

Quadro 1 - Reflexão utilizada para instanciar objetos

```
var typeName = supertypeName;
if (data.ObjectValue.TryGetValue("_TYPE", out var specificTypeName))
{
    typeName = specificTypeName.StringValue;
}

var type = Type.GetType(typeName);
if (type == null)
{
    Debug.LogError("Could not find class for component '" + typeName + "'");
    yield break;
}

if (!typeof(T).IsAssignableFrom(type))
{
    Debug.LogError("Type '" + typeName + "' is not T");
    yield break;
}

if (!superType.IsAssignableFrom(type))
{
    Debug.LogError("Type '" + typeName + "' is not from file supertype '" +
        supertypeName + "'");
    yield break;
}

var _instance = (T) Activator.CreateInstance(type);
_instance.SetData(new PersistenceData(data.ObjectValue));
yield return _instance;
```

Fonte: elaborado pelo autor.

Conforme mostra o Quadro 1, o objeto Type é instanciado através da classe Activator após diversas verificações de existência de tipo com o nome informado e compatibilidade com o método IsAssignableFrom. Após isso, o método SetData passa as informações necessárias para a configuração do objeto em questão e é responsabilidade do objeto preencher suas propriedades.

2.4 TRABALHOS CORRELATOS

Nesta seção, serão apresentados três trabalhos correlatos que abordam temas relacionados a este trabalho. No Quadro 2 é apresentado o EasyEdu, trabalho desenvolvido por Corso (2017) que se trata de uma ferramenta web equipada para o desenvolvimento de jogos educacionais por professores e crianças. No Quadro 3 é apresentado o trabalho RPG4ALL desenvolvido por Pessini, Kemczinski, Hounsell (2015) que se trata de uma Ferramenta de Autoria para o desenvolvimento de jogos. Por fim, o Quadro 4 apresenta o produto RPG Maker MV da empresa Enterbrain (2015), uma ferramenta para a criação de jogos de RPG.

Quadro 2 - EasyEdu

Referência	Corso (2017)
Objetivos	Desenvolvimento de jogos educacionais em ambiente web.
Principais funcionalidades	Criação de jogos educacionais através do uso de templates. Compartilhamento dos jogos através da internet e QR Code. Personalização com o envio de imagens e cadastro de palavras.
Ferramentas de desenvolvimento	AngularJS, Google Drive, QR Code.
Resultados e conclusões	De acordo com o autor, os templates de quebra-cabeças e memória foram inclusos no planejamento mas não puderam ser desenvolvidos em tempo hábil. Porém, a equipe de pedagogia responsável pela aplicação em sala de aula obteve os resultados esperados.

Fonte: elaborado pelo autor.

Com o EasyEdu é possível desenvolver jogos baseados em templates com regras pré-definidas e executar estes jogos a partir de uma galeria. Através da ferramenta, o professor é capaz de desenvolver e compartilhar os jogos desenvolvidos através de QR Code e armazenamento no Google Drive. Um ponto importante da ferramenta é a personalização por parte do professor encarregado da utilização.

Quadro 3 - RPG4ALL

Referência	Pessini, Kemczinski, Hounsell (2015)
Objetivos	Ferramenta para especificação de Jogos Sérios por docentes sem conhecimento em desenvolvimento de jogos.

Não quebrar o quadro com a quebra de página.

Conferir em todas páginas...

3.1 VISÃO GERAL DA FERRAMENTA

A ferramenta desenvolvida se trata de um editor de jogos com foco nas mecânicas normalmente utilizadas no gênero RPG. Para isso, foram utilizados os conceitos de mapas estáticos e entidades dinâmicas formadas de eventos (vide seção 2.2), assim como programação visual através de janelas de cadastro e configuração. Assim, o fluxo da ferramenta é composto de duas partes:

- a) Editor: composto de uma interface com diversas janelas de configuração, recebe as entradas da programação visual realizada pelo usuário pelas janelas, assim como a customização dos mapas e dos atributos do jogador;
- b) Executor: após a leitura dos dados persistidos salvos através do editor, realiza a execução dos mapas e eventos das entidades, recebendo as entradas dos botões de direcionais (movimentação) e interação do jogador e alterando os estados do jogo de acordo com a programação dos eventos.

Conforme apresentado no fluxograma da Figura 5, o editor é composto de janelas para criação e edição dos mapas, criação e edição das entidades e criação e edição dos eventos. Após o editor, o executor realiza o carregamento do modelo de jogo criado e segue seu loop próprio. Este fluxo também sugere a hierarquia dos objetos na modelagem utilizada (vide seção 3.2).



Evitar deixar espaço
em branco nas
páginas.

Quadro 6 - Código do evento de exibição de mensagem

```
public class MessageEvent : GameEvent
{
    public string message;

    public override string GetNameText()
    {
        return "Mensagem";
    }

    public override string GetDescriptionText()
    {
        return message;
    }

    public override void Execute()
    {
        MessagePanel.main.SetMessage(message);
        MessagePanel.main.Toggle(true);
    }

    public override void Update()
    {
        if (Input.GetKeyDown(KeyCode.Z))
        {
            MessagePanel.main.Toggle(false);
            finishedExecution = true;
        }
    }
}
```

Fonte: elaborado pelo autor.

Alinhamento.

Assim como o editor, a cena de execução possui uma rotina similar de instanciamento de mapas dentro da classe `GameMasterBehaviour` seguindo os métodos `InstantiateMap`, `InstantiateLayer` e `InstantiateEntities`. Porém, eles utilizam menos componentes de interação visto que a lógica é ditada principalmente pela própria classe `GameMasterBehaviour` e pela classe `PlayerBehaviour`. Esta classe faz o controle de movimento e interação do jogador para a execução dos eventos supracitados. A Figura 8 mostra a comparação do instanciamento do mapa do editor com o mapa da execução do jogo.

Figura 8 - Comparação do mapa de edição e mapa de execução do jogo



Fonte: elaborado pelo autor.

3.2.3 Persistência

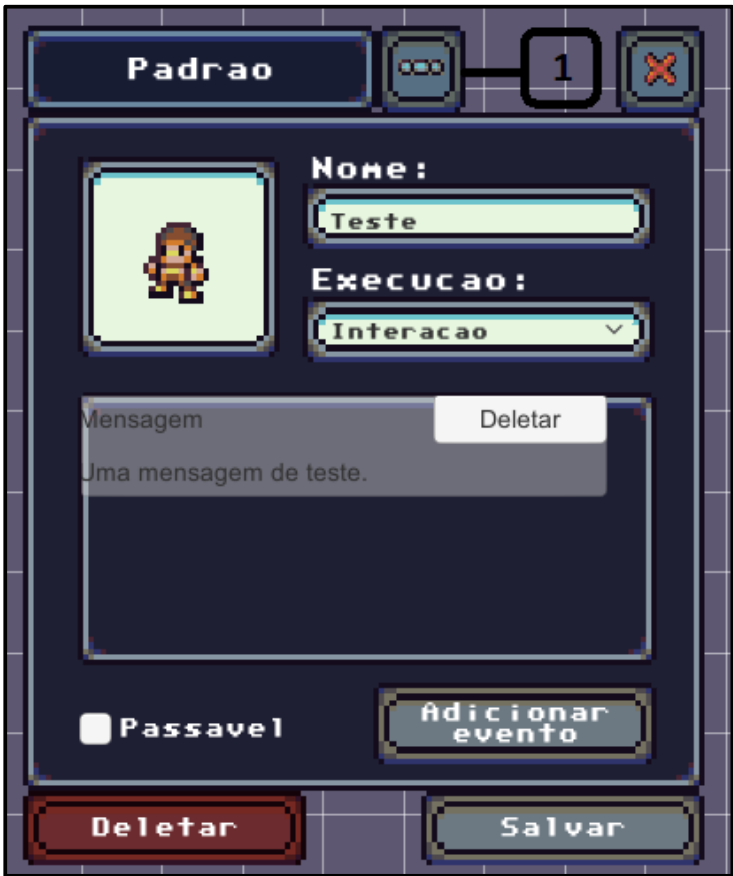
Conforme mostrado no diagrama da Figura 9, a modelagem da ferramenta é realizada visando persistir os dados necessários para a execução do jogo de acordo com a configuração realizada através dos painéis do editor. Em seguida, o Quadro 7 descreve brevemente cada modelo utilizado e seu papel dentro do contexto da ferramenta.

3.3.3 Estados, Interruptores e Variáveis

Cada entidade pode possuir diversos estados, representando comportamentos de acordo com o contexto atual do jogo. Um estado pode ser ativado através de variáveis ou interruptores. Através do botão `Estados` (Figura11-1), é possível criar novos estados definindo um nome e ativações. Uma ativação representa uma verificação de uma variável, um interruptor ou ambos. No cenário da verificação informada ser verdadeira, o estado será ativado.

Para editar a imagem, tipo de execução e eventos de uma entidade, é necessário clicar no botão `Selecionar`. Estes eventos e tipo de execução serão utilizados caso este estado esteja ativo.

Figura 11 - Exemplo do painel de edição de entidades



Fonte: elaborado pelo autor.

4 RESULTADOS

Esta seção se destina a descrever os testes realizados com a ferramenta. Para validar as funcionalidades da ferramenta, foi realizada a criação de pequenos jogos de RPG utilizando todos os eventos e funcionalidades de estado disponibilizados.

Os resultados foram satisfatórios do ponto de vista de implementação, visto que a ferramenta Unity se mostrou útil e capaz de realizar todas as funcionalidades necessárias. A principal utilização da ferramenta Unity foi o `Canvas` (ver seção 3.2.2) que se mostrou de grande valia para a criação rápida dos diversos painéis de cadastro necessários para a criação do editor.

A primeira implementação foi realizada utilizando os conceitos puros de ECS (ver seção 2.2); porém, tais conceitos se mostraram demasiadamente abstratos para a implementação do editor. A solução foi utilizar conceitos mais brandos e substituir a execução simultânea dos componentes e sistemas por eventos que seriam executados linearmente, similar ao trabalho correlato RPG Maker (ver seção 2.4).

Os eventos criados para a utilização do editor foram satisfatórios e a lógica disponível com a utilização das variáveis e interruptores é suficiente para **um pequeno jogo** de RPG. Mais conceitos que interessantes de serem abordados eram itens e batalhas, comuns dentro do gênero de RPG, mas a complexidade ultrapassaria o escopo do projeto. Além disso, a movimentação pode ser melhorada utilizando um algoritmo de `pathfinding` como o A*.

A utilização do editor se tornou mais complexa do que o necessário, levando em conta o leiaute das telas e a relação dos modelos apresentados. Como os conceitos da seção 3.3 são abstratos, existe uma necessidade de exemplos práticos para uma boa compreensão do usuário.

5 CONCLUSÕES

Diante dos resultados apresentados, conclui-se que a ferramenta se provou capaz de criar jogos de RPG utilizando lógicas de programação visual e customização de mapas. Parte das funcionalidades propostas do editor foram alcançadas, ainda que sua usabilidade tenha deixado a desejar do ponto de vista de facilidade de uso.

um jogo pequeno

A ferramenta Unity se mostrou poderosa, embora burocrática em alguns pontos (ver seção 4.1). A biblioteca SimpleJSON foi de grande valia para as rotinas de persistência e abstraíram muito da manipulação dos arquivos JSON.

As possíveis extensões propostas para seguir a implementação são:

- a) adicionar novos eventos para lógicas mais complexas ou funcionalidades visuais;
- b) implementar uma lógica de **pathfinding** como A* para a movimentação de entidades;
- c) adicionar o conceito de itens conforme usado normalmente no gênero RPG;
- d) implementar batalhas conforme usado normalmente no gênero **RPG**;

REFERÊNCIAS

ALVES, Lynn et al. Ensino On-Line, jogos eletrônicos e RPG: Construindo novas lógicas. In: **Conferência eLES**. 2004. p. 49-58.

BITTENCOURT, João Ricardo. GIRAFFA, Lucia Maria. Modelando Ambientes de Aprendizagem Virtuais utilizando Role-Playing Games. In: Simpósio Brasileiro de Informática na Educação, 14, 2003, Rio de Janeiro. **Anais...** Porto Alegre: PUCRS, 2003. p. . Disponível em: < <http://www.nce.ufrj.br/sbie2003/publicacoes/paper71.pdf>>. Acesso em: 02 jun. 2020.

CHAGAS, Artur Alves de Oliveira. **O transbordamento do lúdico e da biopolítica em jogos Massive Multiplayer Online**: um estudo sobre World of Warcraft. 2010. 157 p. Tese (Mestrado em Educação) - Universidade de São Paulo, São Paulo, 2010.

CORSO, Felipe L. **EasyEdu**: Editor web para jogos multitoque. 2017. 93 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CRAWFORD, C. **The Art of Computer Game Design**, Washington State University, 1982.

DRAGONDEPLATINO. DAWNBRINGER. **DawnLike**: 16x16 Universal Rogue-Like Tileset v1.81 | OpenGameArt.org. Disponível em: < <https://opengameart.org/content/dawnlike-16x16-universal-rogue-like-tileset-v181>>. Acesso em: 02 jun. 2020.

ENTERBRAIN, Make Your Own Game With RPG Maker. Disponível em: <<https://www.rpgmakerweb.com/>>. Acesso em: 02 jun. 2020.

FORD, Timothy. Overwatch Gameplay Architecture and Netcode. In: **Game Developer’s Conference**. 2017. Disponível em: <<https://www.youtube.com/watch?v=W3aieHjyNvw&t=839s>>. Acesso em: 02 jun. 2020.

GRANDO, Anita; TAROUÇO, Liane Margarida Rockenbach. **O uso de jogos educacionais do tipo RPG na educação**. RENOTE-Revista Novas Tecnologias na Educação, v. 6, n. 1, 2008.

GURZYNSKI, Cleber; HOUNSELL, Marcelo; KEMCZINSKI, Avaniide. Análise de um jogo RPG educacional produzido pelo próprio docente, auxiliado por Ferramenta de Autoria. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2016. p. 617.

HALL, Daniel Masamune. **ECS Game Engine Design**. [S.I.]. 2014. Disponível em: < <https://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1138&context=cpsp>>. Acesso em: 02 jun. 2020.

HUIZINGA, Johan. **Homo Ludens**. São Paulo: Editora Perspectiva, 2000.

MICROSOFT. **Reflexão (C#)**. Disponível em : <<https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/reflection>>. Acesso em: 02 jun. 2020.

NINTENDO OF AMERICA, Inc. **The Legend of Zelda Instruction Booklet**. Japão. 1986. p. 28.

PEREIRA, C.K., Andrade, F.M., Freitas, L.E.R. **Desafio dos Bandeirantes – Aventuras na Terra de Santa Cruz**, GSA, 1992.

PESSINI, Adriano; KEMCZINSKI, Avaniide; DA SILVA HOUNSELL, Marcelo. **Uma Ferramenta de Autoria para o desenvolvimento de Jogos Sérios do Gênero RPG**. Anais do Computer on the Beach, p. 071-080, 2015.

PINCHEFSKY, Carol. **Role-Playing Gamers Have More Empathy Than Non-Gamers**. [S.I]: Geek and Sundry, 2016.

SALDANHA, Ana Alayde; BATISTA, José Roniere Moraes. **A concepção do role-playing game (RPG) em jogadores sistemáticos**. Psicologia: ciência e profissão, v. 29, n. 4, p. 700-717, 2009.

SQUIRE, Kurt. Video Games In Education. **International Journal of Intelligent Games & Simulation**. Cambridge, v. 1, n. 1, p. 2-4, 2003. Disponível em: . Acesso em: 02 jun. 2020.

1
Fonte courier.

2
Terminar frase com ponto final.

3
Muda este texto para ser um texto “corrido” com itens, mas sem quebra de linha.

As possíveis extensões propostas para seguir a implementação são: a) adicionar ...

4
ATENÇÃO: Todas as citações que aparecem no texto devem ter uma referência bibliográfica descrita aqui.

E todas as referências bibliográficas devem ser citadas no texto... conferir com muita atenção.