

ECOSAR – SIMULADOR DE ECOSISTEMAS UTILIZANDO REALIDADE AUMENTADA

Rodrigo Wernke Pereira, Dalton Solano dos Reis – Orientador, Roberta Andressa Pereira –
Coorientadora

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil
rodrigowernke@furb.br, dalton@furb.br, rapereira@furb.br

Resumo: Este artigo descreve o processo de desenvolvimento e testes de um aplicativo que tem como objetivo simular ecossistemas utilizando as tecnologias de Realidade Aumentada e Interfaces de Usuário Tangível. A atividade de simular ecossistemas utilizando estas tecnologias possibilita ao usuário a observação e controle de fenômenos naturais como o ciclo da água e o ciclo de vida das árvores. O aplicativo foi desenvolvido utilizando o motor gráfico Unity em parceria com a biblioteca Vuforia. Para verificar o desempenho do aplicativo, foi realizado testes com uma professora e acadêmicos do curso de ciências biológicas. Após realizadas estas validações, o aplicativo se mostrou capaz de cumprir seu objetivo.

Palavras-chave: Simulador. Ecossistemas. Unity. Vuforia. Biologia. Ciclo.

1 INTRODUÇÃO

Atualmente, com o advento da tecnologia, existem várias formas de ensinar sobre o meio ambiente. Ecossistema, por exemplo, é apresentado no ensino fundamental.

A palavra ecossistema refere-se a um conjunto de organismos vivos que interagem não só com o meio físico que os rodeia, mas também com a química ambiental e com o meio social e biológico em que estão inseridos (...) (CARAPETO, 2016, p. 15).

O uso da tecnologia cria transformações, incluindo a área de educação, pois ela promove novas formas de adquirir e transmitir o conhecimento. Desta forma, faz-se necessário pensar as diversas maneiras de usar o potencial que a tecnologia oferece para auxiliar no ensino e aprendizagem dos conteúdos, em uma era onde ela se encontra cada vez mais presente na vida das pessoas.

Dentro do espaço de tecnologias disponíveis para serem usadas na educação, pode-se citar o uso de simuladores, que imitam e reproduzem situações reais ou mesmo propostas de forma abstrata dos fenômenos simulados (GREIS; REATEGUI; 2010, p. 3). As vantagens de trabalhar com fenômenos simulados por computador na área educacional são muitas, desde a oportunidade de tornar possível a reprodução de processos lentos ou perigosos para serem produzidos no mundo real, controlar as etapas de observação destes fenômenos e até mesmo a redução de gastos envolvidos no projeto (GREIS; REATEGUI; 2010, p. 3).

Uma das áreas que vem auxiliando o desenvolvimento de simuladores é a Realidade Aumentada (RA). RA é definida por Kirner et al. (2006) como uma técnica para conectar o ambiente virtual ao ambiente real do usuário, proporcionando uma interação natural, sem necessidade de treinamento ou adaptação.

Com o uso da Realidade Aumentada, foi possível o surgimento de novos tipos de interfaces. Ullmer e Ishii (2001) descrevem Interfaces de Usuário Tangíveis (IUT) como representações físicas para dados digitais, permitindo que objetos físicos sejam controles para o mundo virtual. Diferente de teclados e mouses que também são objetos físicos, as formas e posições físicas dos objetos de IUT são um importante papel no mundo virtual.

Diante deste contexto, este trabalho propõe o desenvolvimento de um aplicativo de simulação de ecossistemas, através da utilização da RA para visualização e o uso de IUT para o usuário manipular características simuladas da cena.

O trabalho proposto tem como objetivo desenvolver um simulador de ecossistemas para dispositivos móveis, no qual seja possível manipular elementos da natureza, como o vento e a temperatura. Os objetivos específicos são: disponibilizar um aplicativo que seja capaz de simular um ecossistema; permitir o controle da simulação com algum meio de interface tangível.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os aspectos da fundamentação teórica utilizados para a construção do aplicativo. Na seção 2.1 é comentado sobre a Realidade Aumentada e Interfaces de Usuário Tangível. Na seção 2.2 é apresentado sobre simuladores. A seção 2.3 apresenta ecossistemas e na seção 2.4 são apresentados três trabalhos correlatos com o trabalho relatado neste artigo.

2.1 REALIDADE AUMENTADA E INTERFACES DE USUÁRIO TANGÍVEL

Diferente da Realidade Virtual, onde o usuário é imerso em um ambiente criado digitalmente, a realidade aumentada combina o mundo real com o mundo virtual, onde estes dois coexistem alinhados e em tempo real (AZUMA, 2001 apud ROMÃO; GONÇALVES, 2013, p. 23). A realidade aumentada não cria mundos virtuais, mas maximiza elementos do mundo real para que se possa melhorar a interação e perceber aspectos sensoriais imperceptíveis nas dimensões reais (FRANÇA; SILVA, 2017, p. 3). De acordo com Kirner e Siscoutto (2007, p. 5), o fato dos objetos virtuais serem trazidos para o espaço físico do usuário por sobreposição, permitiu interações tangíveis mais fáceis e naturais, sem o uso de equipamentos especiais.

Enquanto a realidade virtual depende de equipamentos especiais para a visualização, como monitor ou capacete e normalmente é utilizada em ambientes fechados, a realidade aumentada não apresenta esta desvantagem, podendo ser usada em qualquer ambiente (KIRNER; SISCOUITTO, 2007, p. 5).

A Realidade Aumentada permite explorar a Interface de Usuário Tangível. Segundo Jetter (2013 apud REIS; GONÇALVES, 2016, p. 95) interfaces podem ser entendidas como uma camada de comunicação entre dois elementos: um usuário que emite comandos e um artefato ou sistema que responde a esses comandos, promovendo, assim, uma interação. Ullmer e Ishii (2001) definem que Interfaces de Usuário Tangíveis são representações físicas para dados digitais, permitindo que objetos físicos tenham controle no mundo virtual.

Diferente de teclados e mouses que também são objetos físicos, as formas e posições de interfaces tangíveis tem um papel importante para o mundo virtual (ULLMER; ISHII, 2001). De acordo com Reis e Gonçalves (2016, p. 97) o termo tangível foi usado para contrastar o aspecto intangível das interfaces gráficas, sendo que estas interfaces não são constituídas de objetos físicos diretamente manipuláveis pelo usuário.

2.2 SIMULADORES

Através da simulação é possível imitar ou reproduzir situações reais ou mesmo propostas de forma abstrata, dos fenômenos a serem simulados (GREIS; REATEGUI, 2010). Os experimentos que utilizam estas possibilidades buscam entender o comportamento ou avaliar estratégias para a sua operação, segundo Aldrich (2009). Segundo Greis e Reategui as vantagens de trabalhar com modelos simulados por computador no campo educacional são muitas. Desde a oportunidade de tornar possível a reprodução de processos muito lentos ou muito perigosos para serem reproduzidos no ambiente natural, para a observação dos fenômenos e até mesmo pela redução dos custos envolvidos no projeto.

De acordo com Torga (2007, p. 43) simulação é a importação da realidade para um ambiente controlado no qual se pode estudar o comportamento do mesmo, sob diversas condições, sem riscos físicos ou grandes custos envolvidos. A simulação é o ato de imitar um procedimento real em menor tempo e com um custo menor, possibilitando um melhor estudo do que vai acontecer e de como consertar erros que gerariam grandes gastos (O'KANE; TAYLOR; SPENCELEY, 2000).

Reunindo características que são encontradas em jogos e em simulações, utilizando tecnologias de realidade virtual, estes ambientes permitem a representação física do usuário em um espaço virtual recriado tridimensionalmente (GREIS; REATEGUI, 2010).

2.3 ECOSSISTEMAS

Ricklefs (2010, p. 4) afirma que ecossistemas são sistemas ecológicos complexos e grandes, as vezes incluindo muitos milhares de diferentes tipos de organismos, vivendo cada um em numa grande variedade de meios. Podemos pensar em um ecossistema como um organismo, que possui processos internos e troca com os arredores externos (RICKLEFS, 2010, p. 4).

Segundo Odum (1988, p.13) chama-se ecossistema qualquer unidade (biossistema) que abranja todos os organismos funcionando em conjunto, interagindo com o ambiente físico de modo que o fluxo de energia produza estruturas bióticas claramente definidas e uma ciclagem de materiais entre as partes vivas e não vivas. Os ecossistemas são caracterizados pela magnitude de processos ecológicos fundamentais, como a produção e a ciclagem de materiais, de acordo com Odum (1972 apud COELHO; MOTTA, 2009, p.70).

Fiedler et al. (1997 apud VIGLIO; FERREIRA, 2013, p. 3) relatam que ecossistemas são unidades autorreguladas que seguem uma trajetória linear de desenvolvimento em direção a uma particular diversidade biológica

e um estado de estabilidade denominado clímax. Com esta perspectiva, são considerados muito mais como eventos externos do que propriedades intrínsecas dos sistemas ecológicos. O paradigma de equilíbrio é uma das ideias mais antigas e dominantes na ecologia e foi fundamental no desenvolvimento do conceito de ecossistema segundo Pickett et al. (1992 apud VIGLIO; FERREIRA, 2013, p. 4).

2.4 TRABALHOS CORRELATOS

A seguir são apresentados trabalhos com características semelhantes aos principais objetivos do trabalho desenvolvido. O primeiro descreve o trabalho de Reiter (2018) que desenvolveu uma ferramenta de criação de animações em 3D através de uma combinação de Interface de Usuário Tangível e Realidade Aumentada (Quadro 1). O segundo descreve o trabalho de Piske (2015) que teve como objetivo desenvolver um aplicativo que simulasse um ecossistema de aquário marinho com animação comportamental (Quadro 2). O terceiro relata o aplicativo Weather (TINYBOP, 2016), que se trata de um simulador de características do clima para dispositivos móveis (Quadro 3).

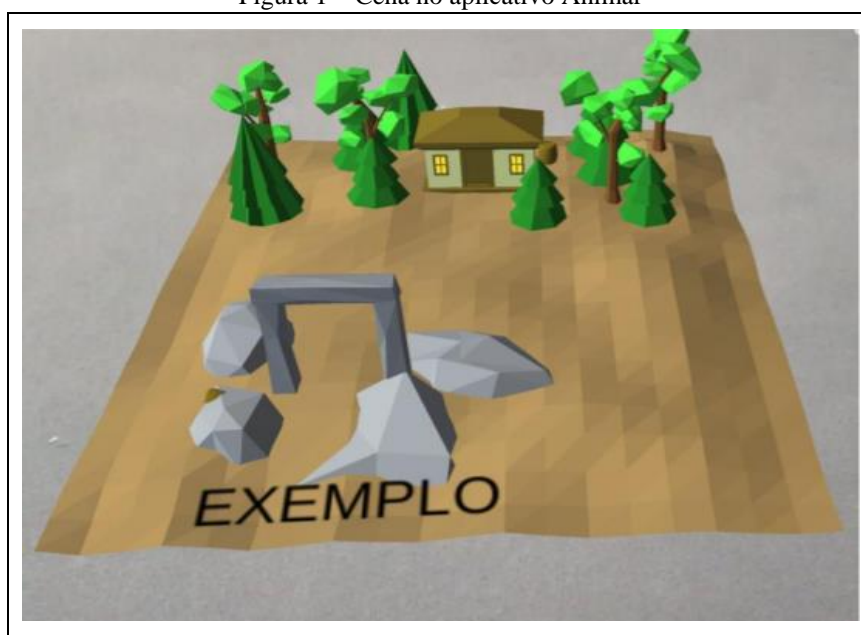
Quadro 1 - Animar: Desenvolvimento de uma Ferramenta para Criação de Animações com Realidade Aumentada e Interface Tangível

Referência	Reiter (2018).
Objetivos	Desenvolver uma ferramenta de criação de animações em 3D através de uma combinação de Interface de Usuário Tangível e Realidade Aumentada.
Principais funcionalidades	O software, batizado de Animar, é uma ferramenta de criação e manipulação de cenários e animações com objetos tridimensionais.
Ferramentas de desenvolvimento	Foram utilizadas as bibliotecas: <ul style="list-style-type: none"> a) Unity: motor gráfica utilizada na ferramenta; b) Vuforia: responsável pelos processos de realidade aumentada; c) AR Marker Generator e Adobe Photoshop CS6: responsáveis por gerar as imagens dos marcadores.
Resultados e conclusões	Segundo Reiter (2018, p. 73), os resultados foram satisfatórios, entretanto, observou-se uma certa dificuldade no começo do uso da aplicação, pois a maioria dos alunos nunca tinham utilizado aplicações com Realidade Aumentada. O objetivo de disponibilizar uma nova maneira de se criar animações foi atendido, mas o objetivo de disponibilizar o uso da aplicação com Cardboard não foi satisfatório com os equipamentos utilizados.

Fonte: elaborado pelo autor.

O trabalho de Reiter (2018) também permite a utilização de um *head-mounted display*, ou Cardboard, mas não foram realizados testes com uma quantidade significativa de pessoas, pois o uso da ferramenta não se mostrou eficiente (REITER, 2018, p.74). Parte da proposta do aplicativo apresentado neste artigo consiste em utilizar Realidade Aumentada e Interface de Usuário Tangível, o que representa parte da pesquisa realizada por Reiter (2018). A Figura 1 mostra uma cena de exemplo no aplicativo Animar.

Figura 1 – Cena no aplicativo Animar



Fonte: Reiter (2018).

Quadro 2 – VisEdu – Aquário Virtual: Simulador de Ecossistemas Utilizando Animação Comportamental

Referência	Piske (2015).
Objetivos	Desenvolver um simulador de ecossistema de aquário marinho e permitir a inserção de agentes dotados de representações gráficas.
Principais funcionalidades	O software VISEDU simula uma cadeia alimentar pequena, na qual o tubarão é o predador e a sardinha como presa que se alimenta de plânctons.
Ferramentas de desenvolvimento	Foram utilizadas as ferramentas: <ul style="list-style-type: none"> a) linguagem de programação Javascript; b) o elemento canvas do HTML5; c) a biblioteca gráfica ThreeJS; d) interpretador Jason para o desenvolvimento de agentes sob o modelo BDI, utilizando a linguagem AgentSpeak.
Resultados e conclusões	Piske (2015, p. 99) expõe em suas conclusões que o trabalho obteve sucesso em criar um aquário virtual que simulasse um ecossistema marinho através de animação comportamental. O objetivo de permitir inserções de agentes dotados de representações gráficas também foi contemplado.

Fonte: elaborado pelo autor.

O Aquário Virtual de Piske (2015) possui o objetivo de incluir funcionalidades para permitir a geração de animações comportamentais, que também foi alcançado. Assim como o Aquário Virtual, o trabalho proposto utiliza animações comportamentais para gerar o comportamento dos objetos da cena, porém o Aquário Virtual não utiliza Realidade Aumentada para a visualização da simulação. Outro ponto que difere os trabalhos é a plataforma para a qual ele foi desenvolvido. O Aquário Virtual é um software que é executado em um navegador, já o trabalho realizado é executado em dispositivos móveis. A Figura 2 mostra o aquário virtual de Piske (2015).

Figura 2 – Aquário virtual no aplicativo VISEDU



Fonte: Piske (2015).

Quadro 3 - Weather

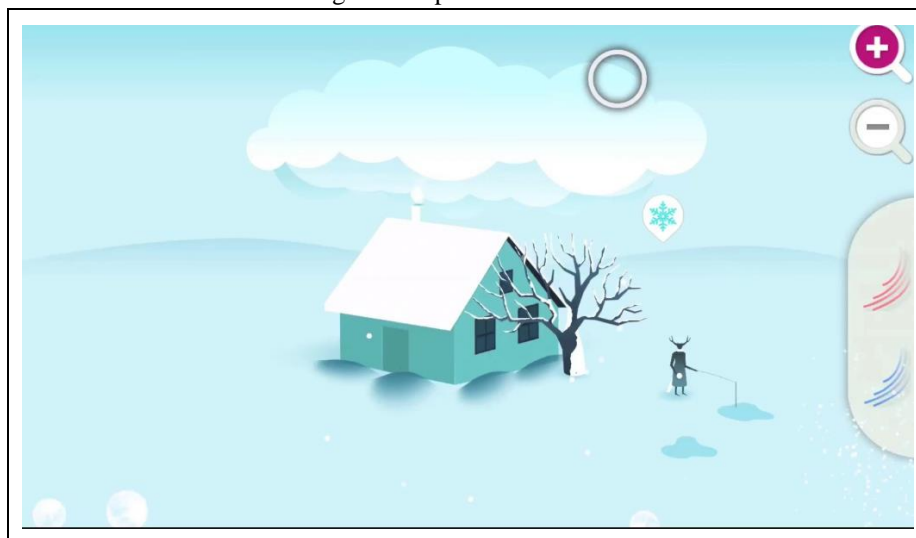
Referência	Tinybop (2016).
Objetivos	Weather é um aplicativo interativo, baseado em ciência e engenharia, que foca na educação infantil e trata sobre a natureza.
Principais funcionalidades	Simular sistemas climáticos e interativos de forma educacional.
Ferramentas de desenvolvimento	Não encontrado.
Resultados e conclusões	Com cerca de 86 avaliações na Play Store, o aplicativo mantém sua nota na loja de aplicativos em 3.8 (escala máxima 5.0). Na App Store o aplicativo possui a nota 4.2 (escala máxima 5.0) com 24 avaliações, o que demonstra um bom nível de satisfação do público que o utiliza.

Fonte: elaborado pelo autor.

Weather utiliza simulações de elementos da natureza para criar um jeito diferente de ensinar sobre o meio ambiente. Assim como no trabalho proposto, o aplicativo Weather possibilita a manipulação de elementos da natureza,

e apresenta os comportamentos gerados por estas manipulações, mas como o Aquário Virtual, ele não utiliza Realidade Aumentada como meio de visualização da cena. A Figura 3 mostra uma das cenas do aplicativo Weather, na qual é possível controlar a precipitação que ocorre nas nuvens.

Figura 3- Aplicativo Weather



Fonte: Tinybop (2016).

3 DESCRIÇÃO DO APLICATIVO

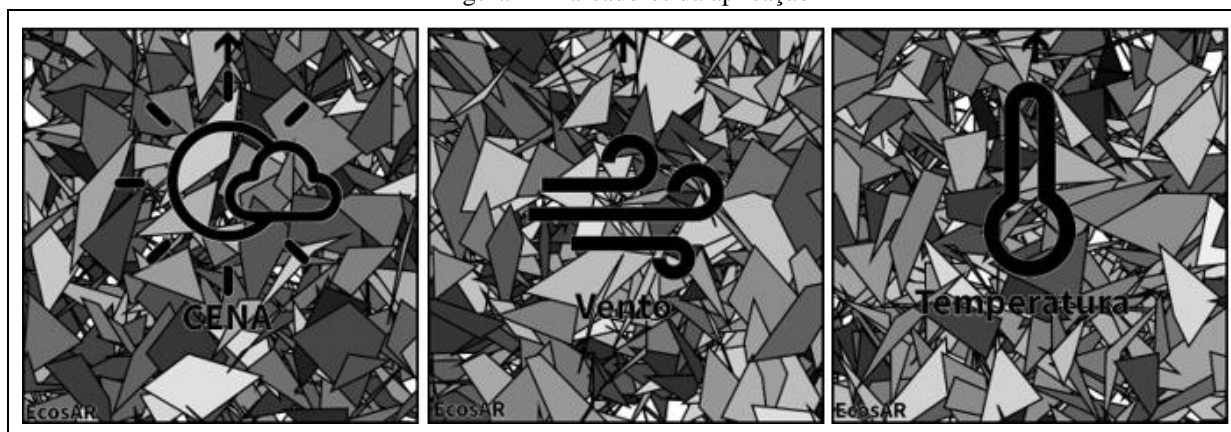
Este capítulo pretende apresentar os detalhes de especificação e implementação do aplicativo. Para tanto, são apresentadas três seções. A primeira seção apresenta a visão geral do aplicativo, mostrando ao leitor o funcionamento da simulação que é proporcionada pelo aplicativo e as ações que o usuário pode realizar para alterar os comportamentos realizados na simulação. A segunda seção apresenta a arquitetura do aplicativo, mostrando a implementação do mesmo (desenvolvido em C#), responsável por controlar a simulação. A terceira seção apresenta características do uso da aplicação, detalhando estados e comportamentos realizados na simulação.

3.1 VISÃO GERAL DO APLICATIVO

O aplicativo desenvolvido disponibiliza para o usuário uma forma de simular comportamentos de um ecossistema, concedendo ao usuário 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 para visualizar a aplicação.

Estes marcadores, conforme apresentado na Figura 4, são formados por imagens que são reconhecidas e interpretadas pelo aplicativo, fazendo possível o desenho da cena sobre o marcador cena. Desta forma, o usuário é capaz de interagir com a aplicação utilizando Realidade Aumentada e Interfaces de usuário tangível, no qual o mesmo irá rotacionar os marcadores Vento e Temperatura para controlar os elementos que são visualizados sobre o marcador Cena.

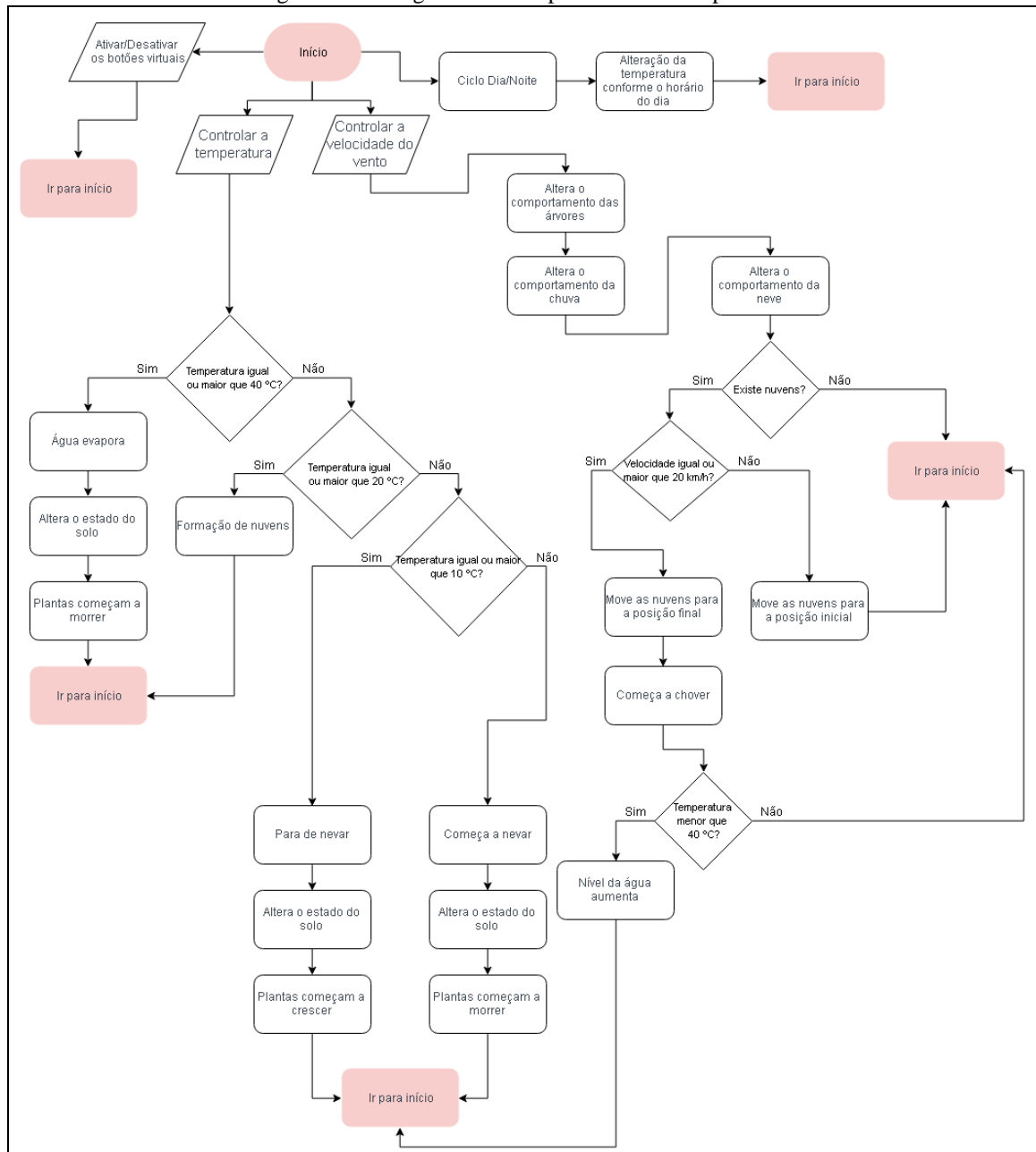
Figura 4- Marcadores da aplicação



Fonte: elaborado pelo autor.

Rotacionando estes marcadores, diversos comportamentos podem ser simulados. A Figura 5 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.

Figura 5 - Fluxograma de comportamentos do aplicativo



Fonte: elaborado pelo autor.

Como pode ser visto na Figura 5, dependendo das alterações do usuário, a simulação realiza diferentes comportamentos para diferentes alterações, sendo que mais de um comportamento pode ocorrer ao mesmo tempo, assim gerando um ambiente em que várias simulações ocorrem simultaneamente. Fora do controle do usuário, ocorre o ciclo de dia e noite, fazendo com que a temperatura da simulação seja alterada conforme o horário do dia, sendo que durante o dia a simulação aumenta quatro graus e durante a noite a temperatura diminui quatro graus. Quando o usuário altera a temperatura com o uso do marcador, o valor de temperatura calculado pelo ciclo é ignorado e o valor escolhido pelo usuário é utilizado.

3.2 IMPLEMENTAÇÃO

Para o desenvolvimento do aplicativo, foram utilizados o motor gráfico Unity, em conjunto com a biblioteca Vuforia e o ambiente de desenvolvimento Visual Studio 2017. Para a geração dos marcadores foram utilizadas as ferramentas Photoshop CC 2019 e AR Marker Generator (BROSVISION, 2019), que gera imagens aleatórias e

otimizadas para a interpretação de marcadores com Realidade Aumentada. Para a modelagem dos objetos foi utilizado a ferramenta Blender.

Em uma aplicação desenvolvida com o motor gráfico Unity, é utilizado o conceito de `scene`, que é a cena da aplicação em si, como por exemplo a cena de um menu ou a cena do nível de um jogo (REITER, 2018, p.42). Com isso, a aplicação é dividida em duas cenas, a cena menu e a cena principal do aplicativo, a cena `Main`. Dentro dessas `scenes` é utilizado também o conceito de `GameObjects` (GO) que são objetos fundamentais na Unity, que podem representar objetos gráficos e componentes necessários na `scene` (UNITY, 2019, tradução nossa). Com estes conceitos compreendidos, a Figura 6 apresenta a árvore hierárquica dos `GameObjects` utilizados na cena `Main` do aplicativo. Em seguida o Quadro 4 apresenta a descrição de cada componente da árvore hierárquica.

Figura 6 - Árvore hierárquica da cena `Main`



Fonte: elaborado pelo autor.

Quadro 4 - Componentes da árvore hierárquica

Identificador	Descrição
1	Este componente é a câmera da aplicação, ela faz parte da biblioteca Vuforia e faz com que a aplicação reconheça os marcadores.
2	Componente responsável por processar e manipular eventos em uma cena da Unity.
3	Este componente é responsável pelo script que gerencia o acelerômetro do dispositivo móvel.
4	Neste componente é processado a lógica da simulação no script <code>SimulationController</code> .
5	Este componente é o marcador <code>Cena</code> .
6	Este componente é o marcador responsável por controlar a velocidade do vento.
7	Este componente é o marcador responsável por controlar a temperatura.
8	Componente responsável por armazenar os botões da interface virtual.

Fonte: elaborado pelo autor.

O principal `GameObject` na cena `Main` é o `Simulation Controller`. Nele o script `SimulationController` é anexado, no qual grande parte do controle lógico da cena é computado. O método `Update` é o mais importante deste script, pois ele é responsável por atualizar a lógica da simulação e é computado em todos os frames da simulação.

O `GameObject Canvas` é responsável por armazenar o botão de ajuda e todos os textos de ajuda da simulação. Estes textos de ajuda são mostrados ao usuário quando o mesmo utiliza os botões virtuais que são ativados após o botão de ajuda ser selecionado. Os botões virtuais fornecem aos marcadores um mecanismo útil para torná-los interativos (VUFORIA, 2019, tradução nossa).

A aplicação também faz uso do acelerômetro do dispositivo móvel, utilizando o mesmo para reiniciar a simulação para o estado inicial. Com isto, o usuário pode balançar o dispositivo a qualquer momento para reiniciar a cena. O script responsável por este controle se encontra no `GameObject Scene Manager`.

A aplicação é dividida em vários scripts chamados de `Controllers`. Cada `Controller` é responsável por controlar a lógica de algum elemento específico da cena. Como por exemplo, o `Controller SnowController` é

encarregado de controlar o comportamento da neve na simulação, assim como o `TerrainController` é responsável por controlar o comportamento do terreno na cena. Todos estes `Controllers` possuem o método `Update`, que é chamado pelo `SimulationController` em todo frame, para atualizar a lógica destes elementos. O diagrama de classes do aplicativo está disponibilizado no APÊNDICE A.

O `TerrainController` necessita ter conhecimento da temperatura da cena para ser capaz de controlar o terreno de forma adequada para aquele momento da simulação. Logo, o `TemperatureController` deve ser executado antes que o código do `TerrainController` execute. Com isso, o Quadro 5 mostra a ordem de chamada utilizada no `SimulationController` para atualizar a cena corretamente.

Quadro 5 – Método `Update` no script `SimulationController`

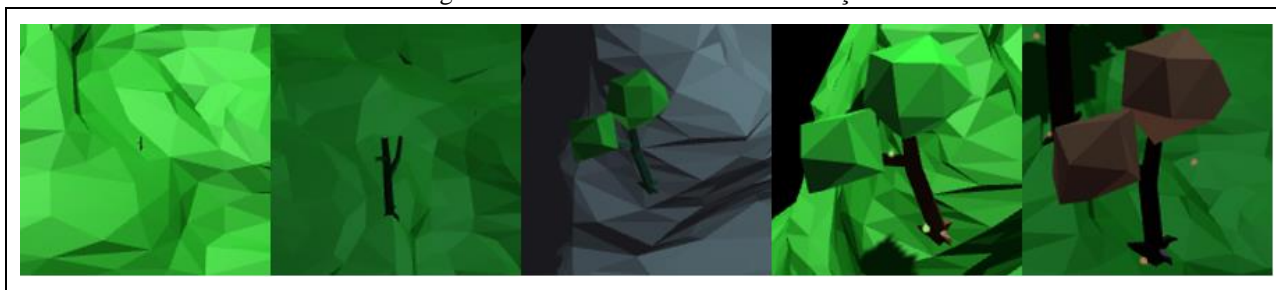
```
void Update()
{
    UpdateSceneState();

    _windController.Update();
    _temperatureController.Update();
    _snowController.Update(_temperatureController.Temperature);
    _cloudController.Update(_windController.WindForce, _temperatureController.Temperature);
    _rainController.Update(_cloudController.IsOnRainingPosition);
    _waterController.Update(_temperatureController.Temperature, _rainController.Raining, _cloudController.IsOnRainingPosition);
    _terrainController.Update(_currentSceneState, _temperatureController.Temperature);
    _treeGrowthStateController.Update(_currentSceneState);
}
```

Fonte: elaborado pelo autor.

Para realizar a simulação do ciclo de vida das árvores, é utilizado o script `TreeGrowthStateController`, no qual é executado o método `UpdateTreeGrowthState`. Este método é executado durante toda a duração da simulação e é encarregado de interpretar o estado da simulação e alterar o estado de crescimentos das árvores na cena. A simulação possui dois estados possíveis, o estado favorável e o estado desfavorável, ambos os estados são obtidos através de alterações que o usuário faz no marcador de temperatura anteriormente citado. O estado favorável ocorre enquanto a simulação estiver com a temperatura entre onze e trinta e nove graus e o estado desfavorável ocorre quando a temperatura estiver abaixo de onze graus e acima de trinta e nove. A Figura 7 mostra os estados possíveis que as árvores na simulação podem estar.

Figura 7 - Estado das árvores na simulação



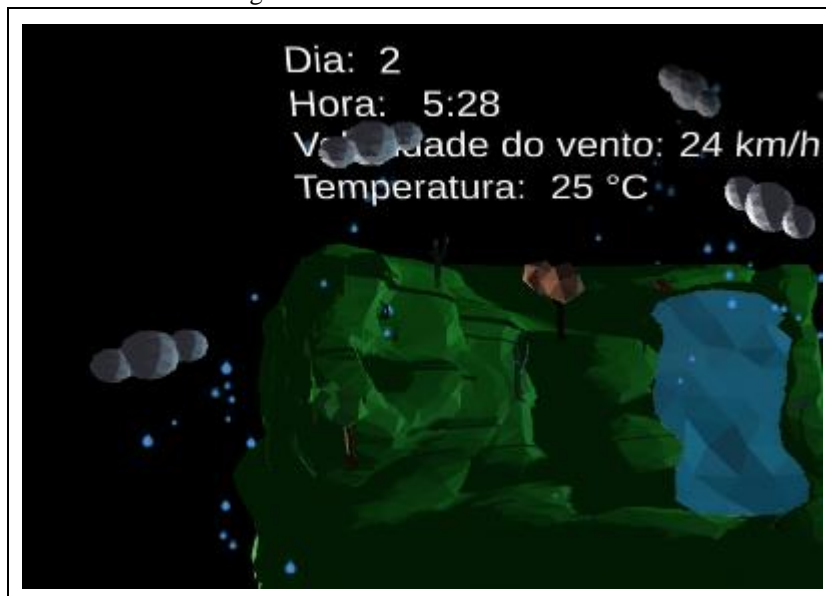
Fonte: elaborado pelo autor.

Estes estágios foram divididos de uma forma no qual o usuário possa diferenciar visualmente que o ciclo de vida das árvores esteja acontecendo, fazendo com que haja um retorno visual em relação às alterações que o usuário faz com os marcadores.

Para ocorrer a simulação de nuvens, o script `CloudController` é responsável por controlar o estado das nuvens na cena. Ele verifica a cada frame da simulação se a temperatura para a formação de nuvens é adequada, e também verifica se a velocidade do vento é correta para ocorrer a movimentação das mesmas. Os valores usados para a velocidade do vento foram baseados na escala de Beaufort (ROYAL METEOROLOGICAL SOCIETY, 2019) que pode ser visualizada no ANEXO A.

Alterando a velocidade do vento, o usuário tem a opção de controlar duas posições para as nuvens, a inicial e a final, caso o usuário posicione as nuvens na posição final, o `Controller` responsável pela chuva `RainController`, fará com que ocorra chuva e, caso o nível do lago esteja abaixo do normal, fará com que o nível retorne para a posição original. A Figura 8 mostra a cena com nuvens e chuva.

Figura 8 - Cena com nuvens e chuva



Fonte: elaborado pelo autor.

Durante a simulação, ocorre o ciclo de dia e noite. Nele é simulado o período de um dia real com a duração de trinta segundos por dia. Para ocorrer um retorno visual sobre o ciclo, é utilizado dois `GameObjects`, um que representa o sol e outro que representa a lua. Estes `GameObjects` controlam a iluminação da cena, orbitando o terreno da simulação. Em relação à posição no qual eles se encontram é calculado o valor da hora e minuto do dia. Esta rotação é o resultado de um cálculo realizado a cada frame da simulação, no qual é calculado o ângulo no qual o objeto deve ser rotacionado. Com o ângulo calculado é possível obter um valor entre zero e um que corresponde a porcentagem no qual este ângulo calculado está em relação a trezentos e sessenta graus, que seria uma volta completa, ou um dia. Este valor é utilizado para calcular a hora do dia simulado, convertendo este valor que varia de zero e um para a variação de zero e vinte e quatro é possível descobrir a relação entre ângulo e hora.

O Quadro 6 mostra o método `TimeOfDay` que calcula a hora da simulação. Com a hora do dia calculada, é possível deduzir se é noite ou dia na simulação, possibilitando que o `TemperatureController` altere a temperatura de acordo com a hora do dia.

Quadro 6 – Método responsável por calcular a hora da simulação

```
private void TimeOfDay()
{
    double decimalTime = ConvertRange(0, 1, 0, 24, _rotationPercentage);

    _hour = (int)(decimalTime);

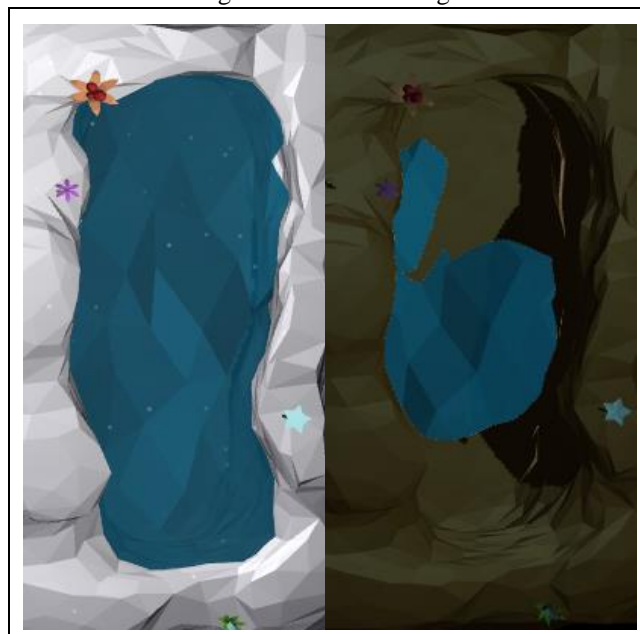
    _minute = (int)((decimalTime - Math.Truncate(decimalTime)) * 60);

    if ((_hour >= 18) || (_hour < 6))
    {
        IsNight = true;
        IsDay = false;
    }
    else
    {
        IsNight = false;
        IsDay = true;
    }
}
```

Fonte: elaborado pelo autor.

O `WaterController` é responsável pela água na simulação, durante todos os frames da simulação, ele verifica a temperatura, se as nuvens estão na posição correta e se está chovendo. Para o nível do lago diminuir é necessário que a simulação esteja com a temperatura acima de 40 graus e que não esteja chovendo, com isso o `WaterController` invoca o método `Evaporate` no componente `Water` que fica no `GameObject` do lago, fazendo com que o nível do lago abaixe. Para o nível do lago voltar ao normal, a simulação deve estar com a temperatura menor que 40 graus, também é necessário que as nuvens estejam na posição correta e que esteja chovendo, com isso o `WaterController` chama o método `Condense` no componente `Water` que fica no `GameObject` do lago. A Figura 9 compara o lago cheio de água e com ele secando.

Figura 9 - Níveis do lago



Fonte: elaborado pelo autor.

Para simular a neve, chuva, as folhas caindo das árvores e a evaporação da água, foi utilizado o `ParticleSystem` disponibilizado pela Unity. Com ele foi possível realizar simulações que exigissem várias partículas com facilidade. Além do `ParticleSystem` da Unity foram utilizados diversos `assets` que foram adaptados para seu uso na simulação. O Quadro 7 mostra os `assets` que foram utilizados no desenvolvimento do aplicativo. Outros objetos como o sol, a lua e o terreno foram desenvolvidos pelo autor utilizando a ferramenta Blender.

Quadro 7 - Assets utilizados

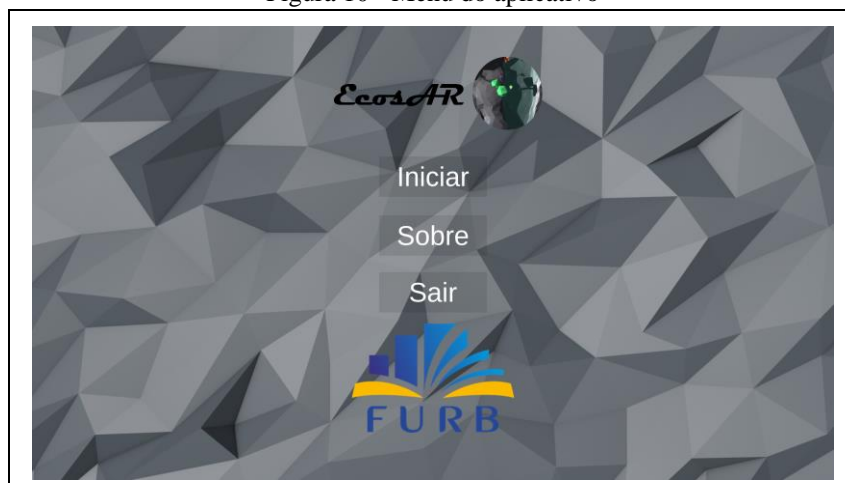
Asset	Utilização
Low Poly Water GPU	Forneceu a modelagem da água na simulação.
White Smoke Particle System	Utilizado na simulação da evaporação da água.
Low Poly Foliage	Forneceu a modelagem de flores na simulação.
LMH Poly	Forneceu texturas para as folhas das árvores.
Polygon Wind	Shader utilizado no movimento das árvores e nuvens.

Fonte: elaborado pelo autor.

3.3 A APLICAÇÃO

A aplicação se divide em duas telas principais, a tela menu inicial e a tela principal onde ocorre a simulação. A tela menu inicial, como pode ser vista na Figura 10, é composta por três botões que servem para iniciar a simulação, conhecer sobre a origem do aplicativo e sair da aplicação.

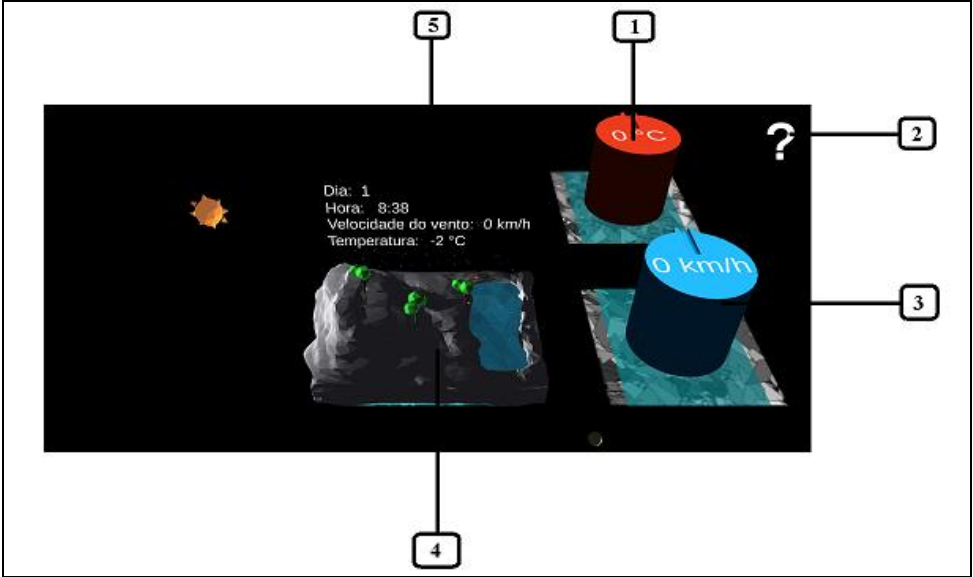
Figura 10 - Menu do aplicativo



Fonte: elaborado pelo autor.

Ao clicar no botão *Iniciar*, é carregada a simulação que o aplicativo utiliza, mas antes o usuário deve mirar a câmera do dispositivo móvel em direção ao marcador *cena*, com isso será possível visualizar a simulação. 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. Como citado na seção 3.1, dependendo da alteração do usuário, diferentes propriedades da simulação podem mudar. A Figura 11 mostra a simulação ao ser iniciada. Em seguida, o Quadro 8 apresenta a descrição de cada componente disponível na visualização do usuário.

Figura 11 - Início da simulação



Fonte: elaborado pelo autor.

Quadro 8 - Descrição dos componentes da tela inicial da simulação

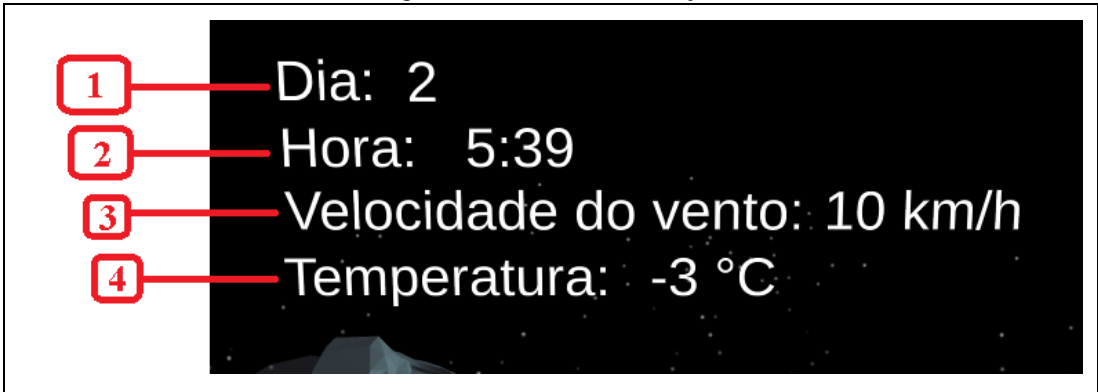
Identificador	Descrição
1	Marcador responsável pelo controle da temperatura.
2	Botão de ajuda.
3	Marcador responsável pelo controle da velocidade do vento.
4	Marcador responsável pela visualização da cena.
5	Painel com as características da cena.

Fonte: elaborado pelo autor.

O item 1 e o item 3 são os marcadores de controle do aplicativo, com os quais o usuário manipula a cena. O item 2 trata-se do botão de ajuda. Quando ativo, o aplicativo mostra a *bounding box* dos marcadores e ativa os botões virtuais. Estes botões são utilizáveis através da Realidade Aumentada para mostrar textos de ajuda para cada marcador. O item 4 é o marcador responsável pela visualização da simulação e juntamente o item 5 que mostra um painel com as características atualizadas da simulação.

Durante a simulação, o usuário pode observar um painel com as características atuais da cena (Figura 12). Nele é possível observar em qual dia a simulação se encontra (item 1). A hora atual da simulação (item 2). A velocidade do vento na simulação (item 3) e a temperatura da simulação (item 4).

Figura 12 – Painel da simulação



Fonte: elaborado pelo autor.

4 ANÁLISE DOS RESULTADOS

Este capítulo apresenta os testes realizados com o aplicativo. Serão apresentadas três seções para abordar os resultados, uma tratando dos testes de funcionalidade, outra relatando sobre os testes de utilização por uma profissional da área de biologia e outra abordando os testes com acadêmicos do curso de ciências biológicas.

4.1 TESTES DE FUNCIONALIDADE

Para validar as funcionalidades do aplicativo foram realizados testes na plataforma Android. Nesta plataforma, todas as funcionalidades se comportaram conforme o esperado. Durante o processo de desenvolvimento do trabalho, diversos testes foram executados para validar o comportamento de vários componentes da cena. Estes testes levaram em consideração a visualização e as alterações que o usuário fazia com os marcadores.

No decorrer do desenvolvimento do trabalho foi notado um problema de sensibilidade nos marcadores, no qual a mesma era muita alta, dificultando a usabilidade do aplicativo. Com isso, foi adicionado um algoritmo para diminuir a sensibilidade dos marcadores e facilitar o seu uso. Outra melhoria desenvolvida trata-se dos textos de ajuda ao usuário. Foi percebido que seria mais fácil a visualização dos mesmo sem a utilização de Realidade Aumentada, pois os usuários já estariam acostumados com o estilo.

Também foi observado durante os testes, que o material utilizado na impressão dos marcadores poderia afetar o uso da aplicação, sendo que materiais que refletem a luz com mais intensidade podem dificultar o reconhecimento dos marcadores pelo aplicativo. Outro fator importante é a qualidade da câmera do dispositivo móvel, que também pode afetar a identificação dos marcadores. Durante a utilização do aplicativo, foi notado o alto consumo da bateria do dispositivo móvel, devido ao fato do aplicativo utilizar a câmera constantemente e ser uma aplicação gráfica, consequentemente aumentando o consumo de energia.

4.2 TESTES DE UTILIZAÇÃO POR ESPECIALISTA

Durante o desenvolvimento do trabalho, no mês de maio de 2019, foi realizado uma entrevista com a professora Roberta Andressa Pereira (PEREIRA, 2019), professora do curso de Licenciatura em Ciências Biológicas da Universidade Regional de Blumenau. Nesta entrevista diversas melhorias foram discutidas e posteriormente implementadas no aplicativo, entre elas foi mencionado em deixar a noite mais escura e oscilar a temperatura em relação ao horário da simulação. Outra sugestão que também foi implementada tratou-se da divisão da cena em condições que seriam favoráveis ou não favoráveis ao crescimento das plantas, levando em consideração a temperatura atual. Também foi discutido sobre alteração no sistema de crescimento das árvores, devido ao fato de que até o momento elas cresciam simultaneamente. Com isso foi adicionado um algoritmo para alterar o ritmo de crescimento e aleatoriedade na geração das árvores, fazendo com que elas cresçam de forma diferenciada.

Pereira (2019) comentou que o aplicativo seria uma ótima ferramenta para auxílio no ensino e que o emprego deste aplicativo, além de incentivar a utilização de tecnologias em ambientes de ensino, é uma ótima forma de aumentar o interesse e participação dos alunos.

4.3 TESTES DE UTILIZAÇÃO COM ACADÊMICOS DO CURSO DE CIÊNCIAS BIOLÓGICAS

No mês de junho de 2019 foram realizados testes no laboratório S-226 do campus 1 da Universidade Regional de Blumenau, nos quais testes estavam presentes oito alunos do curso de Ciências Biológicas da instituição. Também estavam presentes o autor, orientador e coorientadora do trabalho, além da professora tutora (professora Simone Wagner). Fotos dos testes realizados podem ser visualizadas no APÊNDICE B.

Foram distribuídos os marcadores do trabalho desenvolvido para os alunos junto com um questionário online para testar a aplicação. A primeira etapa do questionário consiste em recolher informações sobre o perfil dos usuários, cujas respostas estão disponibilizadas na Tabela 1.

Tabela 1 - Perfil dos entrevistados

Sexo	Feminino	77,8%
	Masculino	22,2%
Idade	18	11,1%
	19	11,1%
	20	44,4%
	21	11,1%
	22	11,1%
	55	11,1%
Grau de escolaridade	Ensino superior incompleto	88,9%
	Ensino superior completo	11,1%
Utiliza dispositivos móveis com frequência	Frequentemente	100%
Já utilizou aplicações com Realidade Aumentada	Sim	55,6%
	Não	44,4%

Fonte: elaborado pelo autor.

Como pode ser observado, a maior parte dos alunos possuíam entre 18 e 22 anos, sendo a maioria do sexo feminino. Todos utilizavam dispositivos móveis com frequência e mais da metade dos entrevistados já utilizou uma aplicação com Realidade Aumentada.

A segunda etapa do questionário trazia o passo a passo que demonstrava todas as funcionalidades básicas do aplicativo, como o controle de temperatura e velocidade do vento, os quais todos os entrevistados conseguiram concluir. A terceira parte do questionário foi utilizada para obter a opinião dos entrevistados em relação a usabilidade de manipulação da cena do aplicativo, a usabilidade em geral do aplicativo, e se o aplicativo cumpriu seu objetivo. A Tabela 2 mostra os resultados da terceira parte do questionário, sendo as respostas destas perguntas valores entre um e cinco.

Tabela 2 – Opinião dos entrevistados sobre o aplicativo

Usabilidade de manipular a cena	3	11,1%
	4	33,3%
	5	55,6%
Usabilidade do aplicativo em geral	3	11,1%
	5	88,9%
Cumprir seu objetivo de desenvolver um simulador de ecossistemas para dispositivos móveis	2	11,1%
	4	11,1%
	5	77,8%

Fonte: elaborado pelo autor.

Em geral os resultados são considerados satisfatórios, visto que a grande maioria dos entrevistados demonstrou interesse pelo aplicativo e pelo seu meio de manipulação e controle da simulação. No final do questionário foram adicionadas críticas e sugestões sobre a aplicação e podem se tornar propostas para a continuação do trabalho.

5 CONCLUSÕES

Diante dos resultados apresentados o aplicativo se provou capaz de simular ecossistemas com dispositivos móveis utilizando Realidade Aumentada e Interface de Usuário Tangível. O aplicativo desenvolvido mostrou-se intuitivo e responsivo, com suas funcionalidades operando conforme o esperado. O objetivo de desenvolver um simulador de ecossistemas para dispositivos móveis foi atingido e comprovado com a realização dos testes. Mesmo sendo realizados com um grupo pequeno de alunos, foi possível obter resultados satisfatórios. No teste realizado com a especialista na área, Pereira (2019) relatou como o aplicativo poderia auxiliar o ensino de conceitos como ciclo e ecossistemas. Já no teste realizado com os bolsistas de ciências biológicas, todos ficaram interessados em utilizar o aplicativo e o avaliaram de acordo com as suas experiências de uso.

As tecnologias utilizadas para o desenvolvimento do aplicativo mostraram-se apropriadas. O motor gráfico Unity mostrou-se eficiente e prático em sua utilização, facilitando o desenvolvimento do aplicativo. A biblioteca Vuforia, que fez a integração da Realidade Aumentada na aplicação, foi competente na utilização da câmera e reconhecimento dos marcadores utilizados. A aplicação AR Marker Generator (BROSVISION, 2019) mostrou-se eficaz no auxílio da criação dos marcadores utilizado pelo EcosAR, gerando imagens com certa aleatoriedade, sendo elas muito eficiente para o reconhecimento pelo Vuforia.

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

- a) adicionar animais na simulação;
- b) incrementar a quantidade de terrenos possíveis;
- c) simular outras características do clima;
- d) aumentar a variedade de plantas na simulação;
- e) incluir novos meios de controle com IUT;
- f) implementar controles para a duração do dia.

REFERÊNCIAS

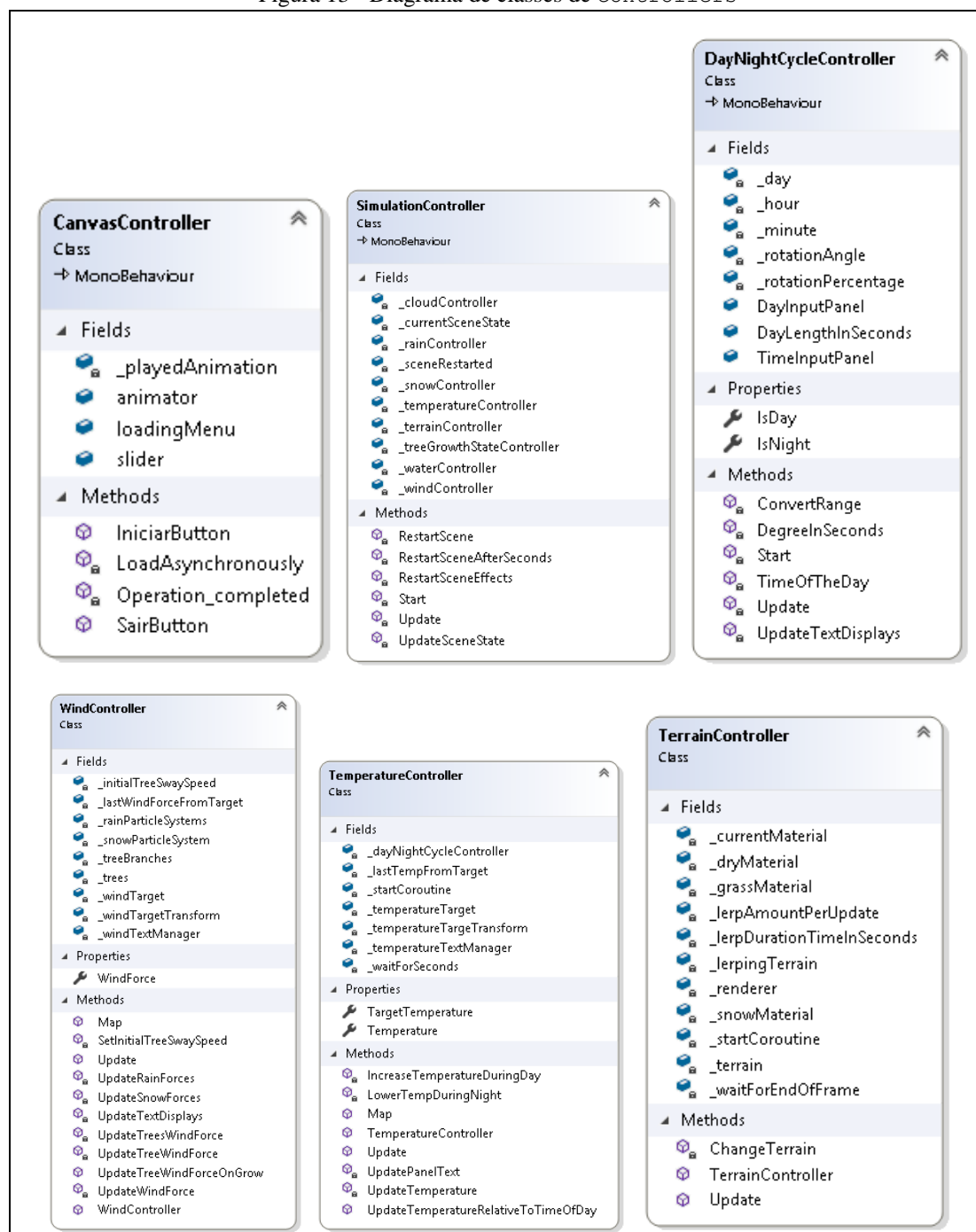
- ALDRICH, Clark. **The complete guide to simulations and serious games**. San Francisco: Pfeiffer, 2009. 576 p.
- BROSVISION. **Augmented Reality Marker Generator**. Disponível em: <<http://www.brosvision.com/ar-marker-generator/>>. Acesso em: 1 jun.2019.
- CARAPETO, Cristina. **Ecosistemas de Transição**. São Paulo: Leya, 2016. 128 p.
- COELHO, Pinto; MOTTA, Ricardo. **Fundamentos em ecologia**. [s. L.]: Artmed Editora, 2009. 252 p.
- FRANÇA, Carlos R.; SILVA, Tatiana da. **A utilização da Realidade Virtual e Aumentada no Ensino de Ciências no Brasil**. [2017?], 18f, Programa de Pós-Graduação em Educação Científica e Tecnológica – Universidade Federal de Santa Catarina (UFSC).
- GREIS, Luciano Kercher; REATEGUI, Eliseo. Um Simulador Educacional para Disciplina de Física em Mundos Virtuais. **Renote: Revista Novas Tecnologias na Educação**, Porto Alegre, v. 8, n. 2, p.1-10, jul. 2010.
- O’KANE, James F.; TAYLOR R.; SPENCELEY J.R. Simulation as an essential tool for advanced manufacturing technology problems. **Journal Of Materials Processing Technology**. [s. L.], v.107, p. 412-424. nov. 2000.
- KIRNER, Claudio et al. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Belém, PA: [s.n.], 2006.
- KIRNER, Claudio; SISCOOTTO, Robson. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações**. Petrópolis, RJ: [s.n.], 2007.
- ODUM, Eugene P. **Ecologia**. Rio de Janeiro: Guanabara Koogan. 1988.434 p.
- PEREIRA, Roberta Andressa. **Apresentação do aplicativo EcosAR**. 2019. Entrevistador: Rodrigo Wernke Pereira. Blumenau. 2019. Entrevista feita através de conversação – não publicada.
- PISKE, Kevin E. **VISEDU - Aquário virtual**: Simulador de Ecossistema utilizando Animação Comportamental. 2015. 114f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- REIS, Alessandro Vieira dos; GONÇALVES, Berenice dos Santos. Interfaces Tangíveis: Conceituação e Avaliação. **Estudos em Design**, Rio de Janeiro, v. 24, n. 2, p.92-111, 2016.
- REITER, Ricardo F. **Animar**: desenvolvimento de uma ferramenta para criação de animações com realidade Aumentada e interface tangível. 2018. 76f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- RICKLEFS, Robert E. **A Economia da Natureza**. 6. ed. Rio de Janeiro: Guanabara Koogan, 2010. 572 p.
- ROMÃO, Viviane Pellizzon Agudo; GONÇALVES, Marília Matos. Realidade Aumentada: Conceitos e Design. **Unoesc & Ciência**, Joaçaba, v.4, n.1, p.23-34, 2013.
- ROYAL METEOROLOGICAL SOCIETY. **The Beaufort Scale**. Disponível em: <<https://www.rmets.org/resource/beaufort-scale>>. Acesso em: 22 jun. 2019.
- TINYBOP. **Weather**: No. 6 of The Explorer’s Library, New York, 2016. Disponível em: <<https://tinybop.com/apps/weather>>. Acesso em: 02 jun. 2019.
- TORGA, Bruno Lopes Mendes. **Modelagem, Simulação e Otimização em Sistemas Puxados de Manufatura**. 2007. 152 f. Dissertação (Mestrado) - Curso de Engenharia de Produção, Universidade Federal de Itajubá, Itajubá, 2007.
- ULLMER, Brygg; ISHII, Hiroshi. Emerging frameworks for tangible user interfaces. In: CARROL, John M. (Ed.). **Human-Computer Interaction in the New Millennium**. Ann Arbor, MI, U.S.A: University of Michigan. Ann Arbor, 2001. p. 579-601.
- UNITY. **UNITY USER MANUAL**. 2019. Disponível em: <<https://docs.unity3d.com/>>. Acesso em: 03 jun. 2019.
- VIGLIO, José Eduardo; FERREIRA, Lúcia da Costa. O conceito de ecossistema, a ideia de equilíbrio e o movimento ambientalista. **Caderno Eletrônico de Ciências Sociais**, Vitória, v. 1, n. 1, p.1-17, 2013.
- VUFORIA. **Vuforia Developer Library**. Disponível em: <<https://library.vuforia.com>>. Acesso em: 18 jun. 2019.

APÊNDICE A – DIAGRAMAS DE CLASSES

Este apêndice apresenta os diagramas de classes do aplicativo. Todos os diagramas apresentados neste apêndice estão em versão reduzida. A Figura 13 e A classe `WaterController` é encarregada pela atualização do comportamento da água. O `SnowController` é a classe responsável pela atualização do comportamento da neve. O `RainController` é encarregado de atualizar o comportamento da chuva. A classe `MenuSceneController` estabelece a orientação da tela com o aplicativo. O `CloudController` é encarregado de atualizar a lógica das nuvens da simulação. A classe `TreeGrowthStateController` faz a atualização da lógica de crescimento das árvores na simulação.

Figura 14 mostram os diagramas de classe para os controllers da aplicação. A classe `CanvasController` é responsável pelo funcionamento do menu principal da aplicação, nela a funcionalidade dos botões é implementada. A classe `SimulationController` é encarregada de controlar a lógica da simulação, executando todas as funções `Update` dos outros controllers. A classe `DayNightCycleController` é responsável por controlar o horário do dia na simulação. Ela também é responsável por fazer a rotação dos objetos que representam o sol e a lua. A classe `WindController` é responsável por atualizar os comportamentos relacionados com a velocidade do vento na simulação. O `TemperatureController` é responsável por atualizar os comportamentos relacionados com a temperatura. O `TerrainController` é encarregado de atualizar o comportamento do solo.

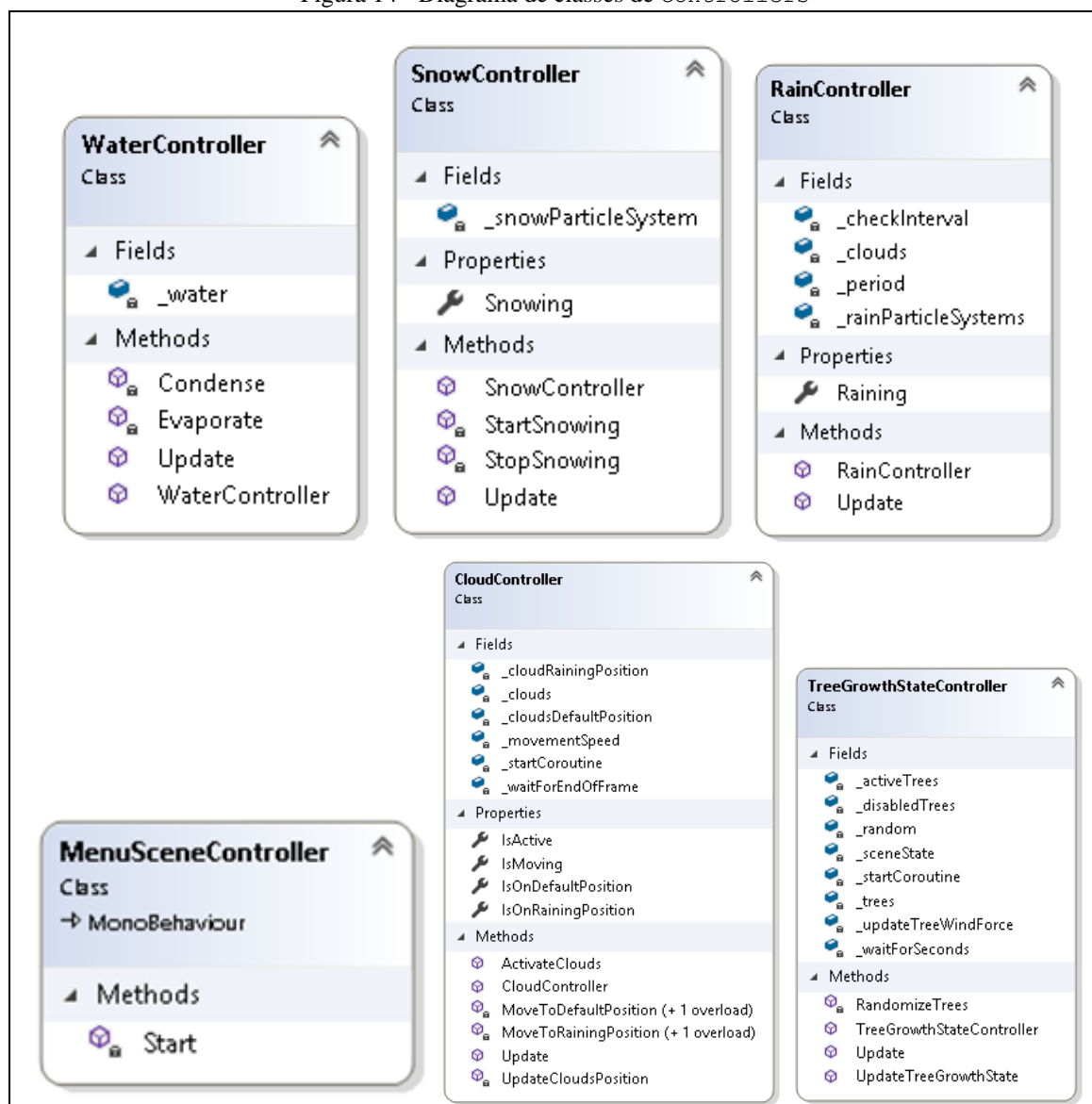
Figura 13 - Diagrama de classes de controllers



Fonte: elaborado pelo autor.

A classe `WaterController` é encarregada pela atualização do comportamento da água. O `SnowController` é a classe responsável pela atualização do comportamento da neve. O `RainController` é encarregado de atualizar o comportamento da chuva. A classe `MenuSceneController` estabelece a orientação da tela com o aplicativo. O `CloudController` é encarregado de atualizar a lógica das nuvens da simulação. A classe `TreeGrowthStateController` faz a atualização da lógica de crescimento das árvores na simulação.

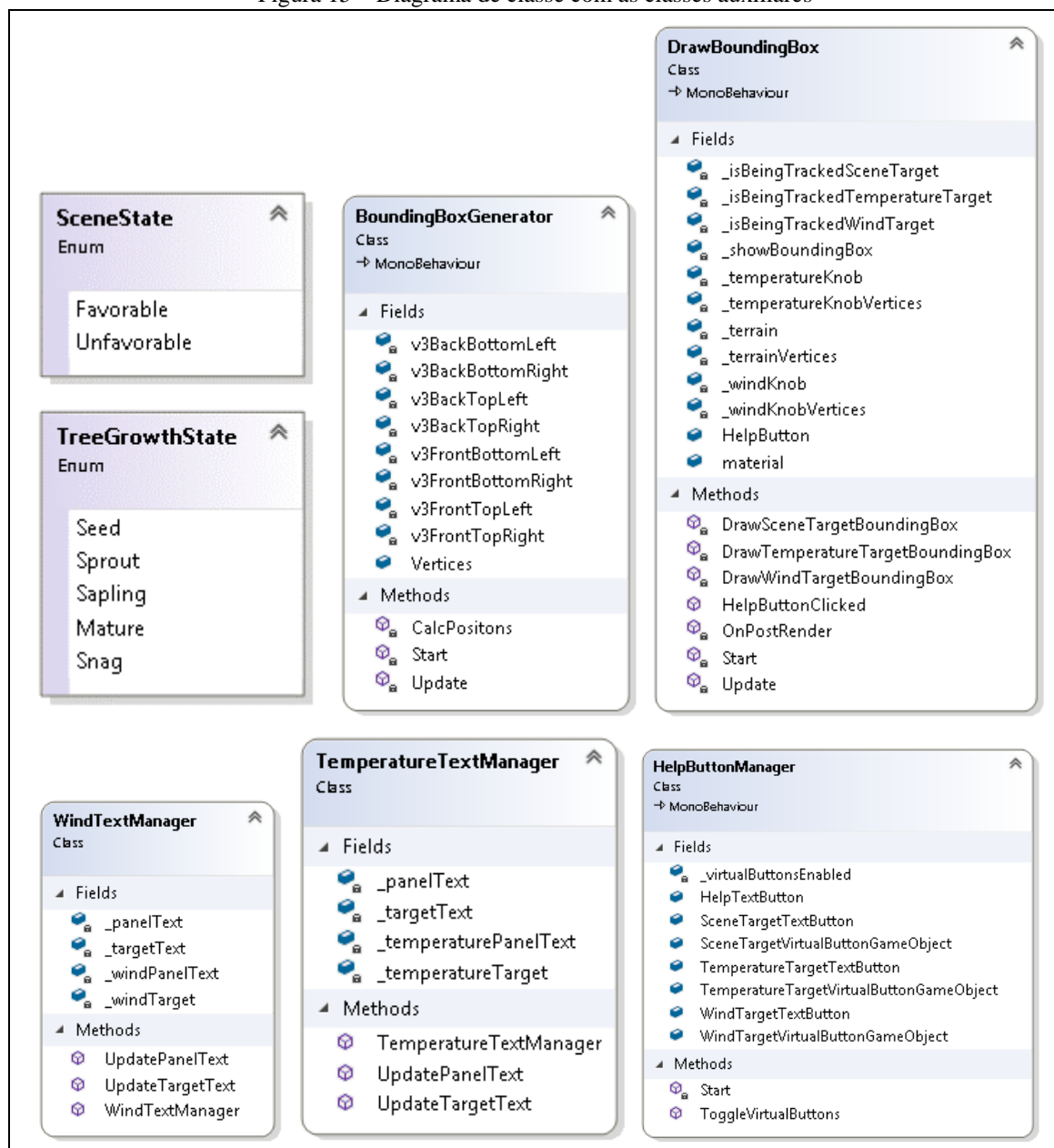
Figura 14 - Diagrama de classes de controllers



Fonte: elaborado pelo autor.

A Figura 15 e Figura 16 mostram os diagramas de classe para as classes auxiliares da aplicação. O enumerador `SceneState` armazena os dois possíveis estados da cena, o estado favorável e desfavorável. A classe `BoundingBoxGenerator` calcula as posições corretas para os vértices utilizados no desenho da *Bounding Box* de cada marcador e a classe `DrawBoundingBox` faz a renderização das mesmas. O enumerador `TreeGrowthState` armazena os estágios possíveis de crescimento das árvores. A classe `WindTextManager` é responsável por atualizar o painel da aplicação e o marcador de velocidade do vento com o valor atualizado da velocidade do vento. A classe `TemperatureTextManager` é responsável por atualizar o painel da aplicação e marcador de temperatura com o valor da temperatura atualizado. A classe `HelpButtonManager` faz o controle das funções dos botões utilizados na interface virtual da simulação.

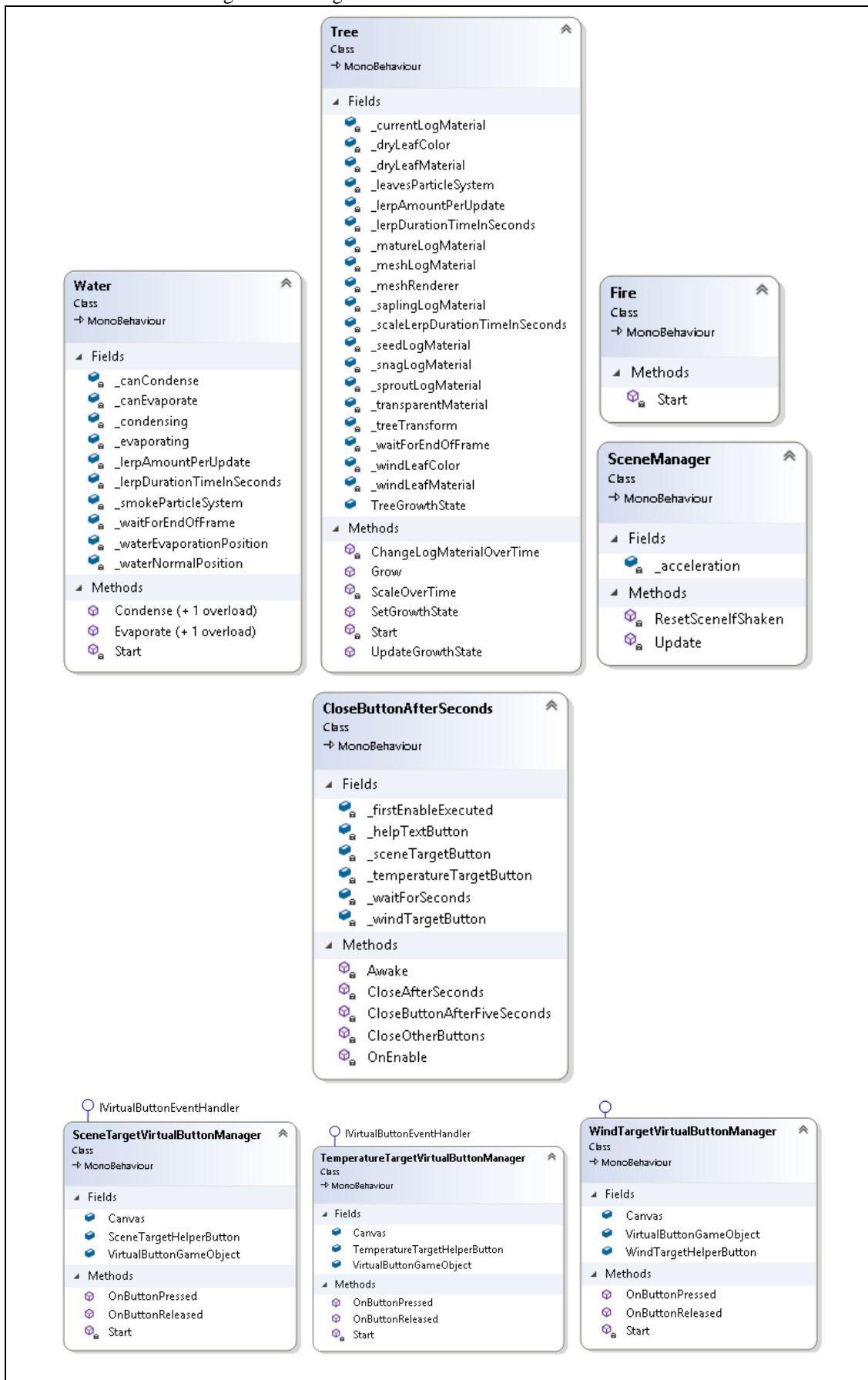
Figura 15 – Diagrama de classe com as classes auxiliares



Fonte: elaborado pelo autor.

A classe *Water* é utilizada para executar os comportamentos da água. A classe *Tree* é utilizada para executar os comportamentos das árvores, sendo que cada árvore da simulação possui uma instancia desta classe. A classe *Fire* executa o sistema de partículas que simula o fogo. A classe *SceneManager* faz o controle do acelerômetro do dispositivo, reiniciando a cena caso utilizado. A classe *CloseButtonAfterSeconds* faz com que os botões utilizados na interface virtual fechem depois de um determinado tempo. A classe *SceneTargetVirtualButton*, *TemperatureTargetVirtualButton* e *WindTargetVirtualButton* fazem o controle dos botões virtuais dos marcadores.

Figura 16 - Diagrama de classe com as classes auxiliares

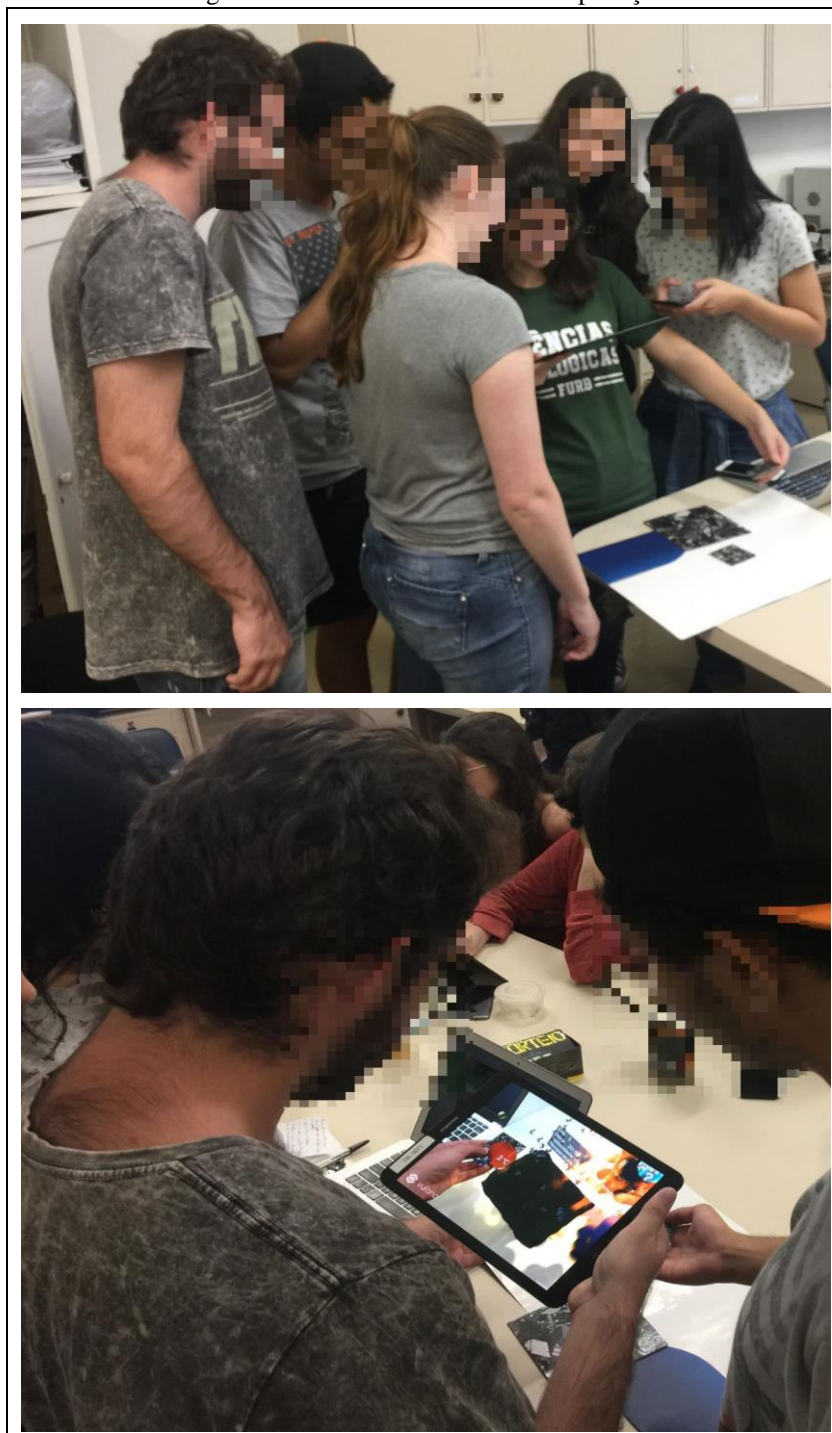


Fonte: elaborado pelo autor.

APÊNDICE B – FOTOS DOS TESTES COM TURMA DE CIÊNCIAS BIOLÓGICAS

Este apêndice apresenta as fotos fotografadas no mês de junho durante os testes do aplicativo desenvolvido, sendo utilizado por uma turma de bolsistas de ciências biológicas. A Figura 17 mostra a aplicação sendo utilizada pelos bolsistas.

Figura 17 – Acadêmicos testando a aplicação



Fonte: elaborado pelo autor.

ANEXO A – ESCALA BEAUFORT

A escala de Beaufort se trata de uma medida empírica que relaciona a velocidade do vento com condições observadas no mar ou no continente, segundo a Royal Meteorological Society (2019, tradução nossa). Ela foi utilizada no trabalho desenvolvido como ponto de referência para os comportamentos que ocorrem com a manipulação da velocidade do vento. A Figura 18 exibe a escala Beaufort.

Figura 18 – Escala Beaufort

Wind Force	Description	km/h	mph	knots	Specifications
0	Calm	<1	<1	<1	Smoke rises vertically
1	Light Air	1-5	1-3	1-3	Direction shown by smoke drift but not by wind vanes
2	Light Breeze	6-11	4-7	4-6	Wind felt on face; leaves rustle; wind vane moved by wind
3	Gentle Breeze	12-19	8-12	7-10	Leaves and small twigs in constant motion; light flags extended
4	Moderate Breeze	20-28	13-18	11-16	Raises dust and loose paper; small branches moved.
5	Fresh Breeze	29-38	19-24	17-21	Small trees in leaf begin to sway; crested wavelets form on inland waters.
6	Strong Breeze	38-49	25-31	22-27	Large branches in motion; whistling heard in telegraph wires; umbrellas used with difficulty.
7	Near Gale	50-61	32-38	28-33	Whole trees in motion; inconvenience felt when walking against the wind.
8	Gale	62-74	39-46	34-40	Twigs break off trees; generally impedes progress.
9	Strong Gale	75-88	47-54	41-47	Slight structural damage (chimney pots and slates removed).
10	Storm	89-102	55-63	48-55	Seldom experienced inland; trees uprooted; considerable structural damage
11	Violent Storm	103-117	64-72	56-63	Very rarely experienced; accompanied by widespread damage.
12	Hurricane	118 plus	73 plus	64 plus	Devastation

Fonte: Royal Meteorological Society (2019).