Ali na página três, ali embaixo, perguntei se as alterações do usuário referia-se às alterações efetuadas pelo usuário

Página 4 - sem observações

Na página 5, só fiz uma observação para vc ter cuidado com frases muito longas

Na página 6, apenas uma observação quanto a repetição de palavras. Veja:

Na página 7, a primeira frase do primeiro parágrafo: se aquela informação é sobre o Seu trabalho, penso que deveria estar em resultados/conclusões.

Página 8 - apenas para ter cuidado com a repetição de palavras na mesma frase/ parágrafo

Página 9:

Veja que ali na segunda frase do primeiro parágrafo, fiz uma sugestão: O primeiro é responsável ...

E me perguntei se o quadro 4 seria mesmo um quadro ou uma figura

Apenas sugestões para evitar frases muito longas

Página 11 - sem observações

E página 13 - na referência de:

- Intelligent Agents, 1 é mesmo 1994? Porque no texto está 1995.
- Juliani vc colocou et al. sugiro dizer quem são os autores, assim como vc fez nas outras referências, para padronizar
- Lopes entre Lopes e Oliveira vc usou ; mas nas outras referências vc usou . sugiro padronizar
- Stone não encontrei no texto
- ulicsak- aqui vc tbm coloca :

Página 14 - sem observações

Página 15:

Página 16 - sem observações

Sente falta de fundamentan teoricemente o uno de jogos/onmoigs ne

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

João Marcos Estevão, Dalton Solano dos Reis - Orientador

Curso de Bacharel em Ciência da Computação Departamento de Sistemas e Computação Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

jmestevao@furb.br, dalton@furb.br

Resumo: Este artigo descreve o processo de desenvolvimento de uma extensão de um simulador de ecossistemas através da adição de animais treinados utilizando aprendizado de máquina. Os animais foram treinados para se alimentar e saciar a sede utilizando recursos do cenário e interagindo uns com os outros. O trabalho foi desenvolvido utilizando o motor gráfico Unity em conjunto com a biblioteca Unity Machine Learning Agents. A definição dos animais e comportamentos foi feita em conjunto com uma professora do curso de clências Biológicas. Após realizados os testes e validações, a implementação e o reinamento dos animais se mostrou eficiente e as tecnologia utilizada se mostrou apropriada, ?

Palavras-chave: Simulador, Ecossistemas, Animais, Aprendizado de maquina, Unity.

Com o rápido avanço da tecnologia na última década, o seu uso foi se tornando mais frequente em diversos ramos da sociedade. Um dos ambientes refetados por esse período de informação digital é a educação e a expansão do uso da tecnologia trouxe muitas formas de melhorar e diversificar o aprendizado em diversas áreas, entre elas a de estudo do meio ambiente. Com o objetivo de facilitar o aprendizado de temas relacionados à vários temas dos quais a observação de fatores no mundo real é complexa, foram desenvolvidos simuladores (GREIS, 2010). Entre eles estão os simuladores de conssistente, que tem como inpuito demonstrar de forma facilitada o funcionamente de diversos aspectos simuladores de ecossistema, que tem como intuito demonstrar de forma facilitada o funcionamento de diversos aspectos do meio ambiente da maneira mais próxima possível a que existe na natureza.

Aldrich (2009) define simuladores como ambientes estruturados, abstraídos de alguma atividade da vida real, que permitem aos participantes praticar suas habilidades no mundo real, pois fornecem um feedback apropriado em um ambiente cujos resultados são controlados e previsíveis. Para Greis (2010), as vantagens em se trabalhar com modelos simulados por computador no campo educacional são muitas. Entre elas tem-se a oportunidade de tornar possível a reprodução de processos lentos ou perigosos de ser reproduzir no ambiente natural, o controle das etapas necessárias para a observação dos fenômenos e até mesmo a redução dos custos envolvidos no projeto.

Segundo Dautenhahn e Nehaniv (2002) um dos aspectos que pode apresentar uma dificuldade no desenvolvimento de simuladores de ecossistemas, no entanto, é simular o comportamento existente nos seres vivos que compõem estes ecossistemas, visto que são criaturas providas de inteligência. Para este fim, podem ser utilizadas técnicas de inteligência artificial de forma a tentar reproduzir comportamentos compatíveis com o de criaturas encontradas na natureza.

(...) imagine quão impressionante seria criar um robô que pudesse emular - isto é, observar a (...) imagine quao impressionante seria criar um robo que pudesse emular – isto e, observar a mudança realizada no ambiente e desenvolver sua própria maneira de reproduzir aquela mudança de estado. Nós presumimos que isso seria um desafio muito mais sério para os programadores do que desenvolver um robô que copie as ações de outros de forma não criativa. (DAUTENHAHN; NEHANIV, 2002, p. 226, tradução nossa)

Decorrente destas considerações, o trabalho aqui apresentado tem como objetivo desenvolver uma inteligência artificial com aprendizado de máquina utilizando o Unity ML-Agents para ser aplicada em animais no simulador de ecossistema desenvolvido por Pereira (2019), afim de possibilitar que os agentes no cenário sejam capazes de interagir de forma autônoma, seguindo regras pré-determinadas para cada cenário. Os objetivos específicos são: adicionar animais no ECOSAR; desenvolver uma inteligência artificial utilizando aprendizado de máquina através do Unity ML-Agents para fazer com que os animais interajam com o ecossistema e uns com os outros de maneira inteligente.

1 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os aspectos da fundamentação teórica utilizados para o desenvolvimento deste trabalho. Na seção 2.1 é falado sobre simuladores. Na seção 2.2 é apresentado o conceito de animação comportamental. A seção 2.3 explica sobre a biblioteca Unity Machine Learning Agents. Na seção 2.4 está a apresentação do aplicativo que será continuado neste trabalho, o ECOSAR desenvolvido por Pereira (2019). Por fim, na seção 2.5 estão apresentados os trabalhos correlatos com o trabalho relatado neste artigo. or como que halado

1 Oton Trabalho de Conclusão de Curso - Ano/Semestre: 2020/1

min

agre

ambiente simulado (JULIANI et al., 2018). Isso pode ser definido tanto via chamadas de scripts do Unity quanto ao ser alcançado o número máximo de passos pré-definido.

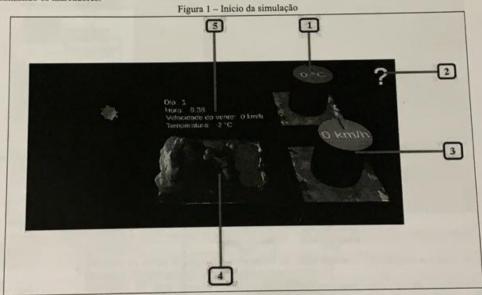
Juliani et al. (2018) explica que a Academy é um singleton e é utilizada para acompanhar o número de passos da simulação e gerenciar os agentes, além disso ela possui a capacidade de definir parâmetros de ambiente, que podem ser utilizados para mudar a configuração do ambiente em tempo de execução.

1.4 VERSÃO ANTERIOR DO APLICATIVO

O "ECOSAR – Simulador de Ecossistemas Utilizando Realidade Aumentada" foi desenvolvido por Pereira (2019) e tem como proposta criar um ambiente virtual utilizando realidade aumentada que simule um ecossistema real. Além disso, o objetivo foi permitir ao usuário interagir com este ambiente através da alteração do comportamento da simulação utilizando-se de elementos como temperatura, velocidade do vento e controle do ciclo dia/noite. Para o desenvolvimento do projeto foi utilizado o motor gráfico Unity em conjunto com a biblioteca Vuforia, além das ferramentas Photoshop CC 2019, AR Marker Generator para a geração dos marcadores e Blender para a modelagem dos objetos. Entre as propostas de extensão expostas por Pereira (2019) está a de inserção de animais na simulação. Este tópico-será um dos objetivos deste artigo, visto que os animais serão os agentes que utilizarão o módulo de animação comportamental desenvolvido.

O aplicativo disponibiliza para o usuário uma forma de simular comportamentos de um ecossistema, concedendo a ele o controle de elementos que existem na natureza, como o vento e a temperatura. Para controlar e mostrar estes elementos, o aplicativo faz uso da câmera do dispositivo móvel, utilizando-a em conjunto com marcadores (que funcionam em conjunto com a biblioteca Vuforia) para visualizar a aplicação (PEREIRA, 2019).

A Figura 1 mostra a simulação ao ser iniciada. Ela começa com a temperatura em zero graus e com a velocidade do vento em zero quilômetros por hora. A partir deste ponto o usuário pode começar a interagir com a cena utilizando os marcadores.



Fonte: Pereira (2019).

Na Figura, l os itens l e 3 são os marcadores de controle do aplicativo, com os quais o usuário manipula a cena. Já o item 2 trata-se do botão de ajuda, que quando ativo mostra a bounding box dos marcadores e ativa os botões virtuais. Estes botões (virtuais de ajuda) são utilizáveis através da Realidade Aumentada para mostrar textos de ajuda para cada marcador. E por fim, o item 4 é o marcador responsável pela visualização da simulação e o item 5 mostra um painel com as características atualizadas da simulação.

A Figura 2 apresenta os possíveis comportamentos que podem ser realizados dentro do aplicativo, sendo que os paralelogramos são os elementos que o usuário pode controlar, os losangos são as condições necessárias para cada processo e os retângulos correspondem aos resultados obtidos através dos controles realizados. Como pode ser visto, dependendo das alterações de usuário, a simulação realiza diferentes comportamentos para diferentes alterações, sendo

Trabalho de Conclusão de Curso - Ano/Semestre: 2020/1

ASSESSMENT OF THE OWNER, WHEN

3

lixadas corretamente denuro da propriedade de mundo suo sur

ário virtual no aplicativo VISEDU



de material educacional, módulo de animação comportamental

2D e do editor de jogos de Harbs (2013) com a criação de um animações comportamentais.

de animações comportamentais com controle de percepção, nagens.

nação Java, com os plugins JBoss Tools e Jasonide;

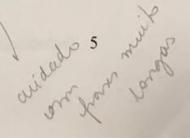
4.1 para a criação do módulo de inteligência artificial;

5 para o servidor de aplicação;

ITML5.

que teve sucesso no objetivo de criar um simulador 2D para a tamentais, caracterizando a orientação a componentes como ponto le Jogos como o ponto fraco, explicando que sua utilização seria de Jogos poderia usar um editor de texto mais sofisticado.

20/1



1.1 SIMULADORES

Para Rosa (2008) os simuladores na área de computação podem ser explicados como a criação de um cenário virtual que crie un ambiente e execute-o de forma mais próxima possível a do mundo real. Segundo Greis (2010) o modelo simulado ofense e execute-o de forma mais próxima possível a do mundo real. modelo simulado oferece várias vantagens, como a possível reprodução de processos muito lentos ou perigosos per metroduzidos no ambiente real, a facilidade da observação de fenômenos e também a redução de custos.

Os simuladores têm sido frequentemente utilizados em países como Alemanha, Reino Unido e Estados Unidos nas áreas de saúde, negócios e militar. Na área militar por exemplo o jogo "America's Army" é considerado uma das maiores fontes de recrutamento militar nos Estados Unidos (LOPES e OLIVEIRA, 2013).

Segundo Ulicsak e Wright (2010) os simuladores oferecem um mecanismo seguro e de baixo custo para treinamentos e atividades a serem realizadas em ambientes perigosos ou que sejam muito dificeis de recriar no mundo real. De acordo com Stone (2012) o alto nível de fidelidade, ou seja, a semelhança com os eventos reais permite esta transferência de ambientes.

1.2 ANIMAÇÃO COMPORTAMENTAL

A animação comportamental é um conceito de controle de ações de objetos em ambientes computacionais de acordo com um comportamento dotado de inteligência artificial.

A animação comportamental busca o realismo a nível de comportamento dos personagens em cena. Neste tipo de animação, os personagens são "atores sintéticos" dotados de personalidade e habilidades próprias. A atuação de um personagem não é mais oriunda exclusivamente de intervenções diretas do animador, mas sim fruto de sua personalidade, seu humor, suas metas e sua interação com os demais. atores. (...) (SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO GRÁFICA E PROCESSAMENTO DE s nas refuncias 1994 IMAGENS, 1993, p. 1)

Os conceitos da animação comportamental são semelhantes aos dos agentes inteligentes. O primeiro conceito é o da percepção, que se resume a um processo de reconhecimento em que se capta informações por sensores externos e executa a transformação dos dados para uso do programa (INTELLIGENT AGENTS, 1995). O segundo conceito é o de raciocinio, que é a capacidade de tomar conclusões sobre um conjunto de hipóteses próprias e/ou de outros (INTELLIGENT AGENTS: AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, 1997). Por fim o terceiro conceito é o de ação, que se trata de um comportamento de resposta a um estimulo externo (INTELLIGENT AGENTS,

A animação comportamental tem dado impeto a uma vasta gama de aplicações capazes de sintetizar, reunir e ensinar comportamentos de animais (EUROGRAPHICS/ACM SIGGRAPH SYMPOSIUM ON COMPUTER ANIMATION, 2005). Entre as áreas de aplicação dos conceitos de animação comportamental pode ser citado o desenvolvimento de jogos de videogame. Para os personagens principais do jogo, que aparecem em tela em periodos extensos, é necessária a utilização de técnicas que tornem o seu comportamento mais realista, visto que o jogador estará mais propenso a observar o padrão de ações destes personagens (INTERNATIONAL CONFERENCE, 2013).

1.3 UNITY MACHINE LEARNING AGENTS

Unity Machine Learning Agents, comumente abreviado para Unity ML-Agents ou ML-Agents, é uma biblioteca de código aberto para o motor de jogos Unity que permite que jogos e simulações sirvam como ambientes para treinar agentes inteligentes. Segundo Juliani et al. (2018), o ML-Agents permite aos desenvolvedores criar ambientes simulados utilizando o editor do Unity e interagir com eles via uma Application Programming Interface (API) em Python.

De acordo com Juliani et al. (2018) o Software Development Kit (SDK) do ML-Agents tem três entidades principais: o Sensor, o Agent e a Academy. O componente Agent é utilizado para indicar que um GameObject dentro de uma cena é um agente, o que significa que pode coletar informações por meio de observação, fazer ações e receber recompensas. As informações são coletadas pelos agentes através de uma variedade de sensores possíveis correspondendo a diferentes formas de informação, como imagens renderizadas e resultados de ray-cast. Cada componente Agent possui uma política que é rotulada com um nome de comportamento. Uma política com o mesmo nome de comportamento pode ser utilizada por qualquer número de agentes, os quais podem executar a mesma política e compartilhar dados de experiências durante o treinamento. Além disso, não há limite para o número de nomes de comportamentos para políticas dentro de uma cena, o que significa que é possível construir cenários com múltiplos agentes com grupos de agentes individuais executando vários comportamentos diferentes.

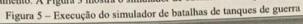
O sistema de recompensas é utilizado para enviar um sinal de aprendizado ao agente e por meio do sistema de scripts do Unity é possível defini-lo e modificá-lo a qualquer momento durante a simulação. Da mesma forma, a simulação pode ser definida como finalizada tanto no âmbito de um agente individual quanto em relação a todo o

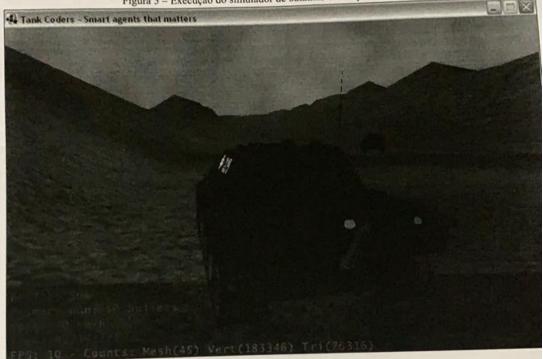
150

101

door

Apesar de encontrar problemas com a performance do programa, obteve-se sucesso em desenvolver os objetivos estabelecidos no artigo. Assim como é proposto neste trabalho, Fronza (2008) utilizou conceitos de inteligência artificial para desenvolver animação comportamental para os agentes do cenário do simulador. Para o controle dos tanques foi utilizado o modelo BDI, que utiliza os conceitos de crenças, desejos e intenções para desenvolver um comportamento. A Figura 3 mostra o simulador de batalhas de tanques de guerra de Fronza (2008).





Fonte: Fronza (2008).

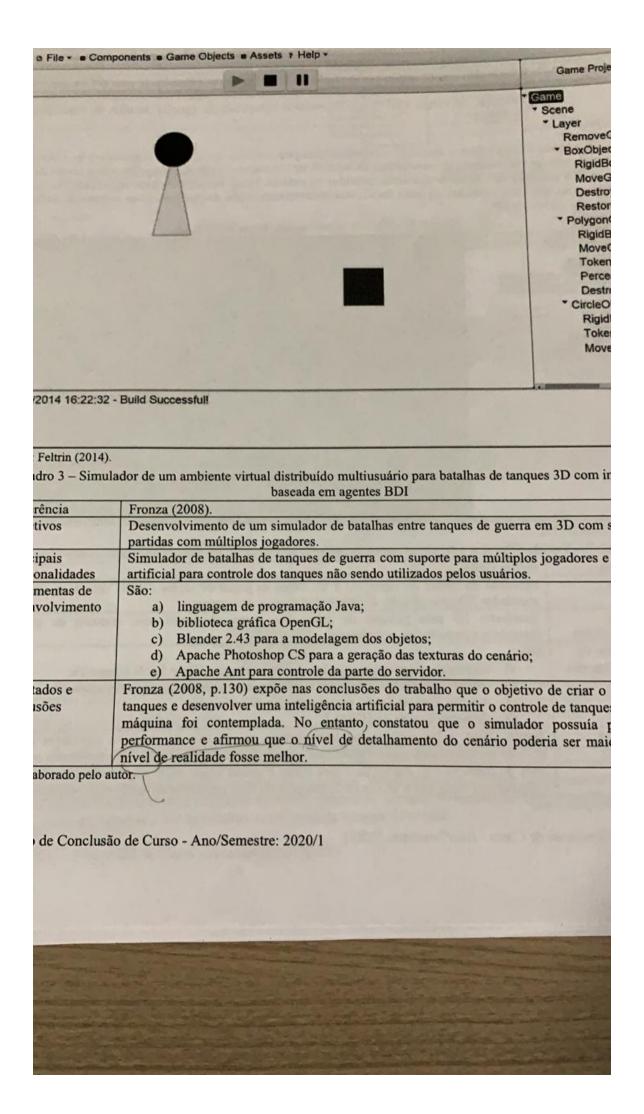
2 DESCRIÇÃO DO APLICATIVO

Este capítulo pretende apresentar os detalhes da implementação do trabalho desenvolvido. Para isso apresentadas duas seções. A primeira seção apresenta a visão geral da implementação, mostrando o funcionamen simulação dos animais a serem adicionados no simulador e explicando a metodologia utilizada para definição animais escolhidos e dos comportamentos a serem implementados. A segunda seção apresenta a implementaç trabalho (desenvolvido em C#), explicando os componentes utilizados e a implementação do aprendizado de má através da utilização da biblioteca do Unity ML-Agents.

2.1 VISÃO GERAL

O desenvolvimento deste projeto consiste na inclusão de animais de espécies distintas no simulaecossistemas desenvolvido por Pereira (2019) e a implementação de uma inteligência artificial para cada a permitindo aos animais interagir com o ambiente e uns com os outros através de um comportamento desenvolv meio de treinamento. Para definir o cenário e os animais que seriam utilizados foi realizada uma reunião professora Roberta Andressa Pereira (PEREIRA, 2020), professora do curso de Licenciatura em Ciências Biológ Universidade Regional de Blumenau. Como resultado da reunião foi desenvolvido um documento com um explicação do projeto de Pereira (2019) e o objetivo deste projeto, além de listar os animais possíveis d adicionados no projeto de acordo com uma pesquisa dos assets disponíveis de animais para o Unity. De acor esse documento em conversa com a Pereira (2020) foram definidos os animais e comportamentos a serem utiliza

Os animais selecionados para a inclusão no projeto foram o coelho, o veado e o lobo. A ideia do cenár tanto o coelho quanto o veado possam se alimentar da vegetação que cresce no cenário, de acordo com as conce temperatura, conforme implementação de Pereira (2019). O lobo por sua vez pode se alimentar tanto do vead do coelho e todos os três animais saciam a sede com a água do lago existente no cenário.



Os componentes Behavior Parameters e Ray Perc biblioteca Unity ML Agents. Séndo primeiro responsável por agente assim como o número de ações disponíveis para cada de (controlado pelo usuário ou pela IA) e do modelo de comportan agente. O Ray Perception Sensor Component 3D é ut especificando as tags detectáveis e estabelecendo as dimenso DeerAgent (Figura 6), é onde a lógica principal de controle do Dentro do script do agente (Quadro 4), está o método Agenta controle do animal na cena.

ic override void AgentAction(float[] vectorAction)

Quadro 4) Método Agent Act

turnAmount = 1f;

Drink();

(vectorAction[2] == 1f)

ControlAnimation();

e words grans? nate? figura?

```
Float forwardAmount = vectorAction[0];
(ferwardAmount > 0)
   currentAction = CurrentAnimalAction.walking;
The turnAmount - Of;
 (vectorAction[1] -- 1f)
   turnAmount - - 1f;
else al (vectorAction[1] == 2f)
    turnAmount = 1f;
 (vectorAction[2] == 1f)
   Drink();
   ControlAnimation();
else if (vectorAction[2] == 2f)
   Eat();
  ControlAnimation();
rigidbody. MovePosition(transform.position + transform.
transform.Rotate(transform.up * turnAmount * turnSpeed
AddReward(-If / agentParameters.maxStep);
(ontrolAnimation();
```

Fonte: elaborado pelo autor.

Neste método (AgentAction) é feita a interprese recompensas. No caso dos animais adicionados, o vetor decisões que o agente deve tomar. Estas decisões são algoritmo de aprendizado de máquina. No caso de cada deve se movimentar para frente ou não, portanto são dua com o valor zero, que significa não fazer nada ou um, tomada é se o agente deve se virar para esquerda, para decisão portanto são três ações possíveis. A terceira decinada, sendo que no caso do lobo há uma quarta alternativo

Trabalho de Conclusão de Curso - Ano/Semestre: 2020/

O método AgentAction irá então receber esse vetor de ações e irá executar essas açõe por ex mplo, se o lobo decidir caçar, o método Hunt será chamado e da mesma forma as outras ações executam seus respectivos métodos. fr beber ou se Além da execução há também a recompensa por determinadas ações. Se o animal estiver com secuestiver com fome e comer será recompensado positivamente. Se o agente estiver com fome, sede ou se tentar realizar as ações de comer e/ou beber enquanto está cheio será recompensado no activamente. De acordo com as recompensas positivas e negativas o agente vai treinando seu comportamento, e após ca la sessão de treinamento é gerado um arquivo com um modelo de comportamento treinado utilizando os dados u sess o de treinamento, que pode então ser inserido no agente da cena. Com um modelo de comportamento adicionado a seu objeto de cena, o agente passa a tomar as decisões de acordo com a inteligência artificial treinada anteriormente. O diagrama de classes do projeto está disponível no Apêndice A.

Para poder realizar o treinamento dos agentes é necessário rodar o programa dentro do ambiente do Unity, porém fe am encontradas dificuldades de rodar o ECOSAR devido à utilização do V. Devido a esta situação e para mon ar um ambiente de treinamento melhor sem precisar alterar o projeto ori final, foi cecidido criar um projeto o tre namento dos animais. Neste projeto foram adicionados apenas os compon intes esse iciais para o treinamento, como o terreno, água e vegetação, além da inclusão de informações adicionais na execução do programa para facilitar a visualização do progresso do treiname .o, .Cor o número de vezes que o animal comeu e bebeu, atual placar do treinamento e número de vezes que o treiname to foi reiniciado. Além disso para agilizar e facilitar o processo de treinamento é possível multiplicar as instâncias do cenário na cena, fazendo com que vários cenários de ecossistema executem ao mesmo tempo e dessa forma todos participem do treinamento simultaneamente, acelerando o processo de aprendizagem do agente e consequentemente o alcance do modelo de comportamento desejado. O treinamento de um agente é composto de várias etapas, visto que cada vez que o agente executa o número de passos limite estabelecido na configuração do treinamento, um episódio é finalizado e outro é iniciado. A cada episódio o agente realiza novamente o treinamento e vai utilizando a experiência das recompensas recebidas para realizar o aprendizado. A Figura 7 mostra três etapas diferentes do treinamento sendo executado no projeto de treinamento com o agente do coelho. A configuração para permitir a execução do treinamento está disponível no Apêndice B.

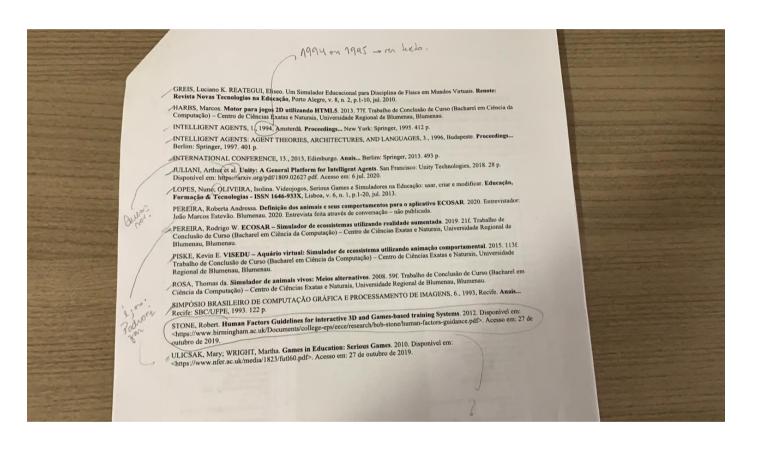
Figura 7 - Evolução do treinamento do agente do coelho

2.2 IMPLEMENTAÇÃO

motor gráfico Unity em conjunto com a biblioteca Unity Machine Learning Agents. E o ambiente de desenvolvimento Visual Studio 2017 com o gerenciador de pacotes e ambientes Python Anaconda 2019.10, para interação com a API em Para o desenvolvimento do comportamento dos animais e adição dos modelos no ECOSAR foram utilizados o Python do Unity ML-Agents responsável pelo treinamento dos agentes.

no âmbito principal da cena o componente Academy, utilizado para gerenciar o treinamento de todos os agentes da qual também estão inseridas as árvores e flores. Além da adição dos animais com seus agentes também foi adicionado inseridos no projeto foram adicionados dentro do componente Terrain, que representa o terreno da cena e dentro do aplicação, o qual contém o ambiente do ecossistema. Nele estão contidos os objetos de terreno, contendo nele as plantas; as partículas de água, fogo e neve; as nuvens; e o controle de rotação do sol e da lua na cena. Os animais Dentro do componente SceneTarget do projeto estão contidos todos os objetos do marcador principal da

o Collider (que pode ser utilizado em diferentes formas, no caso deste objeto foi escolhido o formato retangular) é o componente para identificação de colisão com outros objetos da cena por parte do controle de física do Unity. Com relação a parte de treinamento da inteligência artificial dos animais, foram adicionados três componentes, conforme ao objeto o controle de física do Unity, permitindo controle gravitacional, de massa, de rotação e posicionamento do personagem, além de outras funções. Como também para o controle de movimentação dos animais na cena. Por último Collider. O Animator serve para o gerenciamento das animações do personagem enquanto o Rigidbody adiciona Entre os componentes utilizados em todos os três agentes da cena temos o Animator, o RigidBody e o Figura 6 - Propriedades do objeto de cena do veado pode ser visualizado na Figura 6.



3 RESULTADOS

Após o treinamento realizado foram adicionados os modelos de comportamento treinados aos animais na cena. Após a execução do simulador com esses modelos adicionados constatou-se que houve sucesso em obter o comportamento desejado dos animais. Todos as três espécies passaram a se mover e realizar ações de maneira coerente com a proposta con alimento desejado dos animais. Todos as três espécies passaram a se mover e realizar ações de maneira coerente com a proposta, se alimentando e saciando a sede conforme necessário e de maneira eficiente.

No início do desenvolvimento foi encontrado um empecilho para a utilização do Unity ML-Agents no projeto do ECOSAR, visto que era impossível rodá-lo dentro do ambiente do Unity devido a problemas com o Vuforia, o que impossibilitaria a realização do trainemente area de la composição do trainemente de la composição de la composição do trainemente de la composição de impossibilitaria a realização do treinamento em conjunto com a API em Python. Devido a esta situação foi decidido criar um projeto separado baseado no ECOSAR, porém sem os componentes de realidade aumentada do Vuforia. A longo prazo viu-se que não só essa solução era efetiva para realizar o treinamento como também propiciou um ambiente mais făcil de ser manipulado e que facilitou o processo de treinamento, visto que foram removidos a maioria dos componentes não necessários da cena, como efeitos de partículas e transição do sol/lua. Além disso o projeto separado deu maior liberdade para implementação de placares para visualização do andamento do treinamento e da multiplicação de cenários, que acelera o processo de aprendizagem dos agentes.

Um outro problema encontrado em relação a extensão proposta do projeto do ECOSAR foi que o cenário limitava a utilização de muitos animais, visto que era pequeno e aumentá-lo envolveria um problema maior já que está vinculado ao marcador utilizado para a cena. Além disso o formato do terreno dificultava a movimentação dos animais, resultando numa movimentação menos fluida. Estes problemas acabaram limitando a possibilidade de implementação de recursos como reprodução dos animais ou da criação de uma forma para adicionar e/ou remover animais do cenário, visto que com três animais o cenário já se mostrava cheio.

O trabalho se mostrou relevante pois como pode ser observado em relação aos trabalhos correlatos selecionados, os projetos anteriores procuravam desenvolver um modelo de inteligência artificial apenas aplicado aos seus respectivos trabalhos, sem a utilização de um editor de jogos consolidado. Com a pesquisa realizada e o desenvolvimento deste trabalho, pode-se aprender sobre a biblioteca Unity ML-Agents e compreender como utilizá-la em um projeto, facilitando trabalhos futuros e explicando esta tecnologia que é implementada utilizando um motor de jogos amplamente utilizado por alunos e desenvolvedores de jogos e simuladores.

4 CONCLUSÕES

Com a análise da execução dos modelos de comportamento treinados provou-se que os animais foram capazes de se comportar da maneira desejada e, portanto, tornaram o simulador de ecossistemas mais robusto, adicionando uma nova camada de estudo e aprendizado do meio ambiente. Além disso as tecnologias utilizadas para o desenvolvimento do trabalho mostraram-se apropriadas, sendo o Unity ML Agents uma ferramenta completa e eficiente, permitindo desenvolvimento de modelos de comportamentos complexos e desenvolvimento de múltiplos agentes com sucesso em uma cena do Unity.

Cumpriu-se o objetivo de pesquisar sobre tecnologias de inteligência artificial disponíveis para o Unity e posteriormente estudar e desenvolver uma solução utilizando a biblioteca Unity ML-Agents. Essa pesquisa toma disponível o conteúdo sobre essa biblioteca que está fortemente integrada ao motor de jogos Unity, amplamente utilizado em outros projetos acadêmicos e recebe atualizações constantemente, recebendo foco da equipe de desenvolvimento do Unity para se tomar uma solução cada vez mais robusta. Desta forma a pesquisa deste artigo se prova um eficiente método de aprendizado desta ferramenta. for Hestado >

As possíveis extensões propostas para continuar a linha de pesquisa desse projeto são: 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.

REFERÊNCIAS

ALDRICH, Clark. Learning online with games, simulations and virtual worlds. San Francisco, CA: Jossey-Bass, 2009.

DAUTENHAHN, Kerstin, NEHANIV, Chrystopher L. Imitation in Animals and Artifacts. Cambridge, MA; The MIT Press, 2002.

EUROGRAPHICS/ACM SIGGRAPH SYMPOSIUM ON COMPUTER ANIMATION, x., 2005, Los Angeles. Proceedings... New York: Association for Computing Machinery, 2005. 274 p.

FELTRIN, Gustavo R. VISEDU-SIMULA 1.0: Visualizador de material educacional, módulo de animação comportamental. 2014. 91f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FRONZA, Germano. Simulador de um ambiente virtual distribuído multiusuário para batalhas de tanques 3D com inteligência baseada em agentes BDI. 2008. 141f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) — Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

Trabalho de Conclusão de Curso - Ano/Semestre: 2020/1



APÊNDICE B – CONFIGURAÇÃO DE AMBIENTE PARA UTILIZAÇÃO DO UNITY ML-AGENTS

Este apêndice irá explicar como funciona a instalação e configuração do ambiente com o Anaconda para permitir a realização do treinamento dos agentes de cena utilizando o Unity. Meste realização do treinamento dos agentes de cena utilizando o Unity. Meste realizados o Unity ML-Agents na versão 0.13.1 e o Anaconda na versão 2019.10, utilizando a versão 3.7 do Python. Ambos podem ser baixados através dos sites oficiais dos distribuidores destes componentes.

Após a instalação do Anaconda você deve executar o programa Anaconda Prompt, onde irão ser executados os comandos para configuração do ambiente. Primeiramente é preciso criar um ambiente Python para a execução dos scripts de treinamento do ML-Agents. Para isso deve ser executada uma linha de comando que irá criar um ambiente Python na versão 3.7 chamado ml-agents. Após a criação do ambiente é necessária a execução de uma outra linha de comando para ativar o ambiente. Ambas as linhas de comando podem ser visualizadas na Figura 10.

Figura 10 - Comandos para criação do ambiente

conda create -n ml-agents python=3.7

Fonte: elaborado pelo autor.