Data Science :
Persistência de preferências de jogador

Prof. Ovídio Francisco

Plano de aula

Persistência de dados em jogos

A Classe PlayerPrefs

Métodos

Exemplo de uso

Exercícios

Introdução

O contexto

Normalmente um game possui suas **configurações** que podem ser modificadas pelo **jogador**, como dificuldade, controles, som e ranking.

Essas configurações podem ser **salvas** de forma que seus dados fiquem disponíveis mesmo após o jogo encerrar. Para que esses dados fiquem **gravados no dispositivo**, é preciso alguma estratégia de **persistência**.

Nessa aula, veremos como usar a classe **PlayerPrefs** para esse propósito.

A classe PlayerPrefs

PlayerPrefs

PlayerPrefs é uma classe desenvolvida para salvar e carregar dados entre seções de jogos. Ela utiliza recursos do sistema operacional para gravar e acessar os dados localmente, como o registro do Windows.

É uma classe estática que fornece métodos amigáveis para definir e resgatar dados no formato chave-valor, semelhante a um dicionário ou tabela hash.

https://docs.unity3d.com/ScriptReference/PlayerPrefs.html

Chave-valor

Dados armazenados no forma chave-valor podem ser entendidos como na tabela abaixo:

Chave	Valor
“Nome”	“Jurema”
“Vidas”	3
“Dificuldade”	“Fácil”
“Volume do Audio”	0.78
“Dinheiro”	89.45
“Pontos”	259
“Inimigos abatidos”	72

PlayerPrefs

Possui suas vantagens e desvantagens:

Prós

- É um classe própria do Unity. Não precisa de instalação de pacotes ou bibliotecas.
- Muito fácil de usar.
- Formato de dados simples.

Contras

- Restrito a valores int, float e string
- Dados locais

PlayerPrefs

É utilizada principalmente para:

- Configurações de preferências do jogador
- Dados de configurações de desempenho
- Configurações de execução do jogo
- Dados simples sobre o estado do jogo
- Pontuações

Métodos

Usa métodos estáticos (não necessitam de instâncias de objetos) para salvar e carregar dados.

As sintaxes dos métodos para salvar (definir) valores seguem o seguinte padrão:

Set*Tipo*(“Chave”, valor);

Exemplo:

```
PlayerPrefs.SetInt(“Vidas”, 5);
PlayerPrefs.SetFloat(“Saúde”, 0.15f);
PlayerPrefs.SetString(“Idioma”, “en-uk”);
```

Métodos

As sintaxes dos métodos para salvar (definir) valores seguem o seguinte padrão:

Get*Tipo*(“Chave”, valorPadrao);

Caso a chave “Vidas” ainda não esteja definida, o valor retornado será 1.

Exemplo:

```
int vidas = PlayerPrefs.GetInt(“Vidas”, 1);
float saude = PlayerPrefs.GetFloat(“Saúde”, 1.0f);
String idioma = PlayerPrefs.GetString(“Idioma”, “pt-br”);
```

Mais Métodos

Os métodos disponíveis em PlayerPrefs são:

Método	Descrição
`SetInt`	Define um int
`SetFloat`	Define um float
`SetString`	Define uma string
`GetInt`	Carrega um int
`GetFloat`	Carrega um float
`GetString`	Carrega uma string
`HasKey`	Testa se uma chave está definida
`Save`	Salva todos os dados imediatamente
`DeleteKey`	Deleta uma chave
`DeleteAll`	Deleta todas as chaves

Para saber mais...

https://docs.unity3d.com/ScriptReference/PlayerPrefs.html

https://medium.com/@zachcaceres/persistent-data-between-scenes-how-to-use-playerprefs-in-unity-b6fd409363c3

https://docs.unity3d.com/2020.1/Documentation/ScriptReference/PlayerPrefs.html

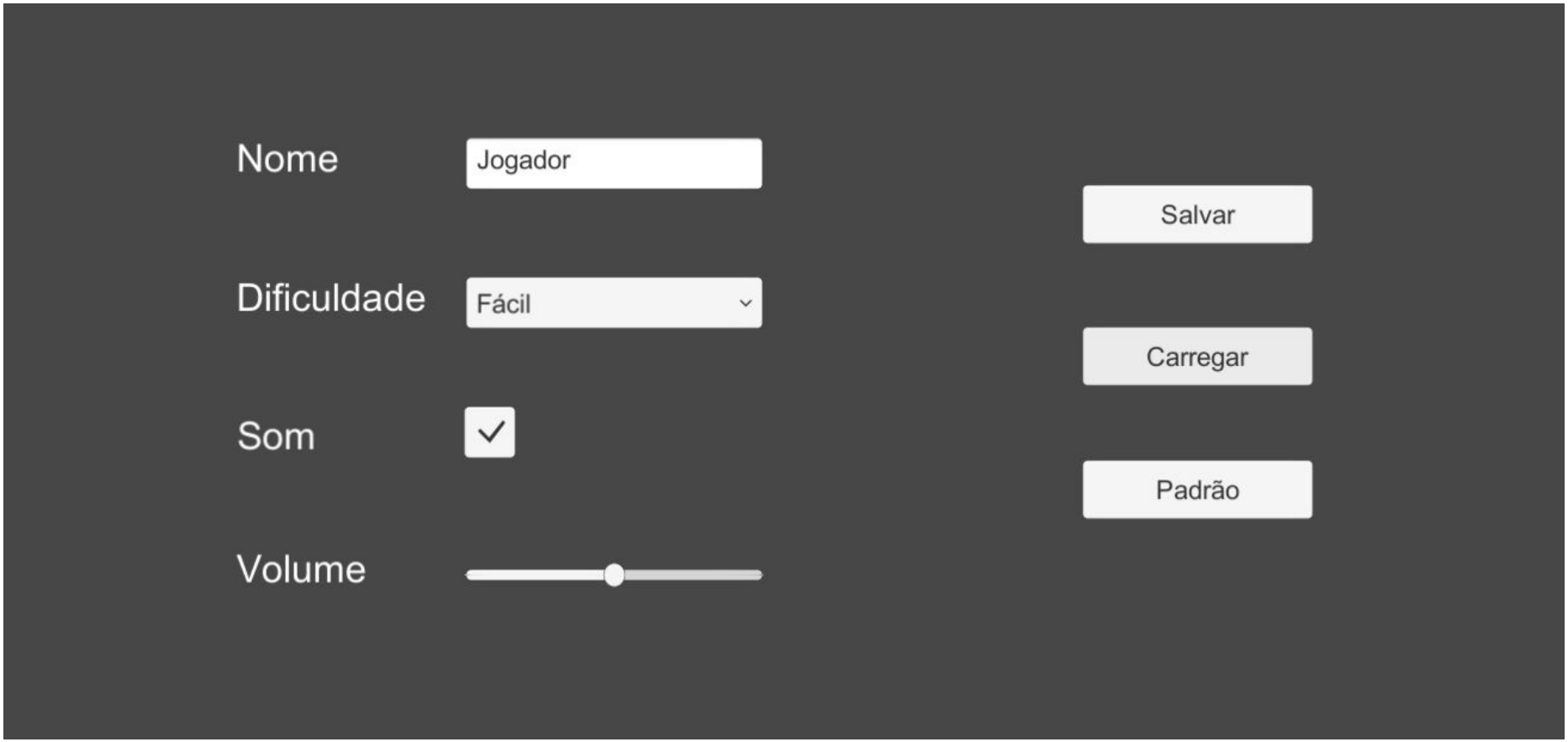
https://www.coursera.org/lecture/intermediate-object-oriented-programming-unity-games/playerprefs-7cmAs

Mais Métodos

Os métodos disponíveis em PlayerPrefs são:

Método	Descrição
void SetInt(string key , int value);	
void SetFloat(string key , float value);	
void SetString(string key , string value);	
int GetInt(string key , float default);	
float GetFloat	
string GetString	
bool HasKey	
void Save	
void DeleteKey	
void DeleteAll	

O contexto


Nome
Jogador
Dificuldade
Fácil
Som
Volume
Salvar
Carregar
Padrão