**Serializer Module**

Кастомный модуль сериализации и загрузки игровых данных в движке Unigine

**Основные компоненты**

|  |  |
| --- | --- |
| SerializerBase.cs | базовый класс сериализации |
| SerializeBaseItem.cs | базовый класс объекта сериализации |
| SerializeHandler.cs | обработчик сериализации в игре |
| ISProvider.cs | интерфейс доступа к базовому модулю |
| ISDisabler.cs | интерфейс отключаемых элементов |

**SerializerBase.cs**

Сериализатор инкапсулирует работу по созданию, открытию, чтению, изменению файлов сохранения состояний игрового мира. Предоставляет API для безопасного сохранения/загрузки игровых данных всех имеющихся узлов (Unigine.Node) игрового мира (Unigine.World). Базово сохраняются идентификатор объекта, идентификатор родительского объекта (если есть), имя, параметры положения, поворота, размера, матрица трансформации, а также, если указанный узел является объектом (Unigine.ObjectNode), сохраняется параметр auxiliary материалов всех его поверхностей.

Класс имеет следующие базовые настройки

|  |  |  |
| --- | --- | --- |
| **Параметр** | **По умолчанию. Тип** | **Описание** |
| mode | WRITE\_NEW\_DATA <enum MODE> | Параметр выбора режима работы. Базовые режимы:  - перезапись новых данных  - загрузка сохраненного ранее состояния |
| folderRoot | DATA  <enum ROOT> | Выбор корня сохранения файлов:  ../data/ или ../bin/ |
| saveStateFolder | “serialization“ <String> | Папка в корне для размещения сохранений |
| useTimeStamp | false <bool> | Добавляет в название файла сохранения штамп текущего времени в формате  год-месяц-число-час-минута-секунда |
| saveDefaultDataState | false <bool> | Включение сохранения при первичной загрузке |
| saveDataOnShutdown | false <bool> | Включение сохранения при завершении работы |

Экземпляр класса или его наследник может быть создан как внутри конкретного игрового мира (в этом случае он будет сохранять данные только мира где подключен), так и внутри AppSystemLogic.cs

В случае подключения в AppSystemLogic.cs экземпляр класса объявляется статическим, и в каждом игровом мире, использующий сериализацию данных, должен обладать обработчиком, реализующим функционал интерфейса ISProvider.cs В данном случае подключения, экземпляр класса-сериализатора будет работать во всём времени исполнения программы и сохранять данные в каждом из активных игровых миров.

Пример объявления и инициализации экземпляра класса в AppSystemLogic.cs

namespace UnigineApp

{

class AppSystemLogic : SystemLogic

{

static private SerializerItems serializer; // статический сериализатор

…

static public SerializerItems GetStaticSerializer() // метод получения ссылки на экземпляр

{

return serializer;

}

public override bool Init()

{

serializer = new SerializerItems(); // инициализируем статический сериализатор

return true;

}

...

**SerializeBaseItem.cs**

Сериализатор инкапсулирует работу по созданию, открытию, чтению, изменению файлов сохранения состояний игрового мира. Предоставляет API для безопасного сохранения/загрузки игровых данных всех имеющихся узлов (Unigine.Node) игрового мира