

A blurred, dark blue background image of a game controller, likely a PlayStation 4 DualShock 4, oriented diagonally. The controller's buttons and touchpad are faintly visible through the blur.

# Unity Crash Course

# 6

EltonCN

# Outline

Scriptable Objects

Navmesh

Performance

## **Objetivo**

Aprender técnicas para arquitetura de código na Unity, assim como noções de performance.



# Scriptable Objects

# Scriptable Object

Classe base da Unity

- Objetos **serializáveis**
- Armazenar dados compartilhados entre objetos
- Não faz parte de um game object → Salvos como assets
- Podem ser referenciados por outros objetos

**Serializar:** transformar uma estrutura de dados em um formato armazenável

```
using UnityEngine;
```

```
[CreateAssetMenu(menuName="MySO")]  
public class MySO : ScriptableObject  
{  
    public int variable;  
}
```



# Comportamentos

Diferente de MonoBehaviour:

- Não pertence a um game object
- Não faz parte da hierarquia da cena
- Possui menos callbacks, com comportamentos diferentes (ex. Update)

Editor vs Build

- Alterações no Play mode são salvas
  - Alterações em builds não são salvas
- Não utilizar para save games

# Uso 1: Data container

*“Armazenar dados compartilhados entre objetos”*

- Evitar dados duplicados, ou que não precisam pertencer a cena
- **Dados que não serão modificados entre as seções de jogo**

Arquitetura data-driven

- Alterável apenas modificando os dados (sem código)
- Dar maior liberdade para o designer modificar o jogo

Espada1

Nome  
Ataque  
Durabilidade

Espada2

Nome  
Ataque  
Durabilidade

Escudo1

Nome  
Ataque  
Durabilidade

# Uso 2: Extendable Enum

Enum: conjunto finito de identificadores

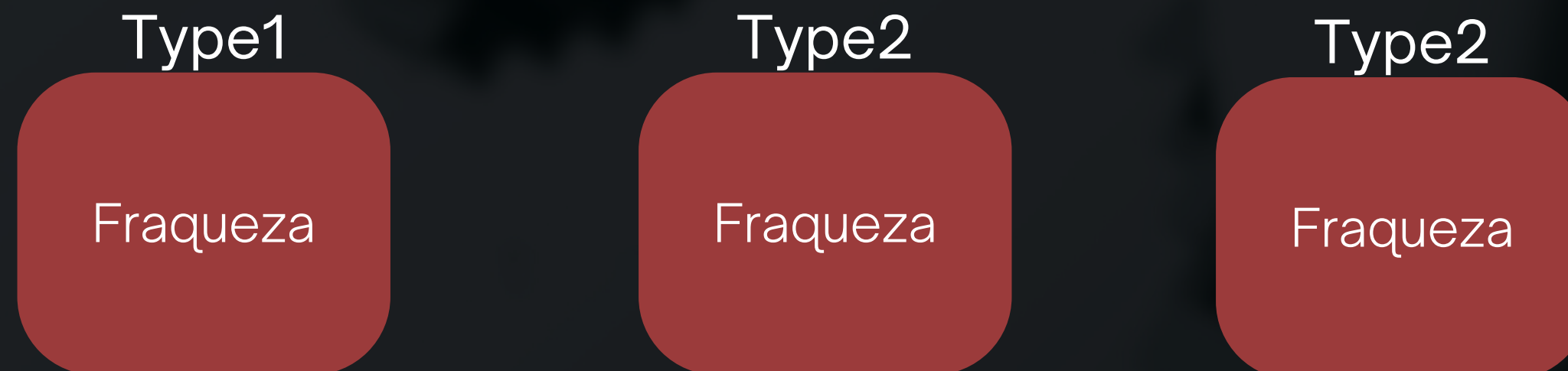
- Ex: tipos de ataques (gelo, fogo, etc)

Problema:

- Designer pode desejar adicionar mais identificadores durante o desenvolvimento
- Alterar o código a cada novo valor é indesejável

Solução:

- Utilizar Scriptable Objects para armazenar os identificadores





# Uso 3: Variáveis

Problema:

- Muitos objetos podem precisar ler ou escrever em valores comuns
- Ex: vida do jogador

Solução:

- Criar “variáveis” utilizando scriptable objects

Cria **desacoplamento**: possível criar objetos que interagem indiretamente entre si





# Uso 4: Observer/Events

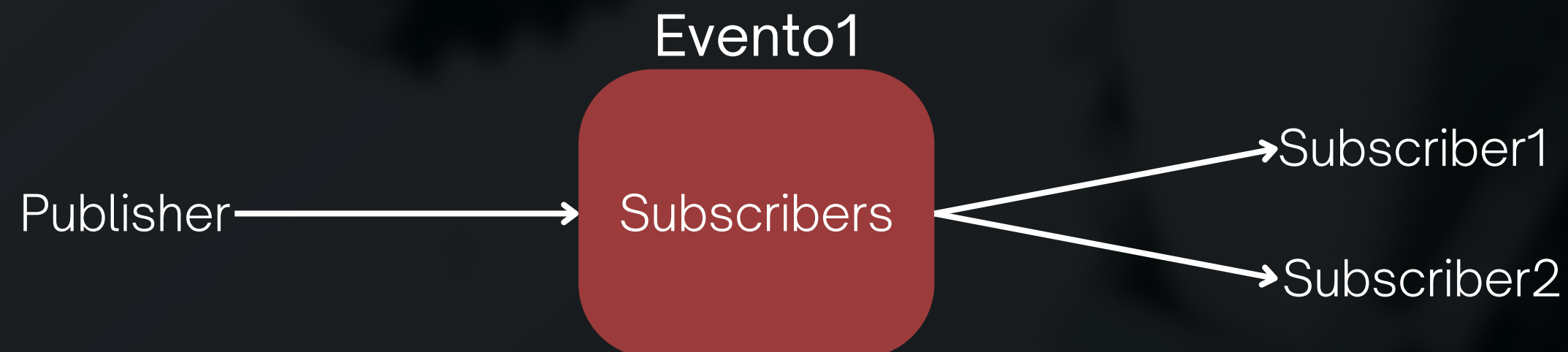
Problema:

- Muitos objetos precisam saber sobre a ocorrência de um evento
- Ex: pausar o jogo

Solução:

- Criar “eventos” utilizando scriptable objects

+**desacoplamento**: objetos que lança e escuta um evento não interagem diretamente



# Desacoplamento

Isolar códigos que executam tarefas diferentes

- Evitar dependências entre objetos

## **Prefabs devem ser isolados**

Inserir na cena não deve depender de outros n prefabs

### **Variáveis**

Desacoplar dados

### **Observer**

Desacoplar eventos



# Navmesh

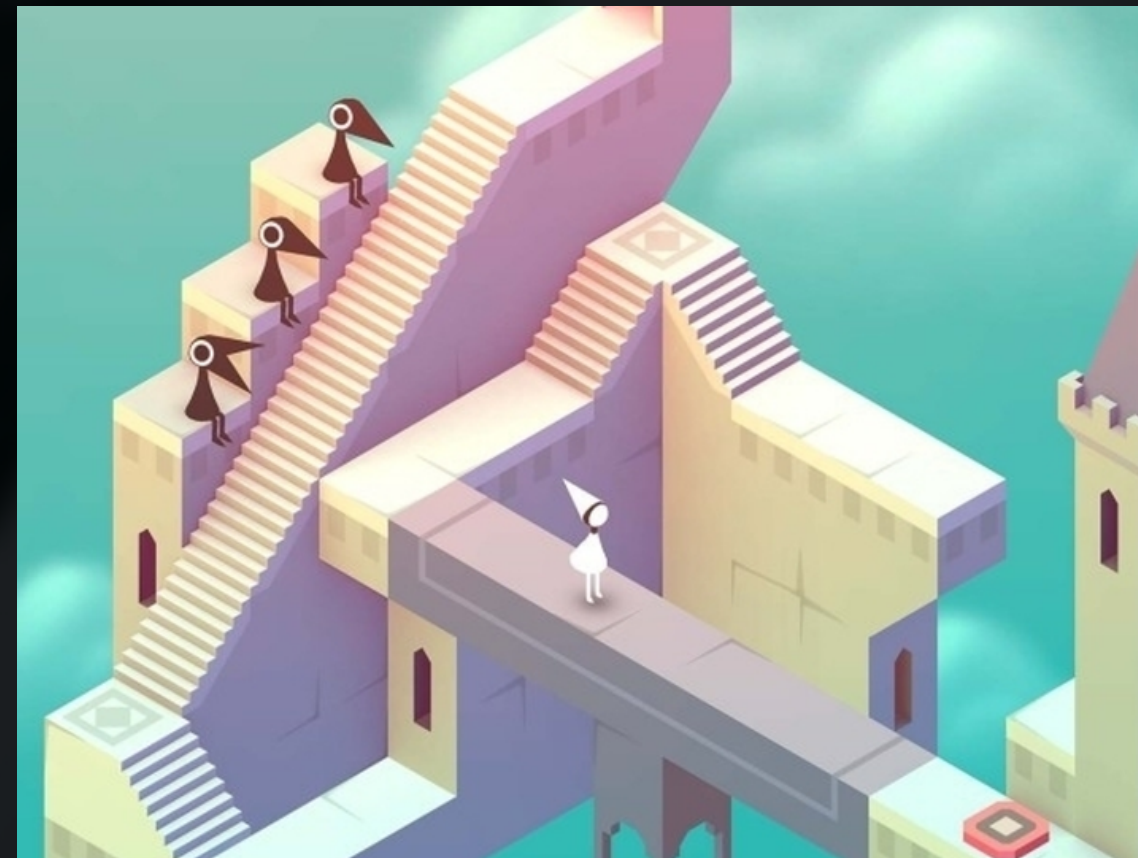


# Sistema de navegação

Permite fazer com que objetos da cena andem de um ponto para outro

- Navmesh: superfície pela qual os agentes podem navegar
- Superfície precisa ser calculada, preferencialmente antes do jogo

Pacote “AI Navigation”



Monument Valley, Ustwo Studio Ltd.





# Perfomance

# Importância

Permitir que mais pessoas joguem o jogo

- Possibilitar que o jogo exista (ex: Mobile, Console)
- Ampliar mercado

Windows, Mac and Linux ▼ Podes usar este menu para filtrares por plataforma.

September 2023 (clica em cada item para veres mais detalhes)

ITEM	MAIS POPULAR	PERCENT.	DIFERENÇA
Versão do SO	Windows 10 64 bit	57.97%	+1.99%
System RAM	16 GB	51.33%	+0.41%
Intel CPU Speeds	2.3 Ghz to 2.69 Ghz	21.28%	+0.52%
Physical CPUs	6 cpus	34.99%	+2.74%
Video Card Description	NVIDIA GeForce RTX 3060	6.10%	+1.37%
VRAM	8 GB	31.11%	+1.26%
Primary Display Resolution	1920 x 1080	61.17%	+0.42%
Multi-Monitor Desktop Resolution	3840 x 1080	60.96%	+0.26%
Language	English	32.33%	-3.24%
Free Hard Drive Space	100 GB to 249 GB	22.92%	-0.86%
Total Hard Drive Space	Above 1 TB	55.26%	+3.58%
VR Headsets	Oculus Quest 2	41.19%	-0.76%
Other Settings	LAHF / SAHF	100.00%	0.00%

Inquérito de Hardware e Software  
do Steam: September 2023

# Métricas importantes

- Armazenamento: quanto espaço o jogo ocupa
- Memória (RAM)
- Memória gráfica (VRAM)
- FPS: quantos frames são produzidos em 1 segundo
  - 30 fps: mínimo
  - 60 fps: ideal
  - Consistência: importante que o FPS não varie muito

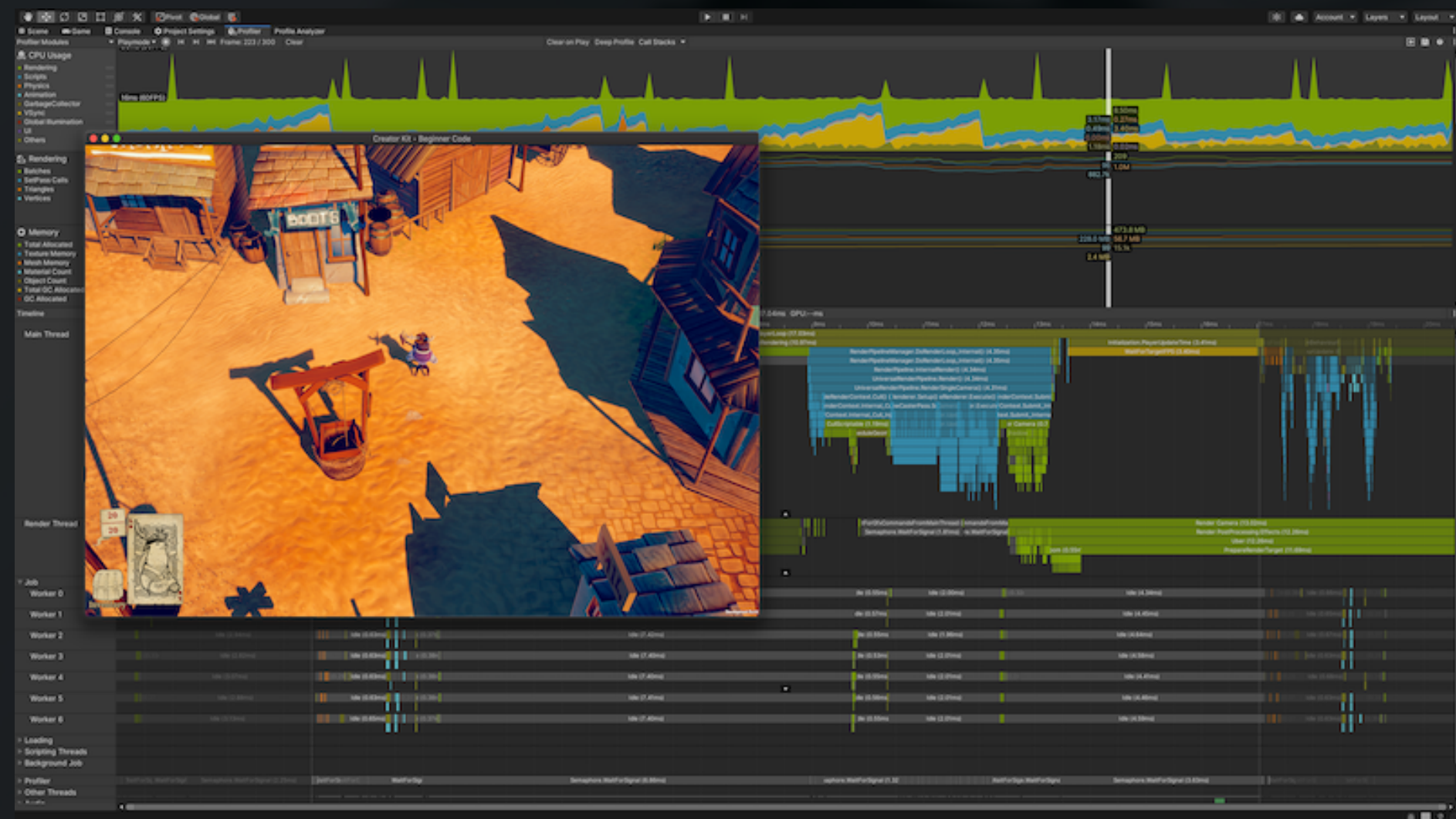
FPS baixo, mas o que otimizar?



# Profiler

Ferramenta para obter métricas do jogo

- Frame a frame
- Avaliar custo de funções



Unity





**Próximos passos**

# O que fazer agora?

Avaliar a arquitetura do projeto sendo criado

- Possui muito acoplamento entre os objetos/prefabs?
- É possível utilizar Scriptable Objects para melhorar a organização?

Avaliar a performance do projeto

- Qual o hardware final?
- A performance é adequada?
- Existem detalhes sendo utilizados em lugares desnecessários