

# **BluCraft: Uma ferramenta para a criação de jogos de RPG digitais**

Aluno: Guilherme Paz Silva

Orientador: Dalton Solano dos Reis

# Roteiro

- Introdução
  - Objetivos
  - Fundamentação Teórica
  - Trabalhos Correlatos
- Requisitos
- Especificação
- Implementação
- Resultados e Conclusões
- Apresentação Prática

# Introdução

- O projeto surgiu do interesse e possibilidades de exploração do tema
- Permite introduzir leigos a conceitos mais complexos de lógica através da criação de jogos
- Diversas formas de extensão

# Objetivos

- Desenvolver um editor de jogos no estilo RPG que disponibilize ferramentas para a criação de contextos e objetivos customizáveis pelo usuário;
  - permitir a criação de entidades executáveis através de programação visual;
  - permitir a customização de cenários interconectados;
  - executar e compartilhar os jogos criados através do editor através de persistência.

# Fundamentação Teórica

- Role Playing Game (RPG)
- Entity Component System (ECS)
- Reflexão em C#

# Role Playing Game (RPG)

- Jogo de interpretação de personagens
- Nasceu no RPG de mesa com Dungeons and Dragons
- Ascendeu aos meios digitais na década de 80 e 90
- Tem como principal foco a história e evolução de personagens
- Zelda, Final Fantasy, Pokémon, Skyrim

# Entity Component System (ECS)

- Arquitetura baseada em entidades, componentes e sistemas
- **Entidades** representam objetos individuais somente com um identificador
- **Componentes** armazenam dados e são atrelados a um indivíduo (entidade)
- **Sistemas** performam ações com base nos dados de entidades
- Unity usa uma arquitetura similar

# Reflexão em C#

- Instanciação através de dados genéricos
- Peça chave para estruturação de projetos baseados em ECS

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



# Trabalhos Correlatos

- EasyEdu
- RPG4ALL
- RPG Maker

# EasyEdu (Corso, 2017)

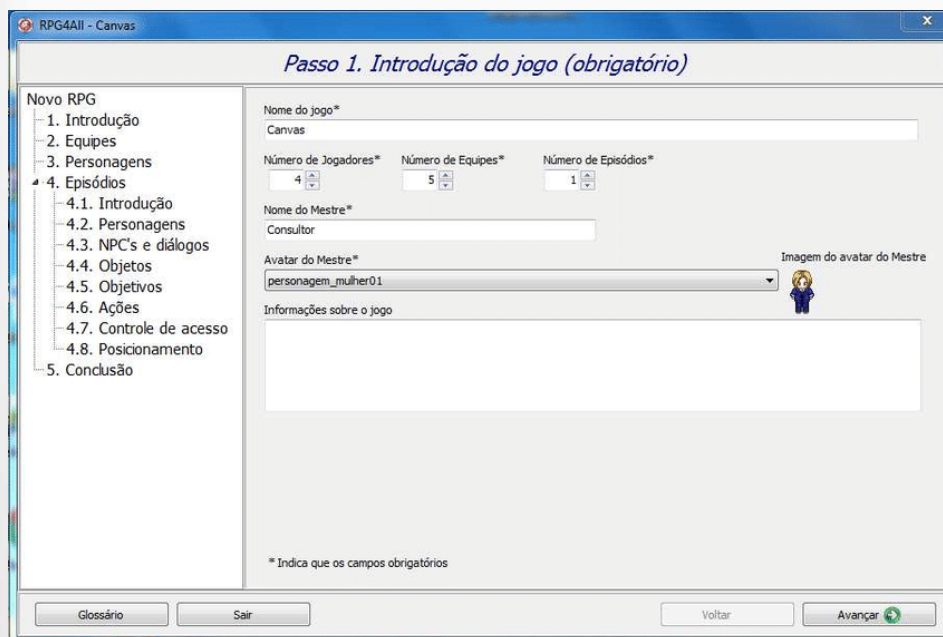
- Criação de jogos educacionais (templates)
- Web
- Integração com Google Drive



The screenshot displays the EasyEdu Editor interface. At the top left, it says "EasyEdu - Editor" with a menu icon. At the top right, there is a "Acessar com o Google" button with the Google logo. On the left side, there is a sidebar with three options: "Galeria" (with a gallery icon), "Meu álbum" (with a folder icon), and "Jogo" (with a game controller icon). The main area is titled "GALERIA DE CONTEÚDOS PADRÃO". Below the title, there is a light blue box containing the text: "Aqui são apresentados os conteúdos padrão. Você pode escolher um deles e customizá-lo sem compromisso ;) Se preferir, você pode acessar com sua conta Google. Clicar no botão 'Adicionar conteúdo' e personalizar o conteúdo como quiser." Below this text, there are three content cards. The first card shows a collection of colorful geometric shapes (triangles, squares, circles, a star) and is labeled "Matemática - Figuras g...". The second card shows the word "Mamíferos" in blue text and is labeled "Biologia - Mamíferos". The third card shows the Brazilian flag and is labeled "Geografia - Estados br...".

# RPG4ALL (Pessini, Kemczinski, Hounsell, 2015)

- Especificação de Jogos Sérios
- Integração com o Tiled
- Conceitos: Episódios, Objetos e Personagens



RPG4ALL - Canvas

*Passo 1. Introdução do jogo (obrigatório)*

Novo RPG

1. Introdução
2. Equipes
3. Personagens
4. Episódios
  - 4.1. Introdução
  - 4.2. Personagens
  - 4.3. NPC's e diálogos
  - 4.4. Objetos
  - 4.5. Objetivos
  - 4.6. Ações
  - 4.7. Controle de acesso
  - 4.8. Posicionamento
5. Conclusão

Nome do jogo\*  
Canvas

Número de Jogadores\* 4    Número de Equipes\* 5    Número de Episódios\* 1

Nome do Mestre\*  
Consultor

Avatar do Mestre\*  
personagem\_mulher01    Imagem do avatar do Mestre

Informações sobre o jogo

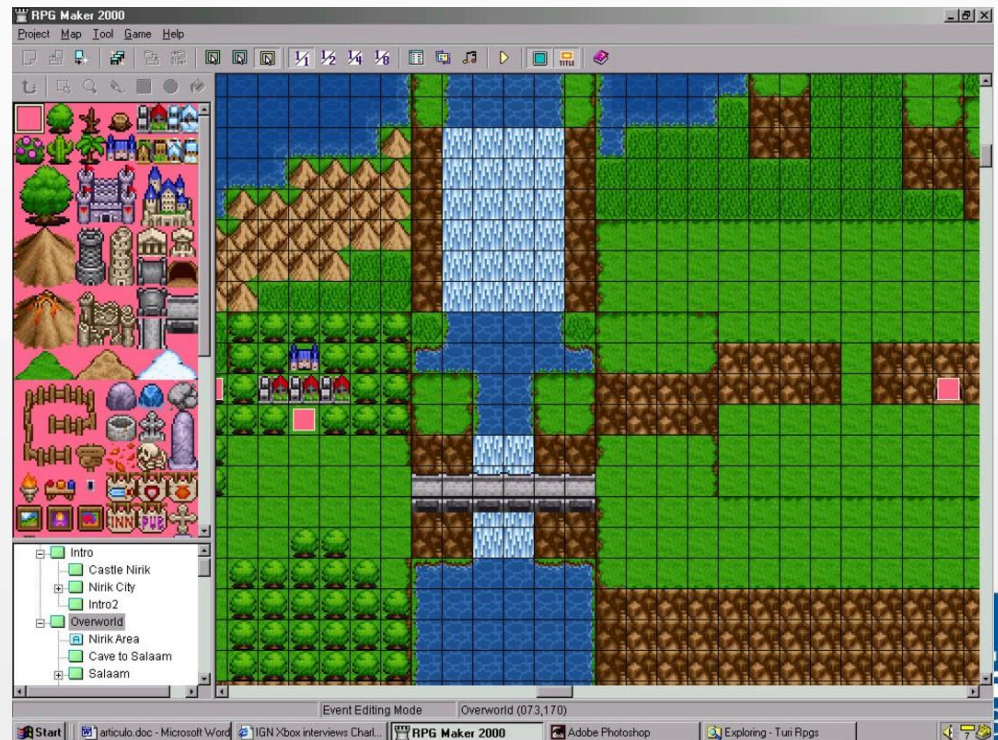
\* Indica que os campos obrigatórios

Glossário    Sair    Voltar    Avançar



## RPG Maker (Enterbrain, 2015)

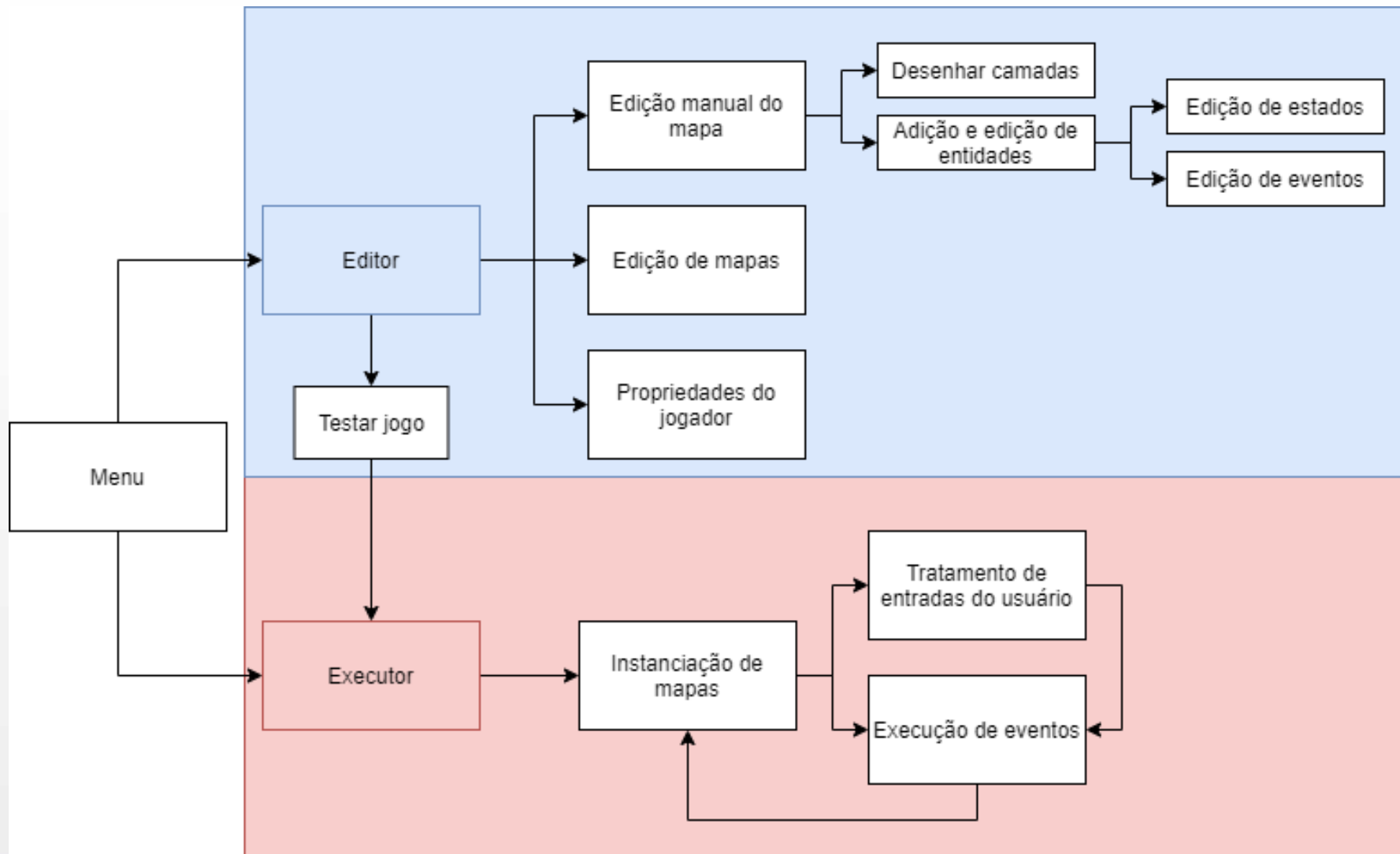
- Software comercial para criação de jogos de RPG
- Baseia-se em cadastros de banco de dados, mapas e eventos
- Distribui pacotes gráficos junto da ferramenta



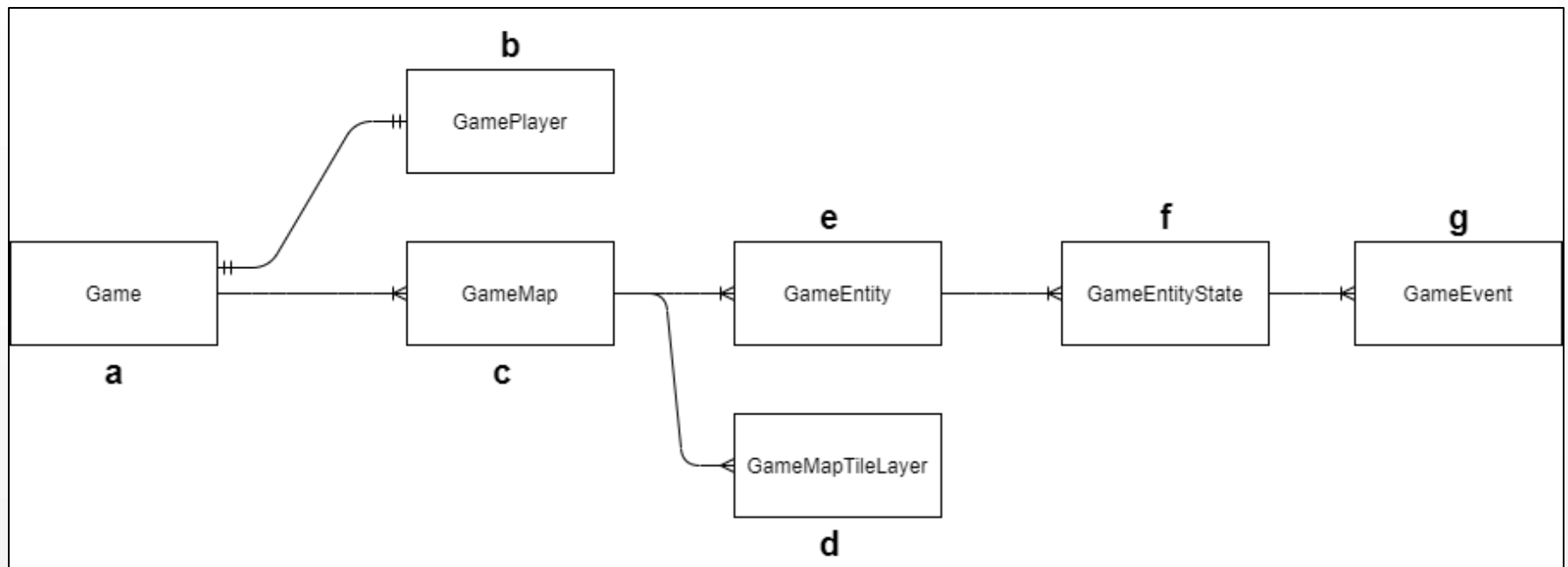
# Requisitos

- Funcionais:
  - Permitir a criação de jogos digitais de RPG através de um editor visual baseado em mapas;
  - Carregar e executar os jogos criados pelo editor.
- Não-Funcionais:
  - Ser desenvolvida com a *engine* Unity;
  - Armazenar a estrutura dos jogos criados em arquivos JSON.

# Especificação Fluxo da ferramenta



# Especificação Persistência dos dados

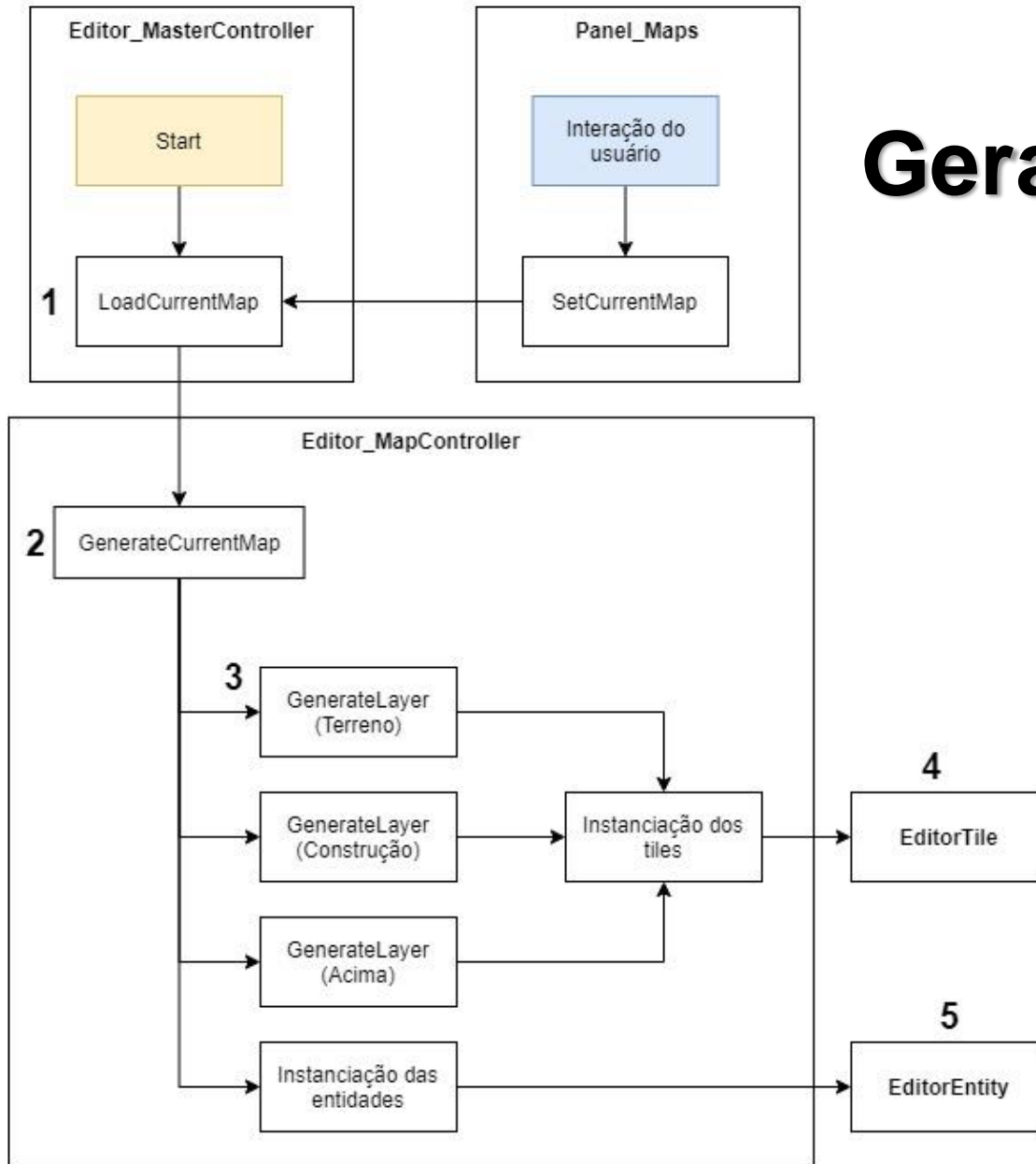


# Implementação

- Geração de Mapas
- Eventos
- Persistência



# Geração de Mapas



# Eventos

- Métodos abstratos
  - GetNameText
  - GetDescriptionText
  - Execute
  - Update

```
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;
        }
    }
}
```

# Persistência

- Reflexão
- PersistenceData

```
public T Get<T>(string key, T defaultValue = default)
{
    var obj = Get(key);
    if (obj == null)
        return defaultValue;

    if (typeof(T) == typeof(string))
        return (T)(object) obj.StringValue;
    if (typeof(T) == typeof(int))
        return (T)(object)int.Parse(obj.StringValue);
    if (typeof(T) == typeof(bool))
        return (T)(object)obj.BooleanValue;
    if (typeof(T) == typeof(PersistenceData))
        return (T)(object)new PersistenceData(obj.ObjectValue);
    if (typeof(T) == typeof(List<PersistenceData>))
    {
        var l = new List<PersistenceData>();
        foreach (var v in obj.ArrayValue)
        {
            l.Add(new PersistenceData(v.ObjectValue));
        }

        return (T)(object)l;
    }
    if (typeof(T) == typeof(Sprite))
        return (T)(object) obj.StringValue.ToSprite();
    if (typeof(T) == typeof(Vector2))
        return (T)(object)obj.StringValue.ToVector2();

    return (T)(object)obj;
}
```

# Análise dos Resultados

- Análise foi a criação de um jogo simples com o editor:
  - Demonstrou possibilidade de criação de lógicas simples
  - Apresentou dificuldade de uso por conta da interface
  - Alguns conceitos fizeram falta, como itens ou batalhas; clássicos no contexto do gênero RPG
- Comparação com Trabalhos Correlatos:
  - RPG Maker: similaridade na utilização de eventos
  - RPG4ALL: similaridades nos conceitos de Personagens e Objetos abstraídos para Entidades; possui outros conceitos interessantes de serem estudados
  - EasyEdu: conceito de template não se encaixou na execução do projeto e teve um distanciamento maior

# Conclusões e Sugestões

- O objetivo foi alcançado visto que a ferramenta desenvolvida foi capaz de criar jogos do estilo RPG com customização de contextos;
- Sugestões de extensão:
  - Novos eventos para lógicas mais complexas
  - Utilização de *pathfinding* na movimentação
  - Conceitos como itens e batalhas
  - Salvar o progresso dentro de um jogo
  - Interfacear o compartilhamento de jogos criados

# Apresentação Prática