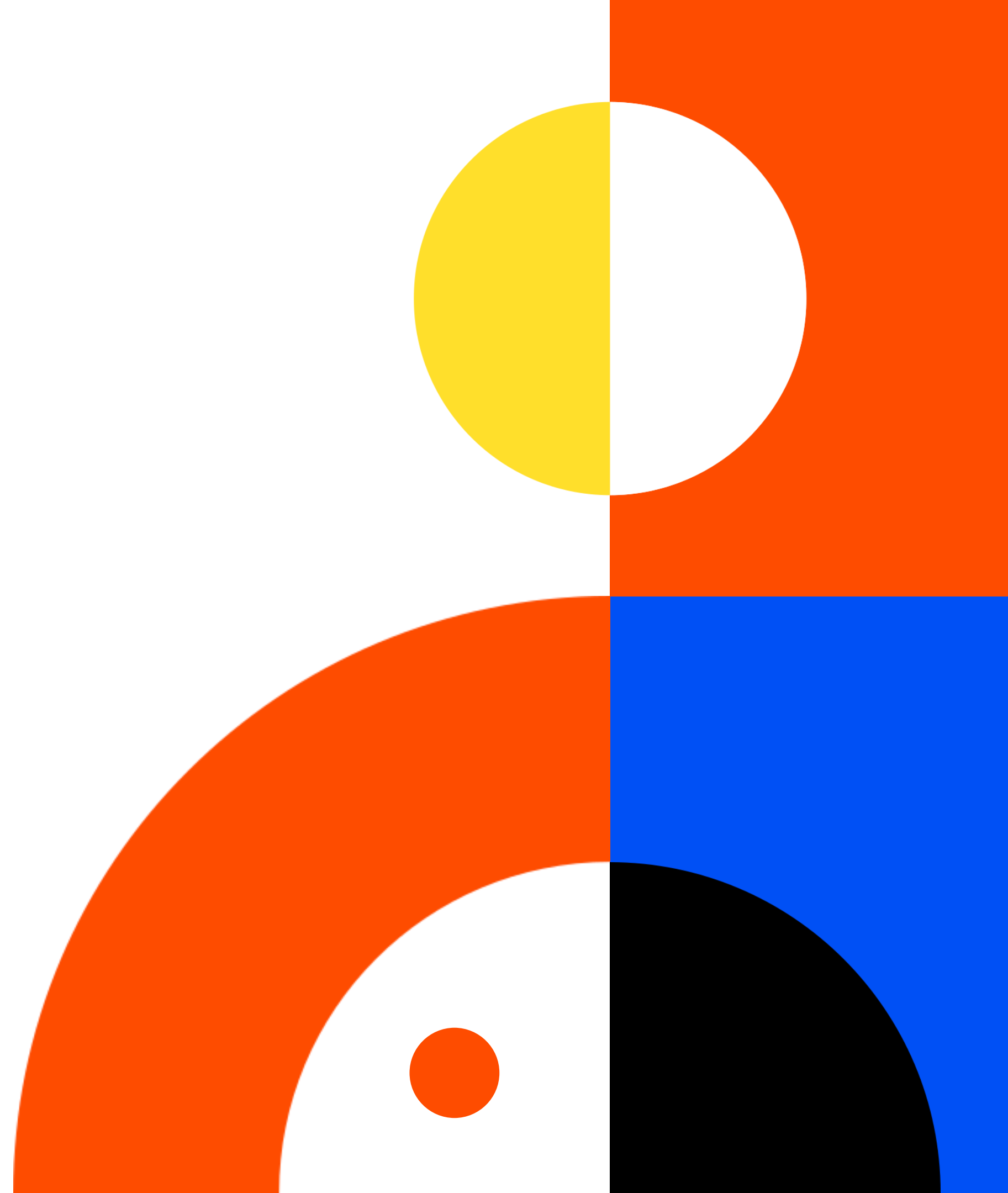
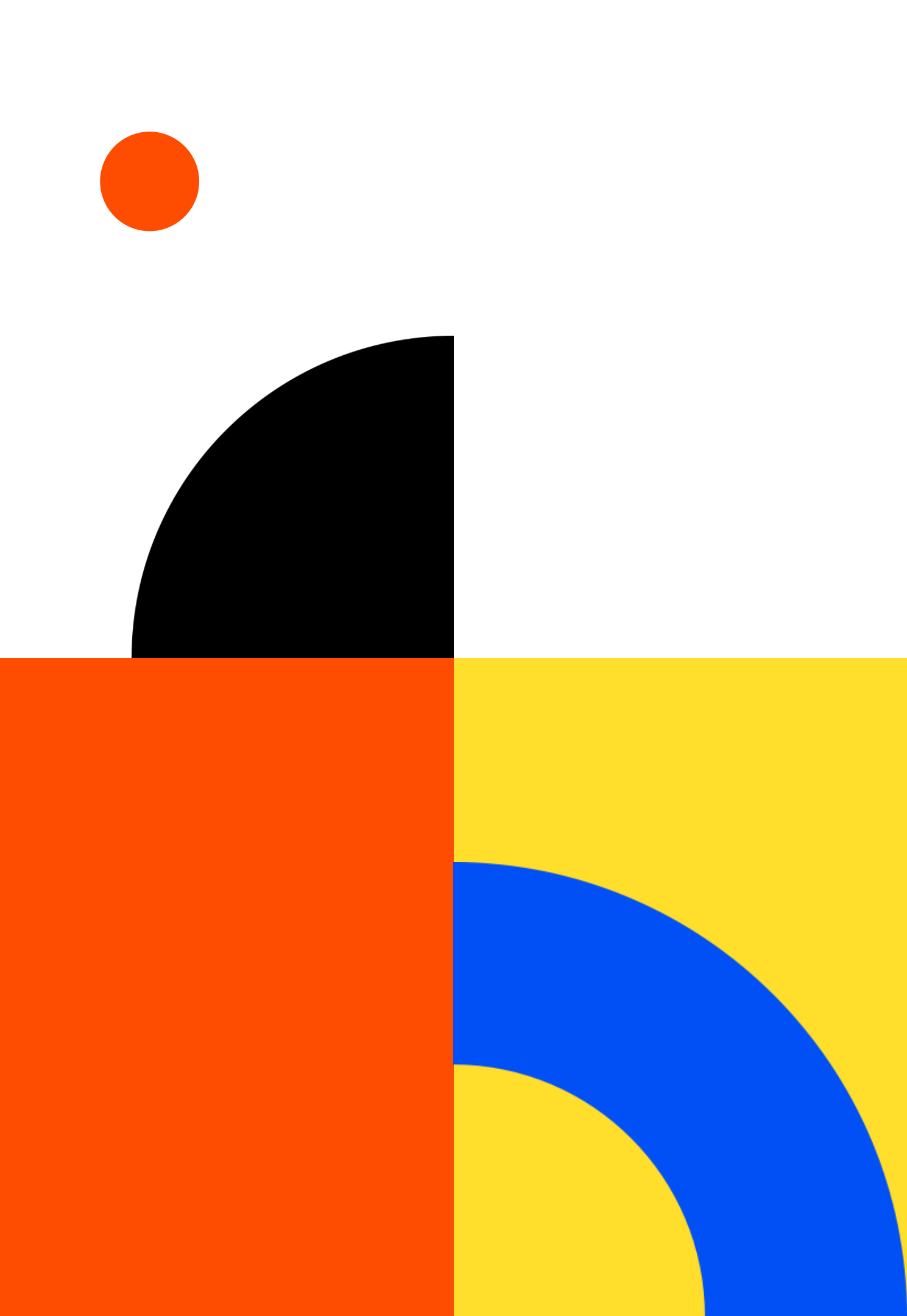


Лекція №8

Збереження даних

2025





План

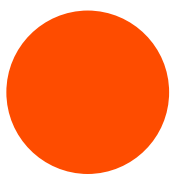
1. Введення
2. PlayerPrefs
3. Збереження в файл
Application.persistentDataPath
4. Newtonsoft Json.Net



Збереження даних у Unity

У більшості ігор гравець взаємодіє зі світом, робить прогрес, змінює налаштування або досягає певних результатів.

Щоб ці зміни **зберігалися між запусками гри**, потрібна **система збереження даних** (*Data Persistence System*).



Що потрібно зберігати

Тип даних	Приклад
 Ігровий прогрес	рівень, кількість очок, інвентар
  Налаштування	гучність, роздільна здатність, управління
 Дані користувача	ім'я, профіль, аватар
 Стан світу	положення об'єктів, виконані місії

PlayerPrefs

Призначення: збереження простих даних (int, float, string).

Місце зберігання: системний реєстр (Windows) або спеціальні файли (Android, iOS).

// Save player score

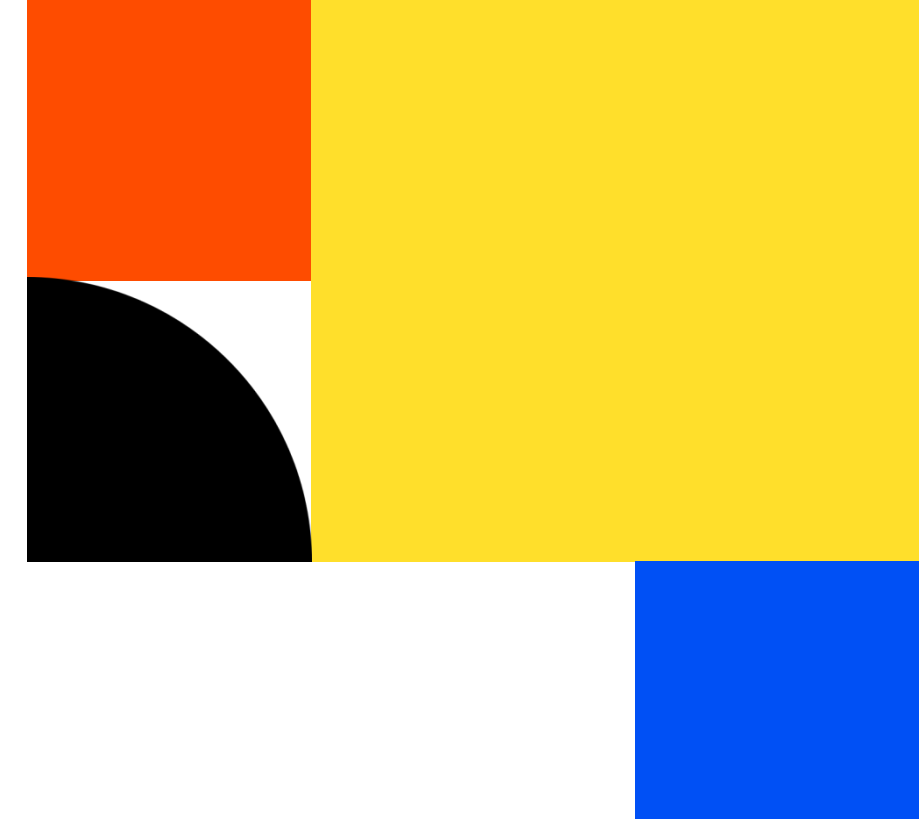
```
PlayerPrefs.SetInt("score", 150);
```

```
PlayerPrefs.Save(); // force save
```

// Load player score

```
int score = PlayerPrefs.GetInt("score", 0);
```

```
Debug.Log("Score: " + score);
```

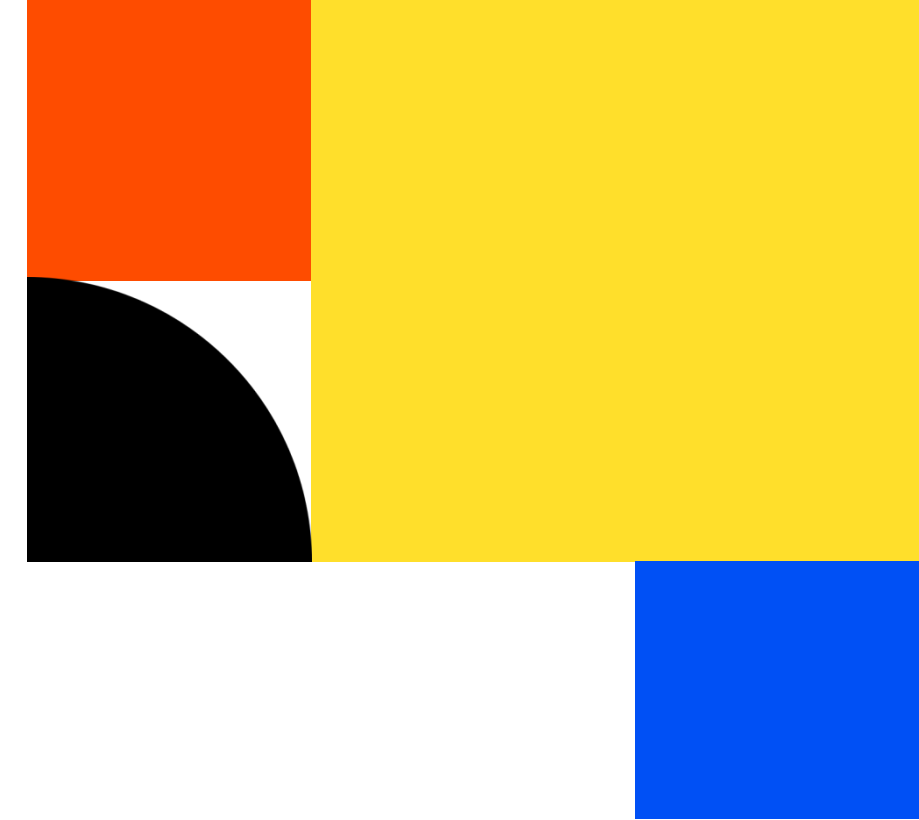


Переваги:

- Простий у використанні.
- Автоматично кросплатформений.

Недоліки:

- Зберігає лише прості типи.
- Дані можна легко змінити користувачем (немає безпеки).

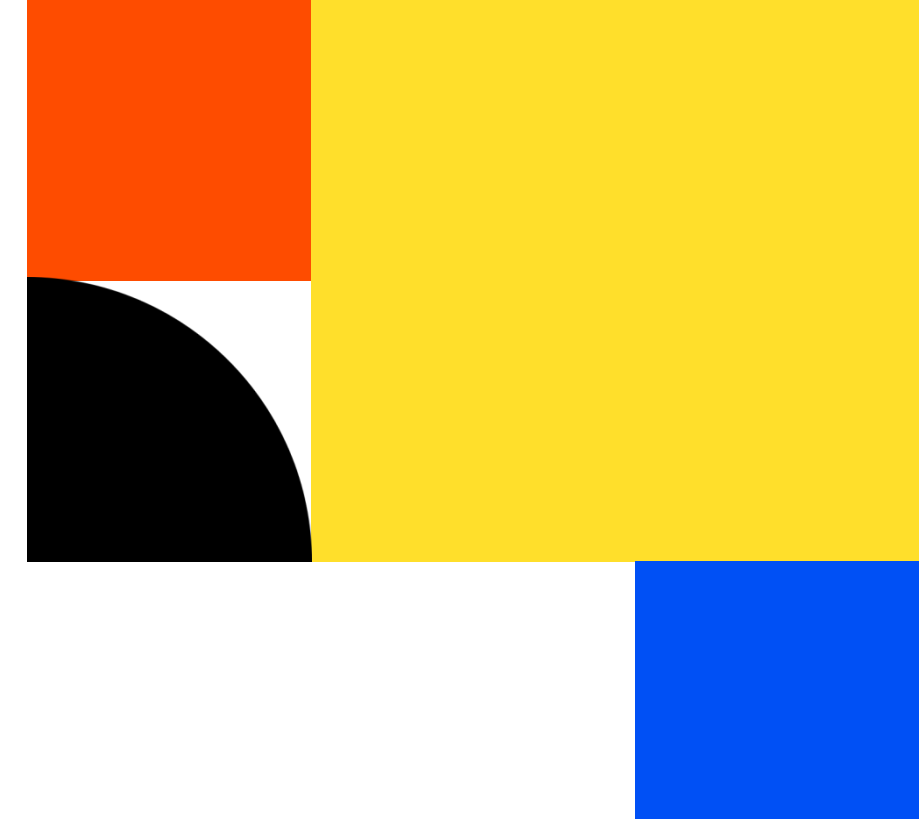


Standalone player and in-Editor Play mode storage location

- **MacOS:** ~/Library/Preferences/com.ExampleCompanyName.ExampleProductName.plist

Standalone Player storage location

- **Android:** /data/data/pkg-name/shared_prefs/pkg-name.v2.playerprefs.xml.
- **iOS:** Uses the [NSUserDefaults standardUserDefaults] API to store PlayerPrefs data.
- **Linux:** ~/.config/unity3d/ExampleCompanyName/ExampleProductName
- **WebGL:** Unity stores up to 1MB of PlayerPrefs data using the browser's IndexedDB API.
- **Windows:**
Computer\HKEY_CURRENT_USER\Software\ExampleCompanyName\ExampleProductName in the Registry Editor.
- **Windows Universal Platform:**
%userprofile%\AppData\Local\Packages\[ProductPackageId]\LocalState\playerprefs.dat

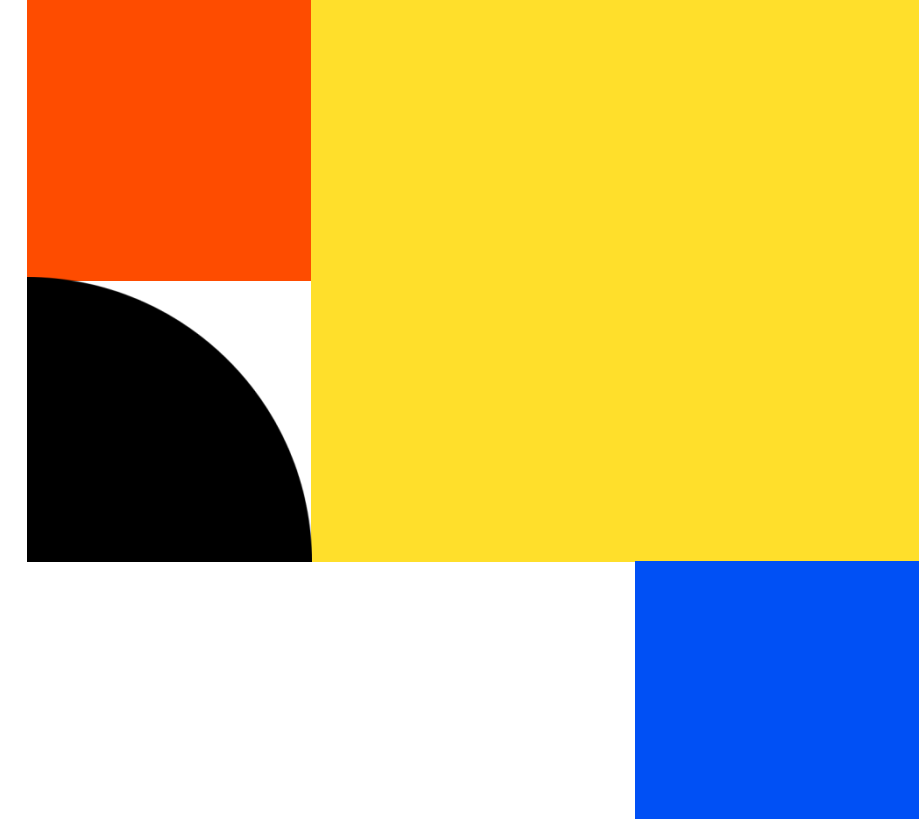


Збереження в файл

Для збереження в файл можна використовувати стандартні можливості C# для зберігання файлів

```
File.WriteAllText(path, data);
```

Для кросплатформенної розробки потрібно використовувати спеціальну папку, яку юніті віддає через спеціальний параметр **Application.persistentDataPath**



UWP Apps: Application.persistentDataPath points to C:\Users\<user>\AppData\LocalLow\<company name>.

Windows Editor and Windows Player: Application.persistentDataPath usually points to %userprofile%\AppData\LocalLow\<companyname>\<productname>. It is resolved by SHGetKnownFolderPath with FOLDERID_LocalAppDataLow, or SHGetFolderPathW with CSIDL_LOCAL_APPDATA if the former is not available.

WebGL: Application.persistentDataPath points to /idbfs/<md5 hash of data path> where the data path is the URL stripped of everything including and after the last '/' before any '?' components.

Linux: Application.persistentDataPath points to \$XDG_CONFIG_HOME/unity3d or \$HOME/.config/unity3d.

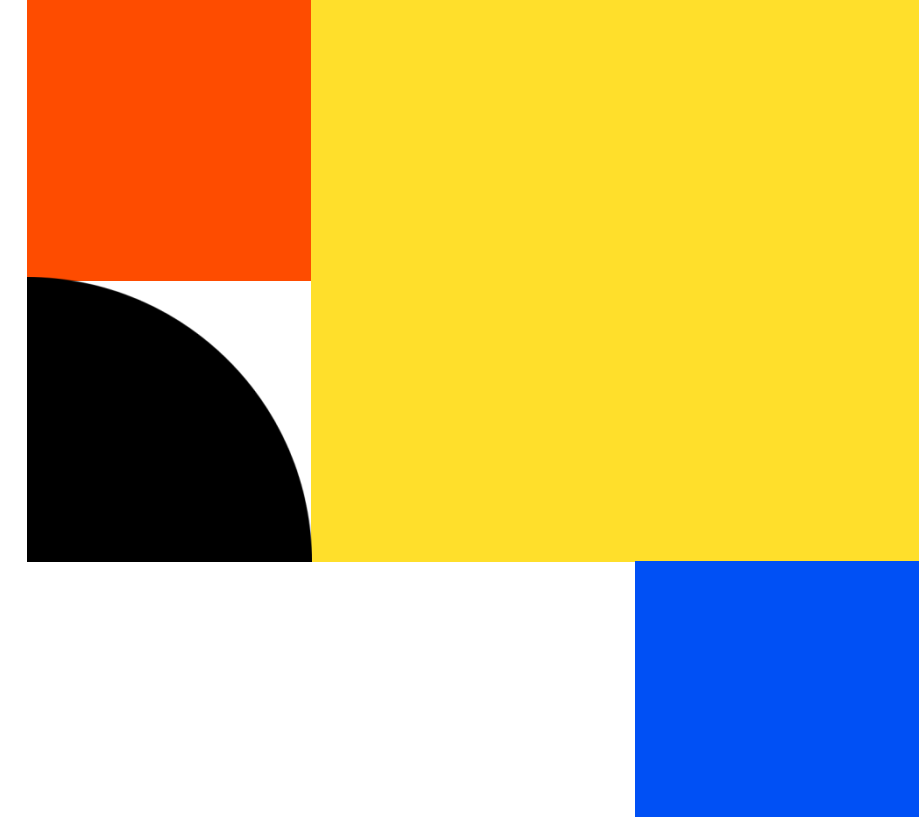
iOS: Application.persistentDataPath points to /var/mobile/Containers/Data/Application/<guid>/Documents.

tvOS: Application.persistentDataPath is not supported and returns an empty string.

Android: Application.persistentDataPath points to /storage/emulated/<userid>/Android/data/<packagename>/files on most devices (some older phones might point to location on SD card if present), the path is resolved using android.content.Context.getExternalFilesDir.

macOS Editor: Application.persistentDataPath points to ~/Library/Application Support/company name/product name.

macOS Player: Application.persistentDataPath points to ~/Library/Application Support/unity.company name.product name, but uses the Editor path if that directory already exists.



Newtonsoft.Json.Net

Unity має вбудований клас **JsonUtility** для серіалізації об'єктів у JSON, але він **обмежений** (не підтримує словники, наслідування, масиви складних типів тощо).

Для більш гнучких задач використовується **JSON.NET (Newtonsoft.Json)** — популярна бібліотека з .NET, яку Unity підтримує як **зовнішній пакет або NuGet-пакет**.

Основна мета:

- Гнучка серіалізація будь-яких об'єктів.
- Підтримка словників, колекцій, наслідування, атрибутів, форматів дат, нульових значень тощо.

Щоб додати Json.Net в Unity потрібно в Package Manager додати пакет з ім'ям *com.unity.nuget.newtonsoft-json*.

```
// Serialize object to JSON string  
string json = JsonConvert.SerializeObject(player);
```

```
// Deserialize JSON back to object  
PlayerData loaded = JsonConvert.DeserializeObject<PlayerData>(json);
```

