The 8th International Scientific Conference
eLearning and software for Education
Bucharest, April 26-27, 2012
10.5682/2066-026X-12-153

SOLUTIONS FOR DEVELOPING SCORM CONFORMANT SERIOUS GAMES

Dragoş BĂRBIERU
National Defence University "Carol I", Panduri Street, Bucharest, Romania
dragos.barbieru@adlunap.ro

Daniel BELIGAN
National Defence University "Carol I", Panduri Street, Bucharest, Romania
daniel.beligan@adlunap.ro

Abstract: *Developing simulations, serious games, and other forms of training in a way that enables interoperability is one means of increasing the depth and scope of instructional materials available to learners while reducing overall development costs and time. Interoperability, the ability of computers and applications to communicate and share resources in a heterogeneous environment, is dependent on standards.*

Keywords*: e-learning, SCORM, serious games, Unity3D*

Programming games is practically different from other kinds of programming. Games probably offer more freedom than other types of programming projects because game code can have a very short life cycle. A SG is made up of three primary components:

- application layer - deals with the hardware and the operating system.
- game logic layer - manages your game state and how it changes over time, outputs from the game logic will be state changes and events, events and state changes are sent to game views.
- game view layer - presents the game state with graphics and sound.

The game state can be changed by external stimulus, such as a mouse, a keyboard, a gamepad key or an AI process. Every game must have a container for game objects. There are two kind of objects: one belongs to the game logic and is called an actor and other belongs to the renderer and is called a textured skeletal mesh. If the game state of actor changes, the game logic sends an event to renderer and it reacts to this event by changing the texture. To conclude the game logic holds the object state and the game view holds model data and textures. Serious games have more than just storyline, design and software. The pedagogy in this type of game plays a major role. For this reason we need a new component to check and report when the learning objectives have been met by learners. The added component is called game tracking layer.

The Game Logic Layer is the heart of the game. It defines the game, what things are inside and how they interact. The state of game can be changed by external devices. There are significant differences between the game logic representation of an object and the visual representation of an object. The game logic holds the object state and the visual representation managed by the game view holds model data and textures. The components of the game logic are:

- game state and data structures - here resides container for game objects. The game engine must be able to find quickly objects from object data structures to change an object's state, and it must be able to hold properties for each object. The game can use list structures, custom data structures, n-dimensional array, object structures etc..

- physics - defines everything from how objects move. The main role is to inject reality in games.
- events - an event-based architectures improves your game system to be robust and efficient. For solving the problems of communications between game subsystems and how they interact with game objects is necessary an well designed game event system. The best performance is obtained when game event system is global to the application and in this case it is a part of Game Application Layer. The game event system is organized into three parts: events and event data, event listeners and event manager.
- process manager - manages the list of processes.
- command interpreter - is responsible for interpretation of external commands.Is better to use an event-based interface instead of direct
- API calls to game logic code.

Real time 3D games have been around for well over ten years now. 3D has become affordable not only in the movie industry, as seen by the number of titles featuring CG (computer graphics), but also in the game industry where we’ve seen a shift in casual games from 2D to a 3D format.

Unity 3D is a new piece of technology that strives to make life better and easier for game developers. Unity is a game engine or a game authoring tool that enables creative folks like you to build video games.

By using Unity, you can build video games more quickly and easily than ever before.

You don’t have to be a programmer to make your own game with Unity, but you will need to be able to understand enough of what the scripts do to know what can be tweaked to your advantage or decide if a particular script will suite your needs. Most game play needs to be scripted in Unity, but there are hundreds of scripts already available that can be readily reused. Unity's primary and most astonishing selling point is that it can deliver a full 3D game experience right inside your web browser. It does this with the Unity Web Player- a free plugin that embeds and runs Unity content on the Web. Unity's development environment runs on Microsoft Windows and Mac OS X, and the games it produces can be run on Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, iPhone, as well as the Android platform. Unity supports integration with 3ds Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D, Cheetah3D, Photoshop and Allegorithmic Substance.

Unity contains all of the tools that you need to create a serious game. It has terrain tools that let you model your level right inside the software. It contains a ready-made First Person Controller Prefab object you can plunk into the world with automatic WASD keyboard controls that will allow you to explore the terrain. Unity automatically takes care of the rendering, collisions, physics, and sound effects.

Unity documentation also contains a wealth of valuable information, but, as with any technology that has a unique vocabulary, it’s sometimes hard to find just what you’re looking for.

We can download Unity from http://unity3d.com/unity/download/ and install it. When we run it for the first time, we’ll need to register the product by following the prompts. Registration can be done quickly online even if the machine you’ve installed it on isn’t connected to the Internet. If your machine is connected, you’ll be taken directly to the registration page. There are two main licenses: Unity and Unity Pro, with the Pro version being available for a price and the non Pro version being free. The Pro version has additional features, such as render-to-texture, occlusion culling, global lighting and post-processing effects. The Free version, on the other hand, displays a splash screen in standalone games and a watermark in web games that cannot be customized or disabled. Both Unity and Unity Pro include the development environment, tutorials, sample projects and content, support via forum, wiki, and future updates in the same major version. Unity for iOS and Unity for Android are add-ons to an existing Unity purchase.

In Unity, objects can be manipulated by any number of scripts in the scene. A script can reside on the GameObject it will manipulate, or, as in the case of a light switch, it could trigger an action on a different object such as a light. While an object could conceivably have a single script containing all of the information and functionality it needs, the script would tend to be too specialized and not very reusable. Scripts consist of four main types of components: variables, functions, equations, and comments. Variables hold values that can be anything from numbers to text. Functions do something, generally with variables and equations. Comments are ignored when the code is executed, allowing you to make notes about what the code is or should be doing or even to temporarily disable the code.

The Unity-SCORM Integration Toolkit is the result of an ADL Technical Team research and development project which focused on using games and simulations as part of an e-learning curriculum.

Unity 3D was selected as the game development tool for this prototype due to its ability to create web-based content. Developers can use simple methods, provided by a ScormManager object, to set the SCORM Run-Time Data Model elements without having prior experience with SCORM. The ScormManager object can be used to set values including scores, objectives and interactions. For advanced users, the entire SCORM data model is available for use.

The Unity-SCORM Integration Toolkit also contains a packaging tool that can be used to create a simple SCORM Content Aggregation Package. The ScormManager object and packaging tool support both SCORM Version 1.2 and SCORM 2004 4th Edition. With minor tweaks to the resulting package, SCORM 2004 2nd and 3rd Editions can also be supported.

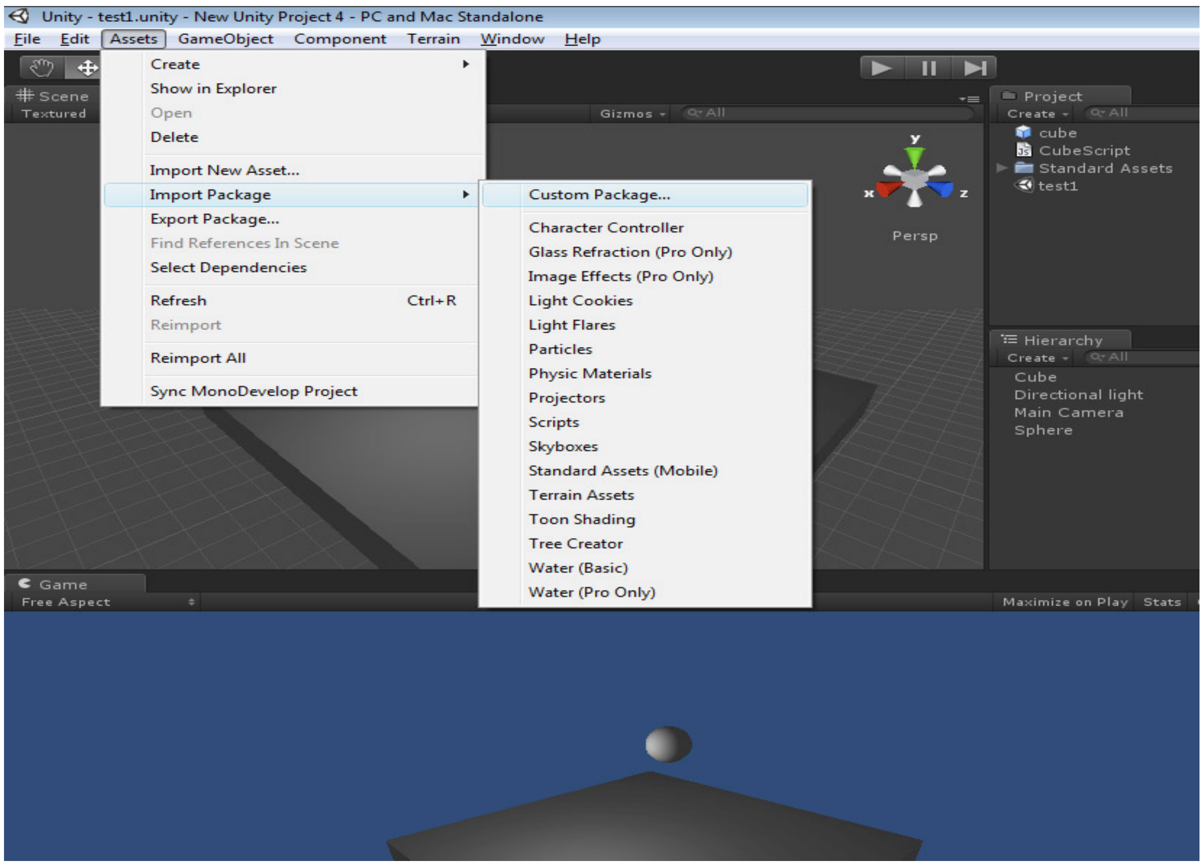
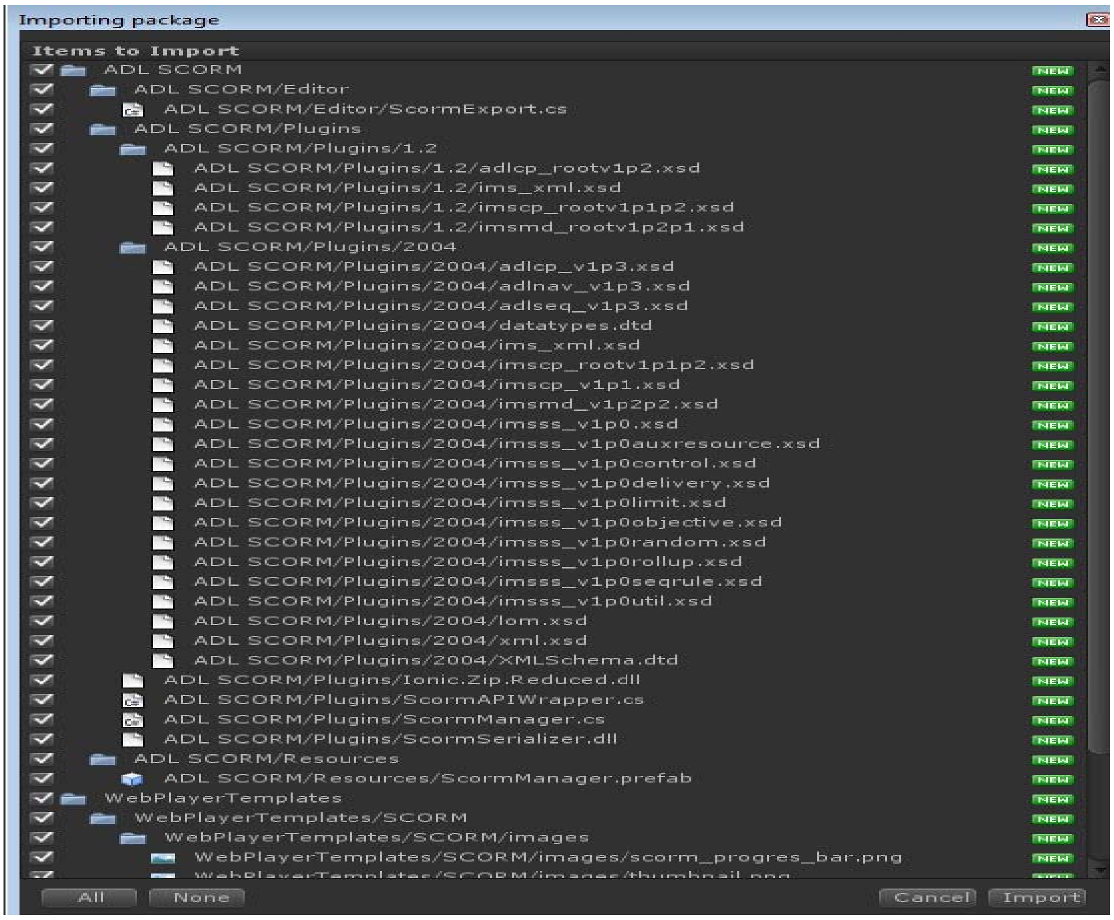


Figure 1. Import Unity-SCORM Integration Toolkit into your Unity project

To install the software following steps are required:

- download the ScormIntegrationKit.unitypackage from http://www.adlnet.gov/scorm-unity-integration;
- import the package into your Unity project by selecting **Assets->Import Asset Package->Custom Package**. You will see a new menu item called **SCORM**. The Unity-SCORM Integration Kit will need to be imported into each of your Unity projects.;



Items to Import

```
ADL SCORM
  ADL SCORM/Editor
    ADL SCORM/Editor/ScormExport.cs
  ADL SCORM/Plugins
    ADL SCORM/Plugins/1.2
      ADL SCORM/Plugins/1.2/adlcp_rootv1p2.xsd
      ADL SCORM/Plugins/1.2/ims_xml.xsd
      ADL SCORM/Plugins/1.2/imscp_rootv1p1p2.xsd
      ADL SCORM/Plugins/1.2/imsmd_rootv1p2p1.xsd
    ADL SCORM/Plugins/2004
      ADL SCORM/Plugins/2004/adlcp_v1p3.xsd
      ADL SCORM/Plugins/2004/adlnav_v1p3.xsd
      ADL SCORM/Plugins/2004/adlseq_v1p3.xsd
      ADL SCORM/Plugins/2004/datatypes.dtd
      ADL SCORM/Plugins/2004/ims_xml.xsd
      ADL SCORM/Plugins/2004/imscp_rootv1p1p2.xsd
      ADL SCORM/Plugins/2004/imscp_v1p1.xsd
      ADL SCORM/Plugins/2004/imsmd_v1p2p2.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0auxresource.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0control.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0delivery.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0limit.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0objective.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0random.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0rollup.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0seqrule.xsd
      ADL SCORM/Plugins/2004/imsss_v1p0util.xsd
      ADL SCORM/Plugins/2004/lom.xsd
      ADL SCORM/Plugins/2004/xml.xsd
      ADL SCORM/Plugins/2004/XMLSchema.dtd
    ADL SCORM/Plugins/Ionic.Zip.Reduced.dll
    ADL SCORM/Plugins/ScormAPIWrapper.cs
    ADL SCORM/Plugins/ScormManager.cs
    ADL SCORM/Plugins/ScormSerializer.dll
  ADL SCORM/Resources
    ADL SCORM/Resources/ScormManager.prefab
WebPlayerTemplates
  WebPlayerTemplates/SCORM
    WebPlayerTemplates/SCORM/images
      WebPlayerTemplates/SCORM/images/scorm_progres_bar.png
      WebPlayerTemplates/SCORM/images/thumbnail.png
```

Figure 2. Items of Unity-SCORM Integration Toolkit

ADL provides a small demo game on github.com website with SCORM support and a quick start guide. In developing a game with Unity and SCORM Integration Toolkit the main issues are:

- in your game startup logic, make sure the simulation does not begin or pauses until you receive the `Scorm_Initialization_Complete` message.
- decide the condition under which you wish to mark your course as complete and add code to your scripts that executes when this condition is met. Use this code:

```
ScormManager.SetCompletionStatus(completionStatusType.complete);
ScormManager.SetSatisfaction(successStatusType.sucess);
ScormManager.Commit();
```

- locate your simulation's shutdown or exit procedure and call `ScormManager.Terminate()` as the last function in the shutdown procedure.

ScormManager object sends all messages to its childs and because of this reason all game objects must have ScormManager like parent. The ScormManager object is created from SCORM Menu.

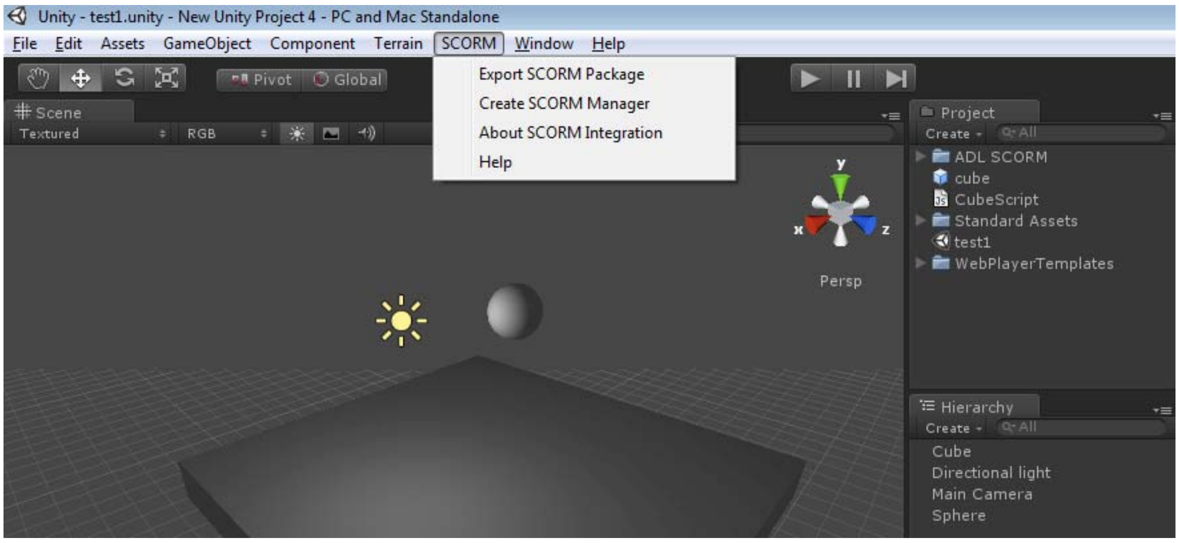


Figure 3. SCORM Menu

The toolkit framework is developed in C# language. But is possible to work with Javascript language to access C# functions of the framework. Place the C# script in the "Standard Assets" folder and the Javascript outside of the "Standard Assets" folder. The Javascript can now reference the C# script directly.

After export the game like a package from SCORM Menu it can be loaded on an e-learning platform.

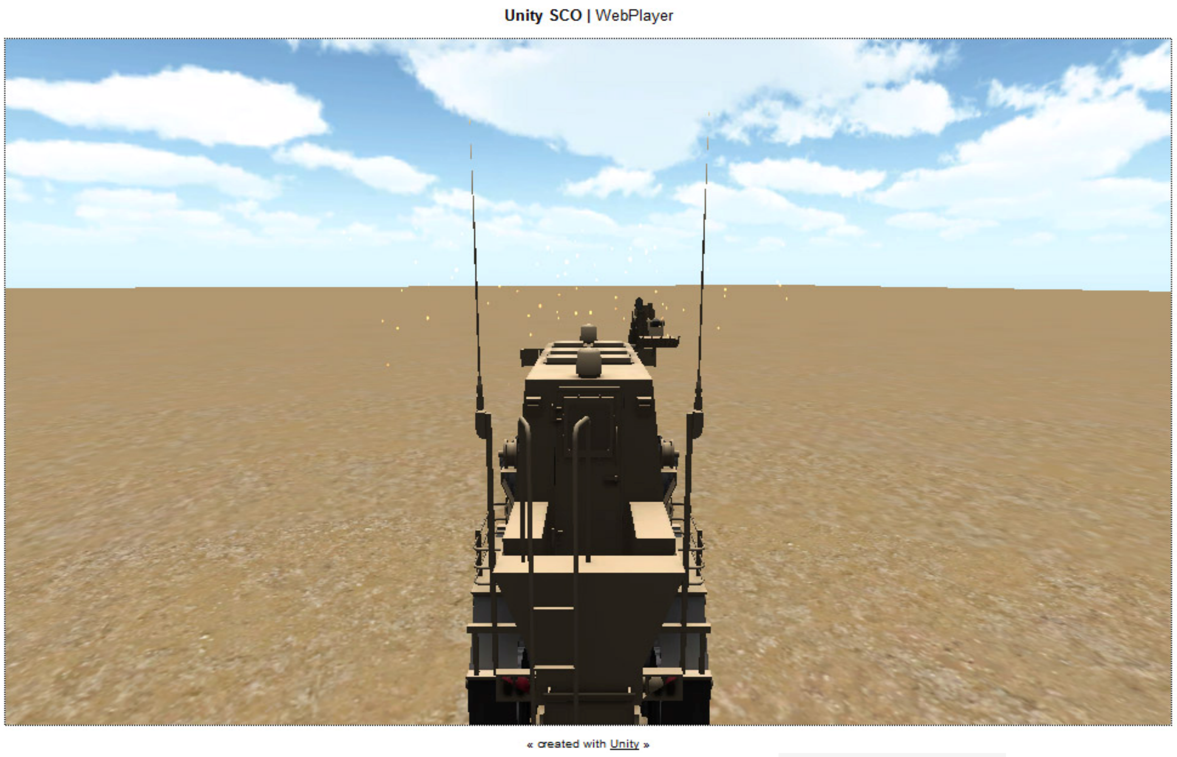


Figure 4. SCORM demo game on ILIAS e-learning platform

The result is a SCO which represents an activity in imsmanifest file. An interesting aspect would be that scenes developed in Unity to represent activities described in SCORM manifest file. In this way it would be possible to implement all kind of sequencing from SCORM standard and also game logic can include a sequencing component.

References

http://unity3d.com/
http://en.wikipedia.org/wiki/Unity_game_engine
https://github.com/adlnet/scorm4unity
http://www.adlnet.gov/