

ЛОКАЛЬНОЕ СОХРАНЕНИЕ ИГРОВЫХ ДАННЫХ С ПОМОЩЬЮ СЕРИАЛИЗАЦИИ

Сериализация представляет собой процесс автоматического преобразования структур данных либо состояний объекта в иной формат. Это один из приемов повышения продуктивности разработки. Сериализация является важной технологией для сохранения игрового прогресса и настроек на локальном устройстве. Она позволяет игрокам сохранять свой прогресс и настройки, что делает игровой процесс более удобным и интересным.

Таким образом, при запуске игры, необходимо проверить наличие сохраненных данных и восстановить их, если они есть. В Unity можно использовать следующие форматы: бинарный, xml, JSON. Для каждого формата предусмотрен свой класс: для сериализации в бинарный формат – класс `BinaryFormatter`, для xml – `XmlSerializer`, для JSON – `JsonSerializer`. Для сохранения данных необходимо определить класс, который будет содержать все нужные поля. Например, для сохранения даты сохранения, количества пройденных уровней, количества игровых монет и порядкового номера модели персонажа можно создать класс, приведенный в листинге 1.

```
[System.Serializable] public class GameData
{
    public string saveData;
    public int levelsPassed;
    public int coins;
    public int characterModel;}

```

Листинг 1 – Класс, содержащий поля для сохранения данных

Использование сериализации для локального сохранения игровых данных может улучшить игровой опыт игроков, позволяя им продолжать игру с того места, где они остановились.

Метод `JsonUtility.ToJson` используется для сериализации объекта в формат JSON. Затем полученная строка записывается в файл на устройстве пользователя. Чтобы использовать сохраненные данные в игре необходимо получить их в исходном виде. Метод сначала проверяет, существует ли файл сохраненных данных на устройстве пользователя. Если файл существует, то мы читаем его содержимое в строку и десериализуем объект `GameData` с помощью метода `JsonUtility.FromJson`. Если файл не существует, то метод возвращает `null`.