

# lòÓneyBITS

## Game\_Core

(internal lib)

29/10/2014

Update 06/06/2015 Loading screen updated, stats data updated.

Update 27/07/2015 New functionality, scenes and look&Feel added.

Update 06/12/2015 AdmobManager added & debugged code.

Update 13/03/2016 SoundManager added & code debugged.

Is a production of lòÓneyBITS

<http://www.looneybits.com>

## Table of Contents

1. Game Core.....	4
1.1.Summary.....	4
1.2.Document tree.....	4
1.3. Social Buttons.....	4
1.3.1 Twitter button.....	4
1.3.2 Facebook button.....	5
1.4. M.V.C.....	6
1.5.Literature.....	7



# 1. Game Core

## 1.1.Summary

This is a useful package with all basic stuff to build a basic game. This complete project will help you develop other games faster. Code is fully commented.

## 1.2.Document tree

The project has the following folders:

- Physics Materials: This folder contains the material of balls/coins.
- Fonts: This folder contains the fonts of game.
- Music: Here is the sound effects and music.
- Resources: This folder contains the GameObjects(Prefabs...) of game.
- Scripts: This folder contains the C# scripts of the game.
- Sprites: This folder contains the sprites .
- Src: This folder contains SVG files.
- Plugins: This folder contains plugins.
- Doc: This folder contains project documentation. **Don't unzip doxfile\_readme.zip in the asset(unity project) folder because it contains JavaScript files.**

## 1.3. Social Buttons

The social buttons perform the function of sharing the player's score on your favorite social network. The Twitter button share a tweet with the hashtags of the game and dose not have previous requirements, in the other hand, Facebook requires a developer account and app id. Both buttons opens a new windows on the browser. **NOT OPENS TWITTER OR FACEBOOK APP. TESTED ON ANDROID / PC / MAC & WEBPLAYER.**

### 1.3.1 Twitter button

**C# Slice:**

```
string twittershare=" #hashtag_N+1")  
score"+score);  
Application.OpenURL(twittershare);
```

*Why hashtags by separated? Because I have had troubles with url encode if hashtag was merged with message.*

### 1.3.2 Facebook button

Right now the best solution to solve the problem of share button with facebook is use the new dialog system. The old Sharer (sharer.php) has suffered some changes.

Dialog System Example:

```
http://www.facebook.com/dialog/feed?  
app_id=123050457758183&  
link=http://developers.facebook.com/docs/reference/dialogs/& picture=http://looneybits.com/assets/img/p04.png&  
name=Facebook%20Dialogs&  
caption=Reference%20Documentation&description=Dialogs%20provide%20a%20simple,%20consistent  
%20interface%20for%20applications%20to%20interact%20with%20users.&message=Facebook%20Dialogs%20are  
%20so%20easy!&redirect_uri=http://www.example.com/response
```

Requirements:

**Facebook Account.**

**Facebook developer account(profile).**

**APP\_ID.** Register a simple app on <http://developers.facebook.com>.

**Hosting/server or whatever you want on the net, you need that to set the redirect\_uri and pictures of game.**

OLD Sharer.php System Example(Only works url data):

```
http://www.facebook.com/sharer.php?s=100  
&p[title]=TITLE  
&p[url]=http://looneybits.com  
&p[summary]=yoursummaryhere  
&p[images][0]=http://looneybits.com/assets/img/p04.png";
```

If you want to use this system you should develop a script on the server side to generate a dynamic score page. For example: <http://mysupergame.com/score/2000>

**C# Slice**

```
String facebookshare=" http://www.facebook.com/sharer.php?s=100  
    &p[title]=TITLE  
    &p[url]=http://mysupergame.com/score/" + SCORE  
    "&p[summary]=yoursummaryhere  
    &p[images][0]=http://looneybits.com/assets/img/p04.png";
```

*Application.OpenURL(facebookshare);*

Requirements:

*Hosting/server or whatever you want on the net, you need that to set the redirect\_uri and pictures of game.*

## 1.4. M.V.C

The project is programmed with the MVC (Model-View-Controller) architectural pattern. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.

- **Model:** notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands. In some cases an MVC implementation
- **View:** Requests information from the model that it uses to generate an output representation to the user.
- **Controller(Behavior):** can send commands to the model to update the model's state. It can also send commands to its associated view to change the view's presentation of the model

## 1.5.Literature

<i>Reference</i>	<i>Application</i>
<a href="#">Unity3D Forum</a>	Learn about social buttons “Trilusion” member comment. <i>Thanks.</i>
<a href="#">Unity3D Asset Store Learn</a>	Learn about boundary control. <i>Thanks.</i>
<a href="#">Video Tutorial (YouTube)</a>	About collisions. <i>Thanks.</i>
<a href="#">Unity3D Asset Store Learn</a>	Learn about spawn system. <i>Thanks.</i>
<a href="#">Wiki MVC</a>	Theoretical concepts. <i>Thanks.</i>
<a href="#">Screen Shake</a>	Screen Shake effect. <i>Thanks.</i>
<a href="#">Filter Pattern</a>	Theoretical concepts. <i>Thanks.</i>
<a href="#">Doxygen</a>	Doc generator. Thanks.