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.

1 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).

1

1 Evitar deixar espaço em branco nas páginas.

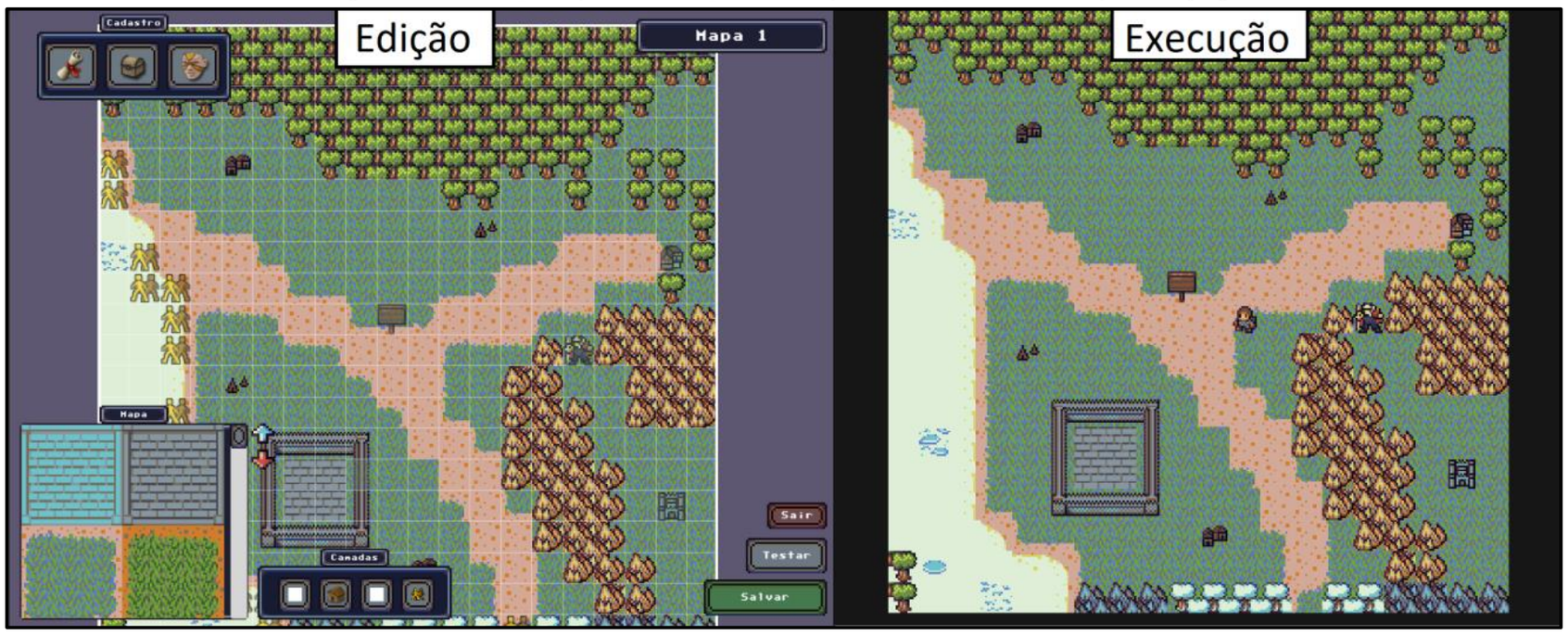
Quadro 6 - Código do evento de exibição de mensagem

```
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.

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.

Alinhamento.