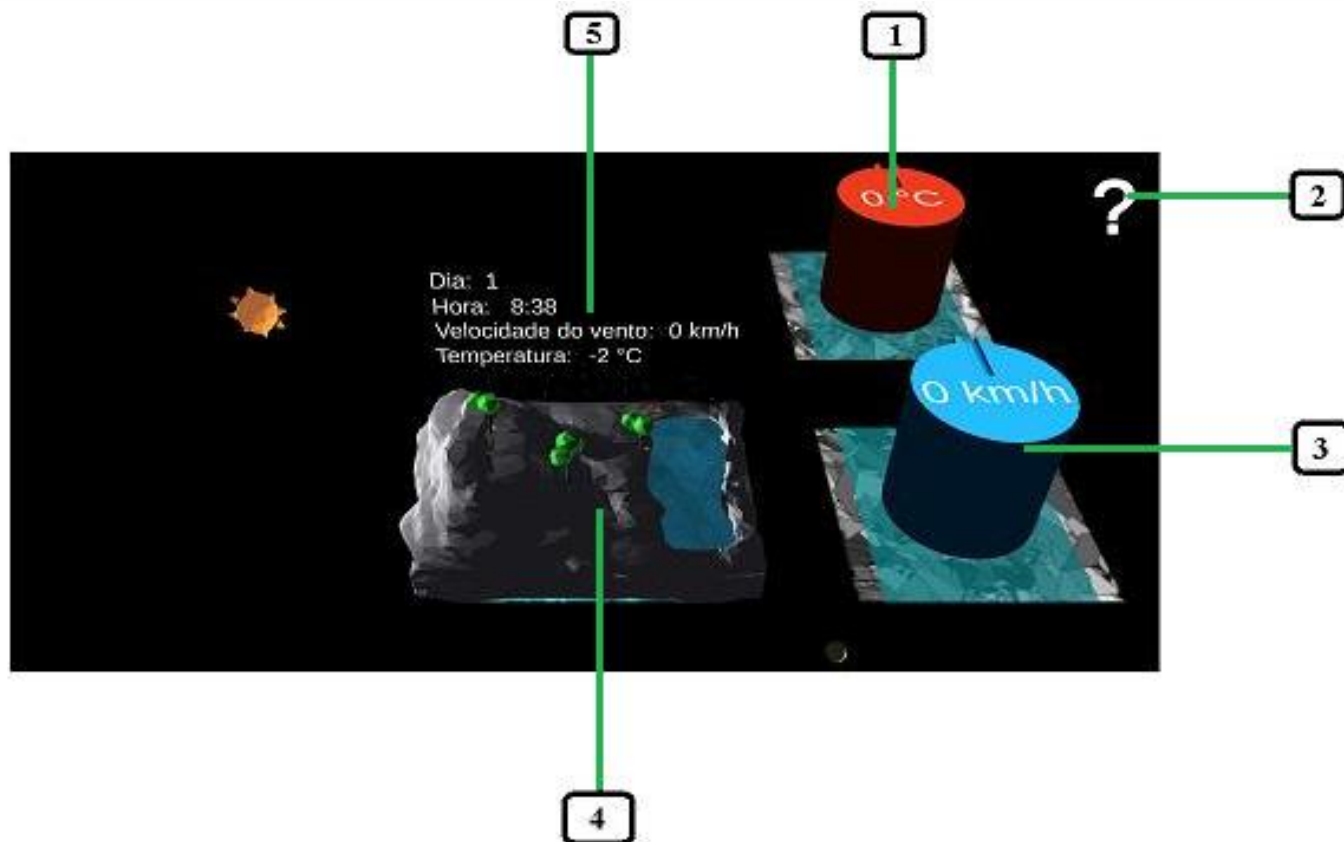
Trabalhos Correlatos

- Simulador de um ambiente virtual distribuído multiusuário para batalhas de tanques 3D com inteligência baseada em agentes BDI
 - Germano Fronza (2008)
 - Ambiente multijogador
 - Tanques controlados pelo computador
 - Java e OpenGL

Versão Anterior do Aplicativo

- EcosAR – Simulador de Ecossistemas Utilizando Realidade Aumentada.
 - Rodrigo Wernke Pereira (2019)
 - Realidade aumentada com uso do Vuforia
 - Permite ao usuário interagir com o ambiente
 - Desenvolvido com o Unity
 - Controle de vento e temperatura e alterações no cenário decorrentes destas variações

Versão Anterior do Aplicativo



Requisitos Funcionais

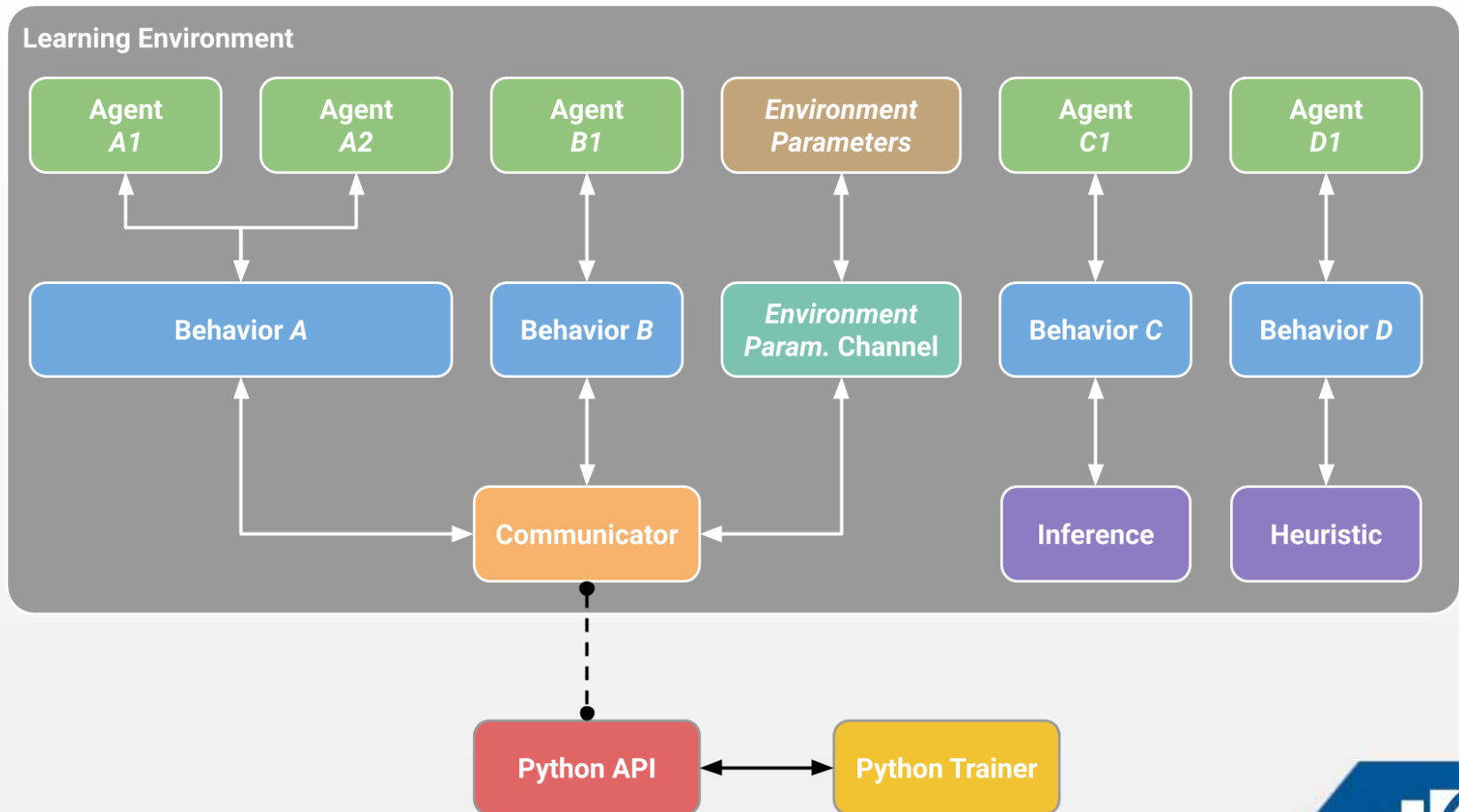
- Possuir três espécies de animais
- Permitir que os animais interajam uns com os outros
- Permitir que os animais se alimentem utilizando os recursos (plantas e água) disponíveis no mundo virtual
- Fazer com que os animais possuam um controle de sede e fome e se comportem de acordo com isso
- Excluir animais que morram na simulação
- Fazer com que os animais interajam com as alterações no ecossistema efetuadas pelo usuário como velocidade do vento e temperatura

Requisitos Não Funcionais

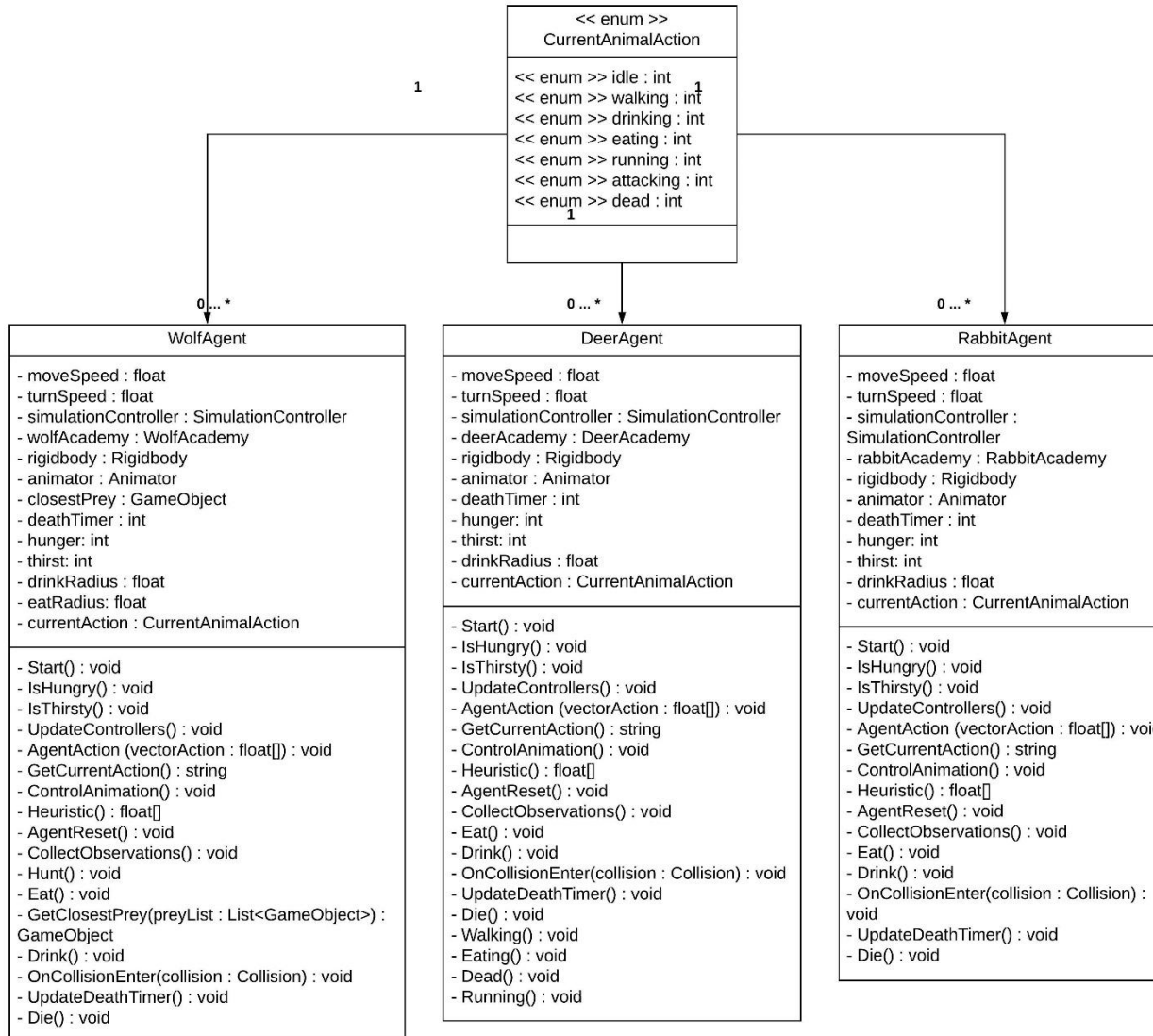
- Ser implementado com a linguagem C#
- Utilizar o motor gráfico Unity
- Utilizar a biblioteca Unity ML-Agents
- Trabalhar em espaço de três dimensões (3D)
- Utilizar a biblioteca Vuforia

Especificação

Diagrama de representação do funcionamento do ML-Agents



Especificação



Implementação

Behavior Parameters (Script)

Behavior Name: DeerLearning

Vector Observation

Space Size: 5

Stacked Vectors: 1

Vector Action

Space Type: Discrete

Branches Size: 3

Branch 0 Size: 2

Branch 1 Size: 3

Branch 2 Size: 3

Model: DeerLearning (NNModel)

Inference Device: CPU

Behavior Type: Default

Team ID: 0

Use Child Sensors:

Deer Agent (Script)

Max Step: 5000

Reset On Done:

On Demand Decisions:

Decision Interval: 4

Script: DeerAgent

Move Speed: 0,17

Turn Speed: 180

Ray Perception Sensor Component 3D (Script)

Script: RayPerceptionSensorComponent3D

Sensor Name: RayPerceptionSensor

Detectable Tags

Size: 2

Element 0: Water

Element 1: Tree

Rays Per Direction: 3

Max Ray Degrees: 60

Sphere Cast Radius: 0,1

Ray Length: 1

Ray Layer Mask: Mixed ...

Observation Stacks: 1

Implementação

```
public override void AgentAction(float[] vectorAction)
{
    if (currentAction == CurrentAnimalAction.dead)
        return;

    currentAction = CurrentAnimalAction.idle;

    // Converte a primeira ação em movimento para frente
    float forwardAmount = vectorAction[0];
    if (forwardAmount > 0)
        currentAction = CurrentAnimalAction.walking;

    // Converte a segunda ação em virar pra esquerda ou pra direita
    float turnAmount = 0f;
    if (vectorAction[1] == 1f)
    {
        turnAmount = -1f;
    }
    else if (vectorAction[1] == 2f)
    {
        turnAmount = 1f;
    }

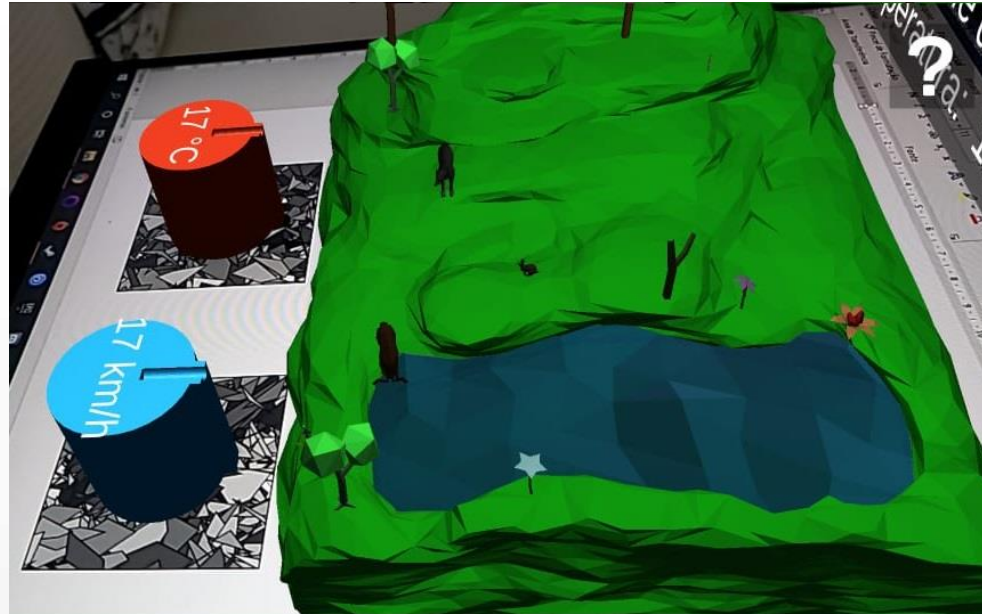
    // Converte a terceira ação para a chamada do método correspondente (1 para beber, 2 para comer)
    if (vectorAction[2] == 1f)
    {
        Drink();
        ControlAnimation();
        return;
    }
    else if (vectorAction[2] == 2f)
    {
        Eat();
        ControlAnimation();
        return;
    }

    // Aplica o movimento
    rigidbody.MovePosition(transform.position + transform.forward * forwardAmount * moveSpeed * Time.fixedDeltaTime);
    transform.Rotate(transform.up * turnAmount * turnSpeed * Time.fixedDeltaTime);

    // Aplica um pequena recompensa negativa para encorajar o personagem a fazer uma ação
    AddReward(-1f / agentParameters.maxStep);

    ControlAnimation();
}
```

Operacionalidade



Análise dos Resultados

- Eficiência do Unity ML-Agents
- Comportamento dos animais
- Treinamento
- Dificuldades e limitações

Conclusões e Sugestões

- Relevância do trabalho
- Resultados do treinamento
- Uso do Unity ML-Agents

Conclusões e Sugestões

- Sugestões de extensões:
 - Ampliação do cenário do marcador principal do ECOSAR
 - Reprodução dos animais
 - Interação dos animais com os ciclos de dia/noite do cenário
 - Possibilitar adição/remoção de animais no cenário
 - Adição de animais aquáticos e/ou voadores

Demonstração

ANÁLISE DO USO DE ANIMAÇÃO COMPORTAMENTAL COM O MOTOR DE JOGOS UNITY

Aluno(a): João Marcos Estevão

Orientador: Dalton Solano dos Reis