УДК 004.4

ИНСТРУМЕНТ КОНВЕРТАЦИИ ФАЙЛОВ В JSON ФОРМАТ

***К.П. Резниченко**, 1 курс*
*Научный руководитель – **Ю.М. Вишняков**, к.т.н., доцент*
Полесский государственный университет

Разработка игр — это процесс, которым может заниматься как один человек, так и целая компания. В некоторых компаниях принято брать сотрудников на удаленную работу, такие люди не занимают место в офисе и могут работать из дома в привычной для них обстановке. Над разработкой игр большого класса (например, AAA) работает большое количество людей, которые находятся в подчинении у менеджеров проекта. Менеджер проекта контролирует работу своих подчиненных и является управляющим звеном в команде. Он выдает разработчикам задачи и информирует их о выполнении плана.

Во многих проектах, для выполнения которых привлекаются удаленные сотрудники, есть сервер, с помощью которого можно настроить доступ всем участникам команды и хранить проект централизованно. С помощью системы контроля версий производится синхронизация файлов проекта на сервере, а также отслеживаются изменения, внесенные разработчиками в файлы проекта. Система контроля версий Subversion [1] — свободная централизованная система управления версиями, официально выпущенная в 2004 году компанией CollabNet. Она является бесплатной и обладает хорошим инструментарием, который облегчает работу с удаленным проектом. К примеру, частью инструментария по отслеживанию изменений является возможность сравнения текущего содержимого файла с содержимым этого файла предыдущей версии. Выглядит это как открытый текстовый редактор с пометками в изменении между двумя копиями файлов.

Для отслеживания изменений в текстовых файлах этого инструментария более чем достаточно, но файлы игры не обладают текстовой читабельной информацией и различия между разными версиями практически не могут быть выявлены человеком. Для оптимизации процесса проверки в проекте было решено с помощью возможностей игрового движка и среды разработки создать инструментарий, способный считывать информацию об объектах игрового мира и конвертировать ее в текстовый файл формата JSON, после чего с помощью инструментария Subversion проверять различия между разными версиями файлов. Для разработки подобного инструментария необходимо знать в первую очередь возможности игрового движка и структуру файлов формата JSON. С помощью документации можно узнать, обладает ли игровой движок встроенными возможностями конвертации данных в необходимый нам формат или же придется создавать свой класс, который сможет с помощью методов создавать строку в JSON–формате, которую впоследствии можно будет записать в текстовый файл. Движок, о котором сейчас идет речь – это Unreal Engine 4 [2] и он обладает возможностью конвертировать данные в формат JSON, состоящий из объектов, записей и массивов. Unreal Engine 4 – игровой движок, разрабатываемый и поддерживаемый компанией Epic Games с 2014 года.

Файл формата JSON выглядит следующим образом:

```
"ObjectName":{
      "Field1": "Value1",
        "Field2": "Value2",
      "ArrayName": ["ArrVal1", "ArrVal2"] }
```

Для создания основы инструмента, который будет расширяться на другие файлы движка, необходимо воспользоваться предоставляемым классом TJsonWriterFactory. С помощью данного класса–фабрики можно создать объект класса TJsonWriter и воспользоваться набором специальных методов, которые способны создавать объекты, массивы и прочие элементы структуры JSON–файла и сразу конвертировать их в строку. Полученную строку необходимо будет выводить в текстовый файл.

Для начала создадим наш объект TJsonWriter следующим образом:

```
FString Stream;
TSharedRef<TJsonWriter<>> Writer = TJsonWriterFactory<>::Create(&Stream);
```

Теперь у нас есть переменные Writer и Stream. Writer – переменная объекта класса TJsonWriter, с помощью которого мы будем заполнять нашу строку (Stream) объектами, массивами и прочими элементами структуры JSON–файла.

У данного объекта есть следующие методы, которые помогут нам в конвертации файла в JSON–формат:

```
Writer–>WriteObjectStart();      //Начать запись объекта
Writer–>WriteObjectEnd();        //Закончить запись объекта
Writer–>WriteValue("Name of value","Value"); //Записать значение
Writer–>WriteArrayStart("ArrayName"); //Начать запись массива
Writer–>WriteArrayEnd(); //Закончить запись массива
```

Теперь зная необходимые нам возможности игрового движка для конвертации данных карты в JSON формат необходимо воспользоваться следующим кодом:

```
if(UWorld* world = GetWorld()){

// Получаем все объекты с уровня(части игрового мира)
TArray<AActor*> Actors;
UGameplayStatics::GetAllActorsOfClass(world, AActor::StaticClass(), Actors);

//Создаем JSON объект, с названием уровня
Writer–>CreateObjectStart(world–>GetName());

//Перебираем все объекты с уровня и записываем их базовую информацию
for (AActor* curActor : Actors){
Writer–>CreateObjectStart(curActor–>GetName());
Writer–>WriteValue("Position", curActor–>GetActorLocation());
Writer–>WriteValue("Rotation", curActor–>GetActorRotation());
Writer–>WriteValue("Scale", curActor–>GetActorScale3D());
Writer–>CreateObjectEnd();
}
Writer–>CreateObjectEnd();

//Записываем строку в файл
FFileHelper::SaveStringToFile(Stream, TEXT("d:\\Map\\MapName.json"));
}
```

Разработанный продукт является инструментом с открытым исходным кодом, доступным для модификации, который в дальнейшем может расширяться на другие форматы файлов проекта данного игрового движка.

Список использованных источников

1. Subversion — Википедия [Электронный ресурс] / Википедия. Режим доступа: https://ru.wikipedia.org/wiki/Subversion

2. Unreal Engine — Википедия [Электронный ресурс] / Википедия. Режим доступа: https://ru.wikipedia.org/wiki/Unreal_Engine#Unreal_Engine_4