

Workshop Interno - GDP

Vinícius Garcia - 07/2017

Scriptable Object



CONSIDERAÇÕES INICIAIS



CONSIDERAÇÕES INICIAIS

- Vai ser recheado de conteúdo.
- Não é nenhum conhecimento arcano.
- Se tiverem alguma dúvida me chamem.
- Basta que prestem atenção e se necessário, anotem!



CONSIDERAÇÕES INICIAIS

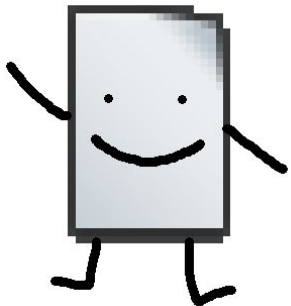
- Motivação
 - Por que é útil? Que problemas soluciona?
- Implementação
 - Botar a mão na massa!
 - ScriptableObject;
 - Unity Attributes;
 - System.Serializable;
 - Introdução a CustomEditor.



MOTIVAÇÃO



MOTIVAÇÃO



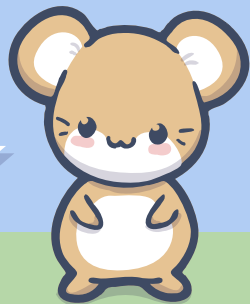
O que é um ScriptableObject?

- Primo do MonoBehaviour;
- Arquivo para armazenamento de dados, assim como XML, JSON;
- Integrado diretamente com a Unity.



MOTIVAÇÃO

- Um prefab com um array de 1 milhão de inteiros consome 40 MB na memória.
 - 10 desses objetos em cena e você está em apuros;
 - ScriptableObjects são dados terceirizados.
- Como são arquivos separados, as alterações em Runtime são armazenadas em disco.
 - Facilita debug e teste.



MOTIVAÇÃO

- JSON
 - Type - não possui forma fácil de garantir a corretude.
 - Como adicionar um novo atributo?
 - Integrar com a Unity requer certo esforço.

```
[  
  {  
    "id": 0,  
    "name": "sword of fire",  
    "type": "sword"  
  },  
  {  
    "id": 1,  
    "name": "sword of water",  
    "type": "sword"  
  },  
  {  
    "id": 2,  
    "name": "sword of earth",  
    "type": "word"  
  }  
]
```



MOTIVAÇÃO

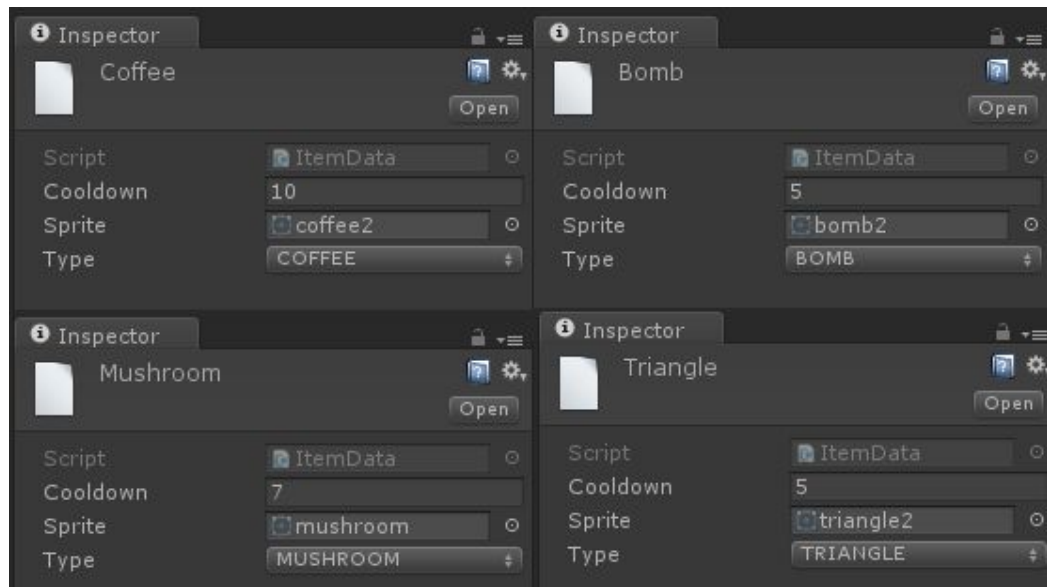
- Visualização não é intuitiva (cadê o meu dropdown?).
- Só piora para casos mais complexos.

```
[  
  {  
    "id": 0,  
    "name": "sword of fire",  
    "type": "sword",  
    "color": "#2a0d93",  
    "sprite": "/sprites/sword.png"  
  },  
  {  
    "id": 1,  
    "name": "sword of water",  
    "type": "sword",  
    "color": "#2ad3f3",  
    "sprite": "/sprites/sword.png"  
  }  
]
```



MOTIVAÇÃO

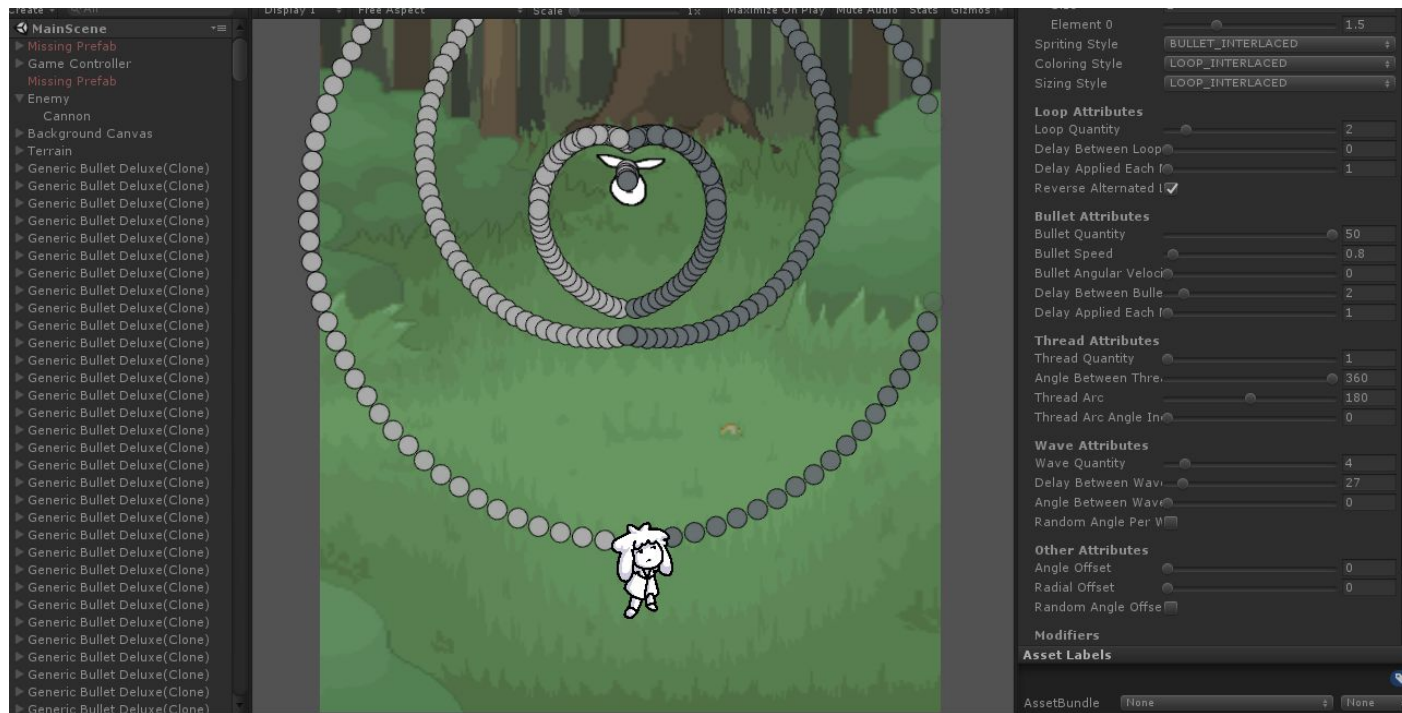
- Não precisa ser algo complexo.



EXEMPLO DO VCS



MOTIVAÇÃO



EXEMPLO DO NOIR



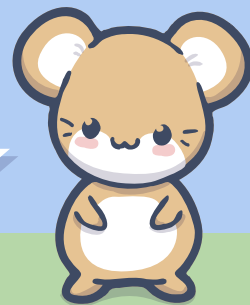
IMPLEMENTAÇÃO



IMPLEMENTAÇÃO



PROJETO ATUAL



IMPLEMENTAÇÃO

Estado atual:

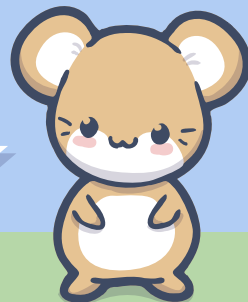
- Movimentação aleatória do bicho;
- *Finesse* do movimento controlada por variáveis públicas;
- Variáveis alteradas são perdidas quando o jogo é fechado.
- [bota pra game]



IMPLEMENTAÇÃO

Onde entra o ScriptableObject?

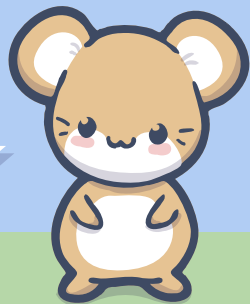
- Armazenar as informações sobre um tipo de inimigo.
- *Distance, speed, name e color.*



IMPLEMENTAÇÃO

O que faremos?

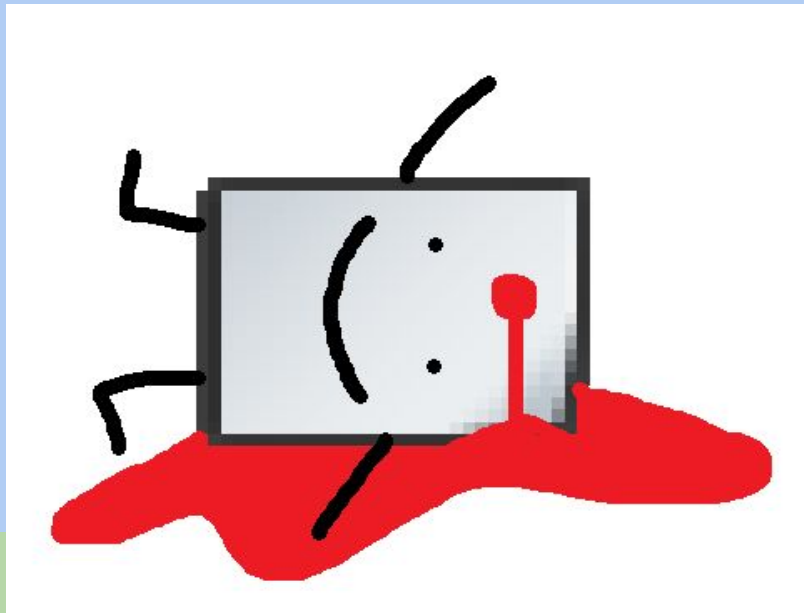
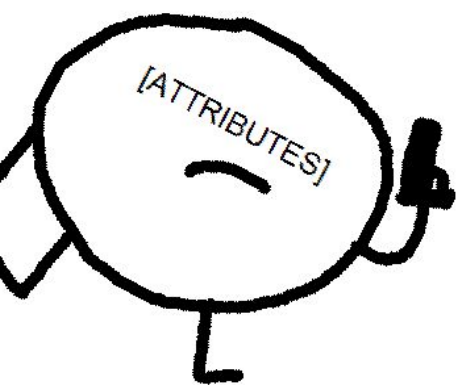
- EnemyData
 - Nosso amigo ScriptableObject.
- EnemyDataHandler
 - Gerenciará a comunicação entre a classe Enemy (com o funcionamento interno do bicho) e a classe EnemyData (com os dados desejados).
- [bota pra game]



FIM DO SCRIPTABLEOBJECT

Mas ainda temos diversos outros temas para cobrir!

- Coisas úteis para visualizarmos sistemas e a Unity.
- Items, skills ou



IMPLEMENTAÇÃO

O que são Attributes?

- Diretrizes internas da Unity para especificar como certas informações serão disponibilizadas no Inspector.

Por que estamos falando disso?

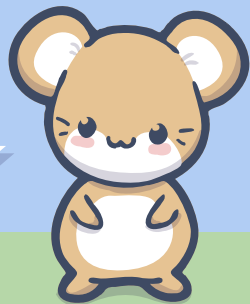
- Porque é rápido, simples e também nos ajuda a gerenciar e visualizar dados dentro da Unity.



IMPLEMENTAÇÃO

Alguns Attributes

- [Header]
 - Organização do Inspector.
- [Tooltip]
 - Descrição de um determinado campo.
- [Range]
 - Delimita um intervalo de valores que um determinado campo pode assumir.
- [bota pra game]



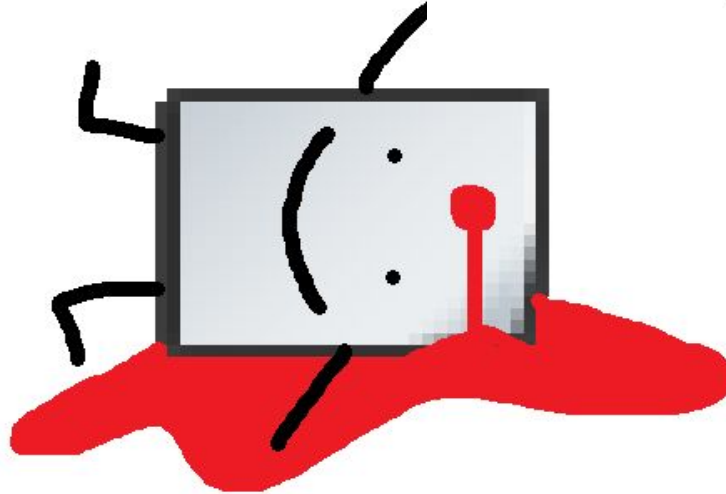
~~FIM DO SCRIPTABLE OBJECT~~ FIM DOS ATTRIBUT



Mas ainda temos diversos outros
temas para cobrir



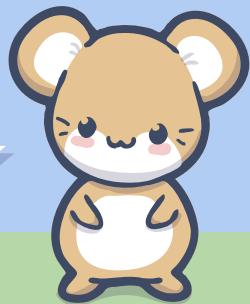
izarmos
se
y.
kills ou



IMPLEMENTAÇÃO

O que é o CustomEditor?

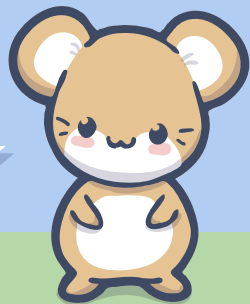
- Editor customizado. Permite ao usuário *realmente* especificar como ele deseja que uma classe seja disponibilizada, podendo inserir funcionalidades que atendam às particularidades ao projeto sendo desenvolvido.



IMPLEMENTAÇÃO

Por que estou falando do CustomEditor?

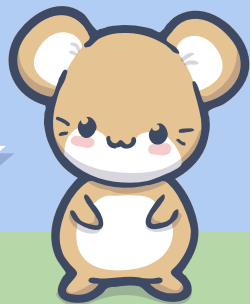
- Pra dar um gostinho do quão fácil é fazer alguma coisa que seja interessante e fazer vocês buscarem mais sobre isso depois.



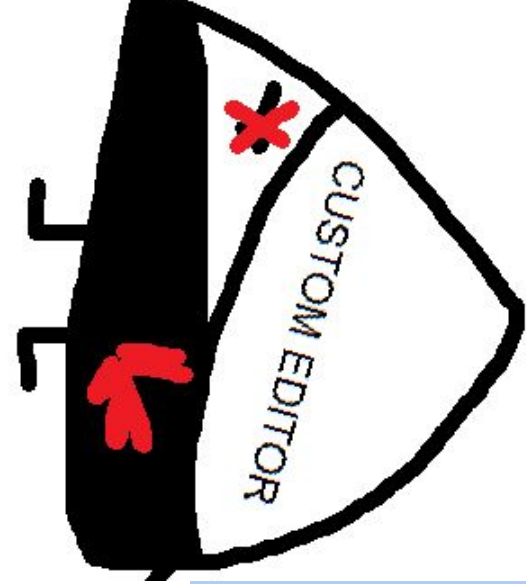
IMPLEMENTAÇÃO

O que que faremos de CustomEditor?

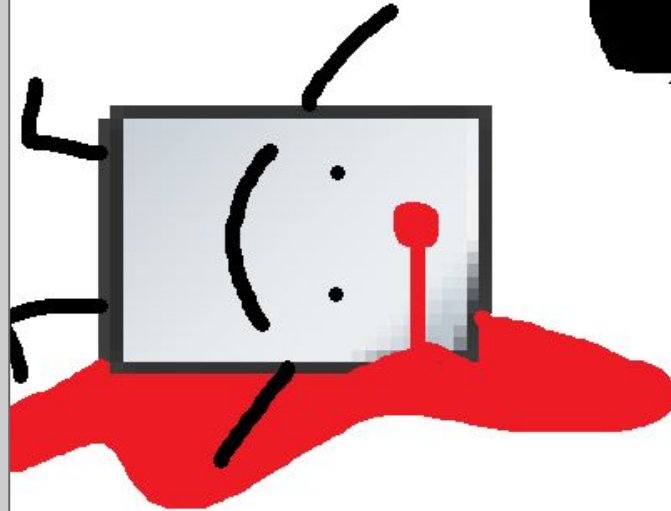
- Um botão *Preview* que nos permite visualizar os efeitos da troca de ScriptableObject antes de dar Play.
- [bota pra game]



FIM DO SCRIPTABLE OBJECT
FIM DOS ATTRIBUTES ?
FIM DO CUSTOM EDITOR



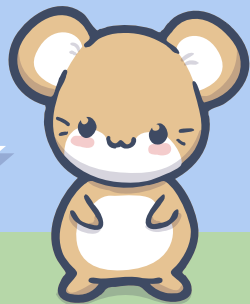
utros
lizarmos
as e
ty.
skills ou

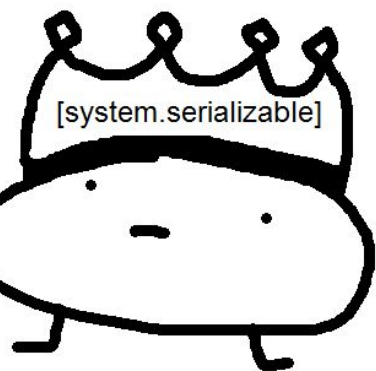


IMPLEMENTAÇÃO

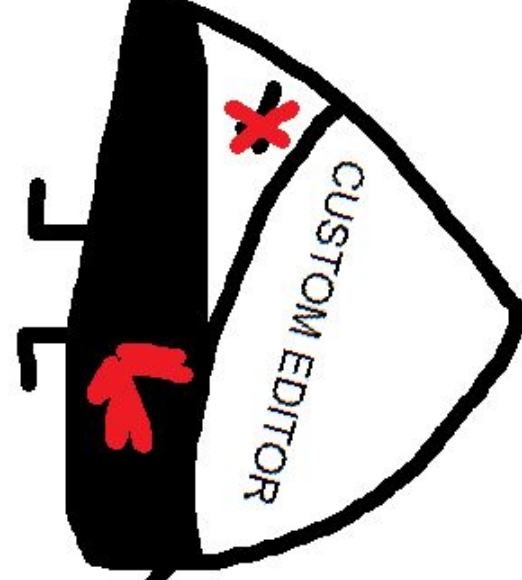
Outro atributo importante:

- [ExecuteInEditMode]
 - Toda instância de objeto que contiver esta classe irá executar no modo *Edição*;
 - Algumas diferenças: Update é chamado não a todo frame, mas só quando alguma coisa muda.
 - Útil para testar alterações em coisas que não dependem de outros elementos do jogo (backgrounds).
 - [bota pra game]





DO SCRIPTABLE OBJECTS
M DOS ATTRIBUTES ?
M DO CUSTOM EDITOR
SYSTEM.SERIALIZABLE

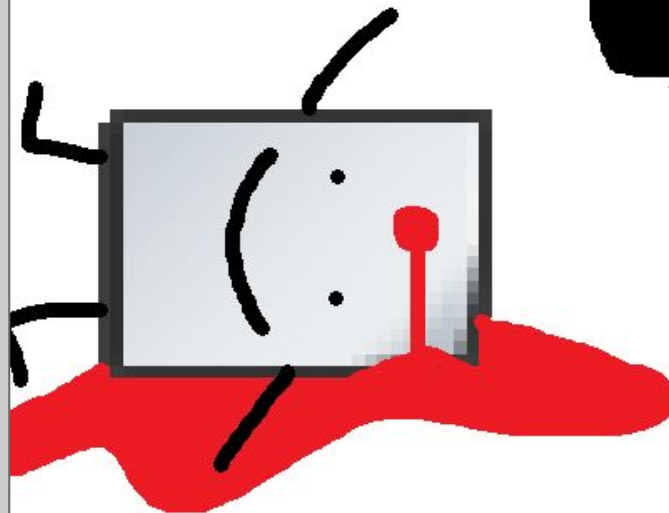


Mas ainda temos diversos outros
temas para cobrir!



a visualizarmos
sistemas e
na Unity.

itens, skills ou



IMPLEMENTAÇÃO

O que é serialização?

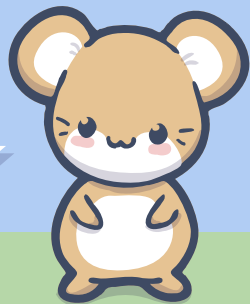
- Processo de converter estruturas de dados ou o estado de um objeto em um formato que poderá ser armazenado e posteriormente reconstruído.



IMPLEMENTAÇÃO

Inspector

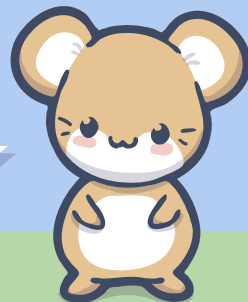
- O valor de um campo apresentado no Inspector não é a mesma variável utilizada no funcionamento interno da Engine, mas sim uma serialização dela.
- Ou seja, a variável dentro da sua classe e a variável mostrada no Inspector são duas variáveis diferentes, que a Unity mantém atualizadas para você.



IMPLEMENTAÇÃO

Inspector

- Como a Unity que realiza esse trabalho pra você, ela que decide quais variáveis vão aparecer no Inspector. Se ela quiser que alguma variável não apareça, ela não vai aparecer.
- Notoriamente, quando elas são marcadas como *private*. Mas nesse caso, é só botar como *public*!



IMPLEMENTAÇÃO

(PARÊNTESES)

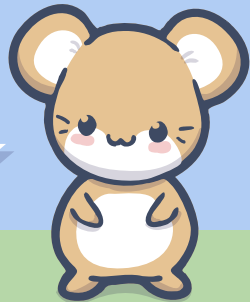
- *Public* significa que aquela variável é pública e pode ser acessada por todas as outras classes. Não só pode ser *acessada*, como *alterada*!
 - Se você só quiser que ela apareça no Inspector, esta descrição de *public* não reflete a realidade!
 - Exemplo: ID de um inimigo.
 - Atenção: não é um erro funcional, mas é um buraco no planejamento dos sistemas do seu jogo!
- Solução: [SerializeField]
- De modo análogo, temos o [HideInInspector]
 - Uma variável *public* que não é mostrada no Inspector.
- [bota pra game]



IMPLEMENTAÇÃO

Inspector

- Não são todas as variáveis *public* que são mostradas no Inspector. Na verdade, existe um subgrupo muito importante de tipos que não são serializados: classes custom.
- Por que você usaria uma classe custom?
 - Armazenar dados!



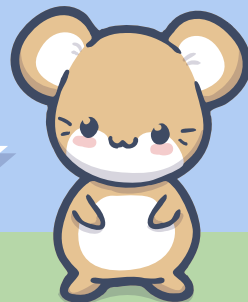
IMPLEMENTAÇÃO

espera nao entendi
por que voce nao faz
um
scriptableobject??



IMPLEMENTAÇÃO

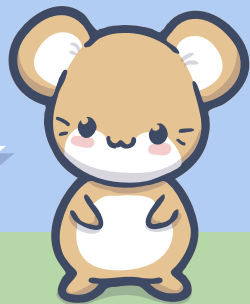
Porque as vezes você acaba lidando com dados 'descartáveis' e não quer fazer um arquivo pra cada coisa!



IMPLEMENTAÇÃO

Exemplo:

- Sistema de Skills
 - Cada Skill possui uma lista de Efeitos.
 - Um Efeito possui:
 - Tipo (BUFF ou DEBUFF)
 - Duração (float)
- Como faríamos isso?

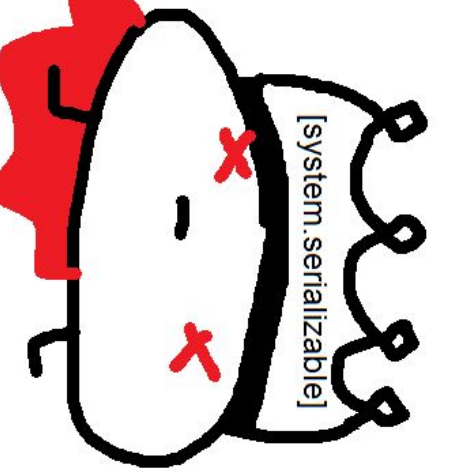


IMPLEMENTAÇÃO

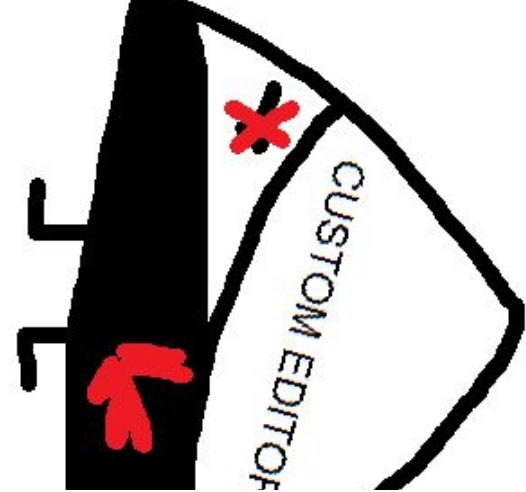
A forma intuitiva não funciona do jeito que você imaginaria!

- [bota pra game]





DO SCRIPTABLEOBJECTS
HAVE ATTRIBUTES ?
CAN WE DO CUSTOM EDITOR
SYSTEM.SERIALIZABLE



Mas ainda temos diversos
temas para cobrir!



fim

na Unity.

items, skills ou



BIBLIOGRAFIA



BIBLIOGRAFIA

- <https://www.youtube.com/watch?v=VBA1QCoEAX4>
- <https://docs.unity3d.com/Manual/class-ScriptableObject.html>
- <https://www.youtube.com/watch?v=ItZbTYO0Mnw>



DÚVIDAS

