Scorm API

Supported Platforms

Scorm packages run on LMS that run on Web Browsers.

- **WebGL** support. Requires a custom WebGL template already included or some changes to your own WebGL template.
- **WebPlayer** support. Requires a custom WebPlayer template already included or some changes to your own WebPlayer template.

Features

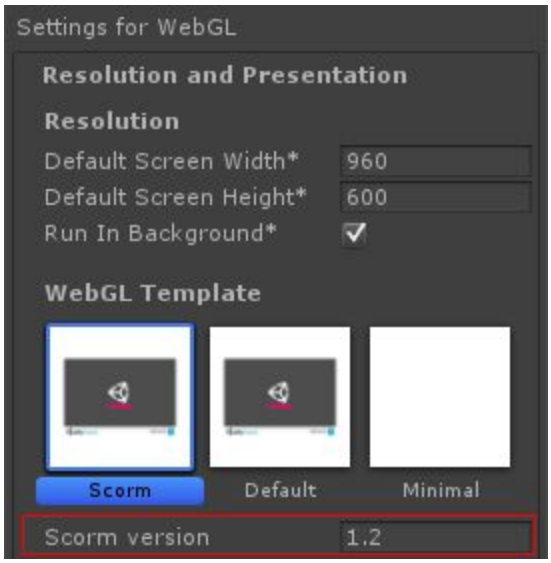
- **Scorm 1.2** and **Scorm 2004** support. Implemented 1.2/2004 equivalent API, If you need something specific of Scorm 2004 that is not available in 1.2, feel free to ask and you will have it as soon as possible.
- **Builds Scorm packages automatically.** You get a "*.zip" with your Scorm package ready to upload to a LMS.
- **Full object oriented API** (you don't deserialize anything, it's already done it for you).
- **C# Source code** included.
- **Documentation** available.
- **Easy integration** to your Unity project.
- **WebGL and WebPlayer templates included** but you can use it on your own template.
- Well **organized and structured** code.
- **Event based**. Example: *OnLearnerNameRetrieved(string learnerName), OnLanguageRetrieved(string language)*
- **Examples** available. Includes an example to test the full API with a console view.

Getting started

You have two choices:

Using included WebGL template

1. Unzip “ScormAPI/WebGLTemplate.zip” to “Assets/WebGLTemplates” folder and rename it to what you want.
2. Go to “File > Build Settings > WebGL > Resolution and Presentation > **WebGL Template**” and select your template.
3. You will see below a “**Scorm version**” field that accepts two values: “1.2” or “2004”.



4. Done!

Adding it to your own WebGL template

1. Copy “js” folder included in “ScormAPI/WebGLTemplate.zip” to your own WebGL template.
2. Make the following change to the “index.html” file in your own WebGL template:

Above this:

```
<script src="TemplateData/UnityProgress.js"></script>
```

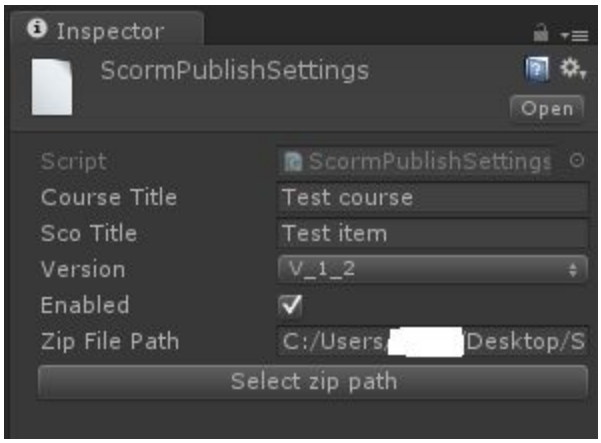
Add this:

```
<script type="text/javascript" src="js/ostynscormtime.js"></script>
<script type="text/javascript" src="js/scorm.js"></script>
<script type="text/javascript">
var processedUnload = false;

var scorm = new Scorm(Scorm.SCORM_%UNITY_CUSTOM_SCORM_VERSION%);

window.onbeforeunload = function(e) {
        doUnload();
};

window.onunload = function(e) {
        doUnload();
};

function doUnload() {
        if (processedUnload) return;

        processedUnload = true;

        SendMessage("ScormAPI", "Close");
}
</script>
```

3. Follow from step 2 of “**Using included WebGL template**” approach.

Scorm Build

It can build a “*.zip” file ready to upload to a LMS:



1. Insert your Course title in “Resources > ScormPublishSettings.asset > Course Title” field.
2. Insert your Sco title in “Resources > ScormPublishSettings.asset > Sco Title” field.
3. Select your Scorm version in “Resources > ScormPublishSettings.asset > Version” field.
4. Enable publish in “Resources > ScormPublishSettings.asset > Enabled” field.
5. Insert your “*.zip” file path in “Resources > ScormPublishSettings.asset > Zip file path” field (or click the button and select the path).

Every time you build to WebGL platform, it will generate a “*.zip” file in the desired path.

Scorm Package Test

If you don’t have or can’t have access to a LMS you can test it with the following applications:

Scorm Cloud

Supports 1.2/2004:

1. You can create a free account if your content doesn’t exceed 100MB:

http://scorm.com/scorm-solved/scorm-cloud-features/content-just-works/testing-content-with-scorm-cloud/

2. Upload your Scorm package and test it!

Scorm 1.2 Player

Supports only 1.2.

1. Download it:

http://www.reload.ac.uk/scormplayer.html

2. Run “reload-scorm-player.jar”.
3. Open your Scorm 1.2 package and name it.
4. Press “Play” to test it!

API Reference

Implemented all the equivalent 1.2/2004 API:

http://scorm.com/scorm-explained/technical-scorm/run-time/run-time-reference/

General

Events

public delegate void InitHandler();
public event InitHandler **OnInitialized**;
 	
public delegate void CloseHandler();
public event CloseHandler **OnUserClosed**;

Methods
public void **Init**(); public void **Commit**(); public void **Exit**(ExitReason reason); public void **Finish**();

LearnerId

Events
public delegate void LearnerIdHandler(string learnerId); public event LearnerIdHandler **OnLearnerIdRetrieved**;
Methods
public void **GetLearnerId**();

LearnerName

Events
public delegate void LearnerNameHandler(string learnerName); public event LearnerNameHandler **OnLearnerNameRetrieved**;
Methods
public void **GetLearnerName**();

LessonLocation

Events
public delegate void LessonLocationHandler(string lessonLocation); public event LessonLocationHandler **OnLessonLocationRetrieved**;
Methods
public void **GetLessonLocation**(); public void **SetLessonLocation**(string lessonLocation);

Credit

Events
public delegate void CreditHandler(CreditType credit); public event CreditHandler **OnCreditRetrieved**;
Methods
public void **GetCredit**();

LessonStatus

Events
public delegate void LessonStatusHandler(LessonStatus lessonStatus); public event LessonStatusHandler **OnLessonStatusRetrieved**;
Methods
public void **GetLessonStatus**(); public void **SetLessonStatus**(LessonStatus lessonStatus);

Entry

Events
public delegate void EntryHandler(EntryType entry); public event EntryHandler **OnEntryRetrieved**;
Methods
public void **GetEntry**();

RawScore

Events
public delegate void RawScoreHandler(float rawScore); public event RawScoreHandler **OnRawScoreRetrieved**;
Methods
public void **GetRawScore**(); public void **SetRawScore**(float rawScore);

MaxScore

Events
public delegate void MaxScoreHandler(float maxScore); public event MaxScoreHandler **OnMaxScoreRetrieved**;
Methods
public void **GetMaxScore**(); public void **SetMaxScore**(float maxScore);

MinScore

Events
public delegate void MinScoreHandler(float minScore); public event MinScoreHandler **OnMinScoreRetrieved**;
Methods
public void **GetMinScore**(); public void **SetMinScore**(float minScore);

TotalTime

Events
public delegate void TotalTimeHandler(int totalTimeInMiliseconds); public event TotalTimeHandler **OnTotalTimeRetrieved**;
Methods
public void **GetTotalTime**();

LessonMode

Events
public delegate void LessonModeHandler(LessonMode lessonMode); public event LessonModeHandler **OnLessonModeRetrieved**;
Methods
public void **GetLessonMode**();

SessionTime

Events
Methods
public void **SetSessionTime**(int milliseconds);

Comments

Events
public delegate void CommentsHandler(string comments); public event CommentsHandler **OnCommentsRetrieved**;
Methods
public void **GetComments**();

CommentsFromLMS

Events
public delegate void CommentsFromLMSHandler(string commentsFromLMS); public event CommentsFromLMSHandler **OnCommentsFromLMSRetrieved**;
Methods
public void **GetCommentsFromLMS**();

Language

Events
public delegate void LanguageHandler(string language); public event LanguageHandler **OnLanguageRetrieved**;
Methods
public void **GetLanguage**();

SuspendData

Events
public delegate void SuspendDataHandler(string data); public event SuspendDataHandler **OnSuspendDataRetrieved**;
Methods
public void **GetSuspendData**(); public void **SetSuspendData**(string data);

FAQ

1. **Why returned results are always ‘undefined’?**

Read Getting started and Using included WebGL template section. Chances are that you haven't set Scorm version in step 3 or you have it with an incorrect version.

2. **I get this message when uploading a Scorm to Docebo LMS: "Some of the archive files are not allowed". What can I do?**

Unity generates a “htaccess” that it’s not a valid file for Docebo. You must remove it from the output Scorm zip file.

3. **I get a some errors regarding “WebGLTemplates”, what can I do?**

In the “Getting started” section it was mentioned that the zip file needs to be decompressed in “Assets/WebGLTemplates” folder because “WebGLTemplates” only can be in the root directory.