MultiOnline

Multiplayer online kit

1. Introdution

The MultiOnline is a kit for multiplayer online game.
With this package, a player can view the list of the current games or create and host a game himself.

This package require a web hosting with PHP 5 (with the mail function activated) and MySQL with PhpMyAdmin (If you don't have PhpMyAdmin it will work too, but you'll have to know how create and intall your database alone). You can use your own web server if you have one, or a shared hosting. The web hosting is only used to centralize the open games, so you needn't a big disk space (for my examples, I use the free hosting offer of alwaysData, it work very well).

With that, you need a FTP software (to send your file on your web hosting) and a PHP editor (you can use Notepad++ it'll be enough for what you have to do).

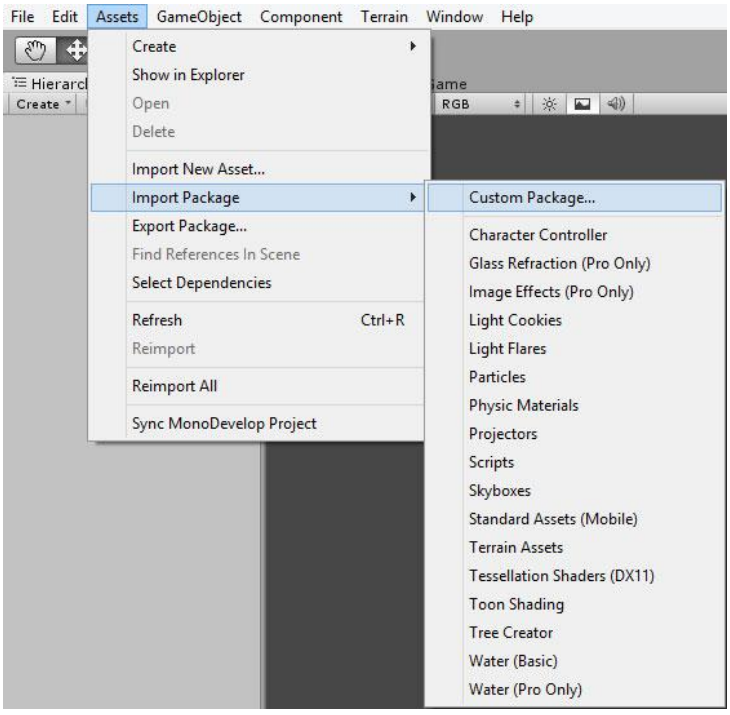
Note : this documentation will not explain how use a FTP software or how create or host a website. If you have no experience about that, please search on Internet, it's very easy to find documentation and tutorials for that.

This package is full compatible with MultiLan Asset.

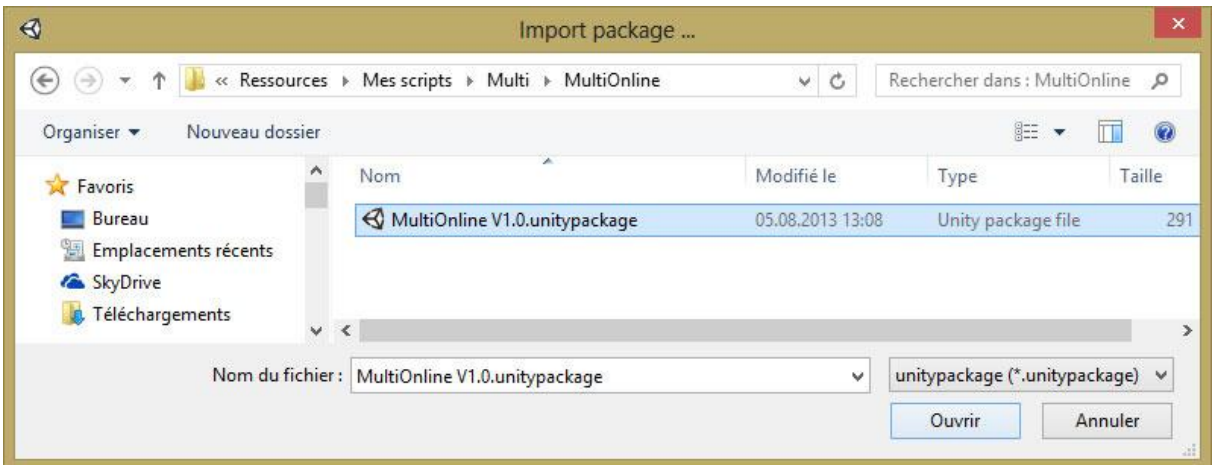
2. Installation

2.1. Import the package

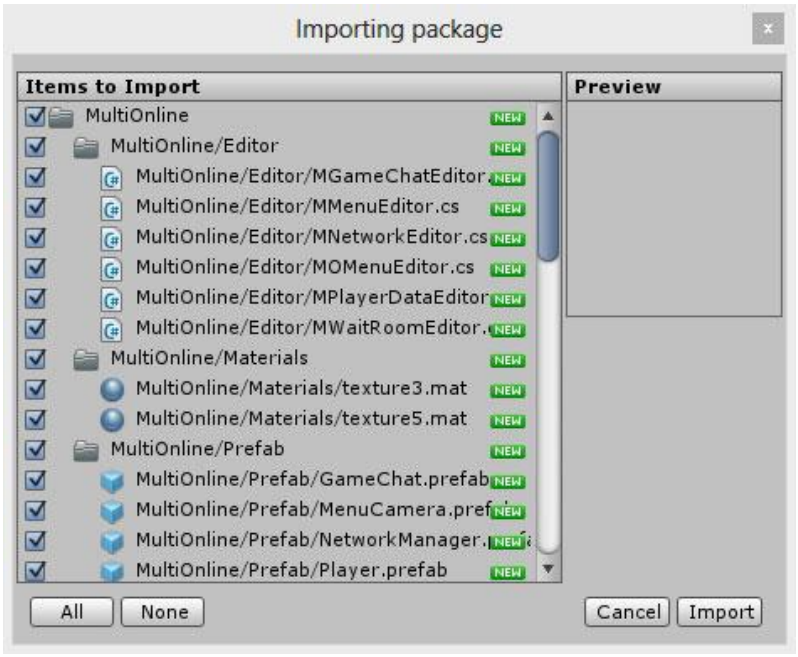
Once you have download the package, open Unity Editor, and click on Assets/Import Package/Custom Package... :



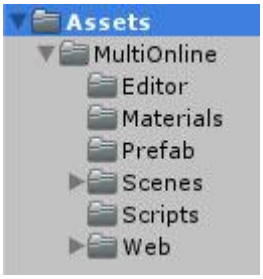
Select the package and click on "Open" :



Let all boxes checked and click on "Import" :



Your package is now imported, you have now a folder named "MultiOnline" on the Asset folder, which contains sixe other folders :



2.2. Package content

Now, we'll see the different elements of the package.

- Editor : contains the editions script, you have no reason to open them
    - MGameChatEditor
    - MMenuEditor
    - MOMenuEditor
    - MNetworkEditor
    - MPlayerDataEditor
    - MWaitRoomEditor

- Prefab : contains the game prefabs
    - GameChat : if you want to have a chat on your games scenes
    - MenuCamera : a camera prefab used on the menu scene and on the waiting room for manage the background color. You can use it if you add menus on your game
    - NetworkManager : this prefab contains the script which manage the network, it is automatically loaded and destroyed when he must be
    - Player : for the sample it's a simple cube but it is strongly recommended to change it for your final game :P
    - PlayerData : this prefab contains the script which manage all the player informations (private and public IP, username, status in game, ID in game). You don't have to worry about him: it is automatically loaded when he have to.
    - Spawns : this gameObject is necessary for load the players, if you create your own maps, you have to add it on your scenes

- Scenes : contains the four scenes of the package and an images folder :
    - Menu : the main menu of the game. It's the entry point of the game
    - WaitRoom : the waiting room, used for wait for players before start game
    - Map1 : the first sample map
    - Map2 : the second sample map
    - The MapScreen folder : contains the screenshots of the games map (so that we can show the game's map on the waiting room)

- Scripts : contains the game scripts :
    - MCheckPort : functions script, used to know which player can be host
    - MFilter : functions script, use to check player's entries before use it
    - MGame : this class is only used for formatting the open network games
    - MGameChat : manages the game chat
    - MMenu : manages the main menu
    - MNetwork : manages the all the network

- MOMenu : manage the menu specific to MultiOnline
- MOServer : manage all the connexion with your web hosting
- MPlayerData : contains the players informations during game (is player login, IP, username, status...)
- MPlayerMove: manages the player movements, it is currently a very basic script, you can modify many things on it and add functions
- MPlayerNetwork: disables the others players and saves our player current position in prevent of an host migration
- MSpawn : loads the player on the map
- MUser : this class is only used for formatting game's user list
- MWaitRoom: manages the waitRoom

- Web : contains the MySQL build and the PHP scripts you'll upload on your web hosting. We will see how install that on two next parts.

Note : many scripts and components are the same in MultiLan and MultiOnline packages. The scripts specific to MultiOnline package begin with the letters "MO", the others begin with the letter "M".
In order to make MultiLan and MultiOnline full compatible and usables separately in the mean time, I had to put in comments few lines which required MultiLan components on the scripts MMenu. If you want to know how use MultiLan and MultiOnline together, everything is explain on the other documentation.

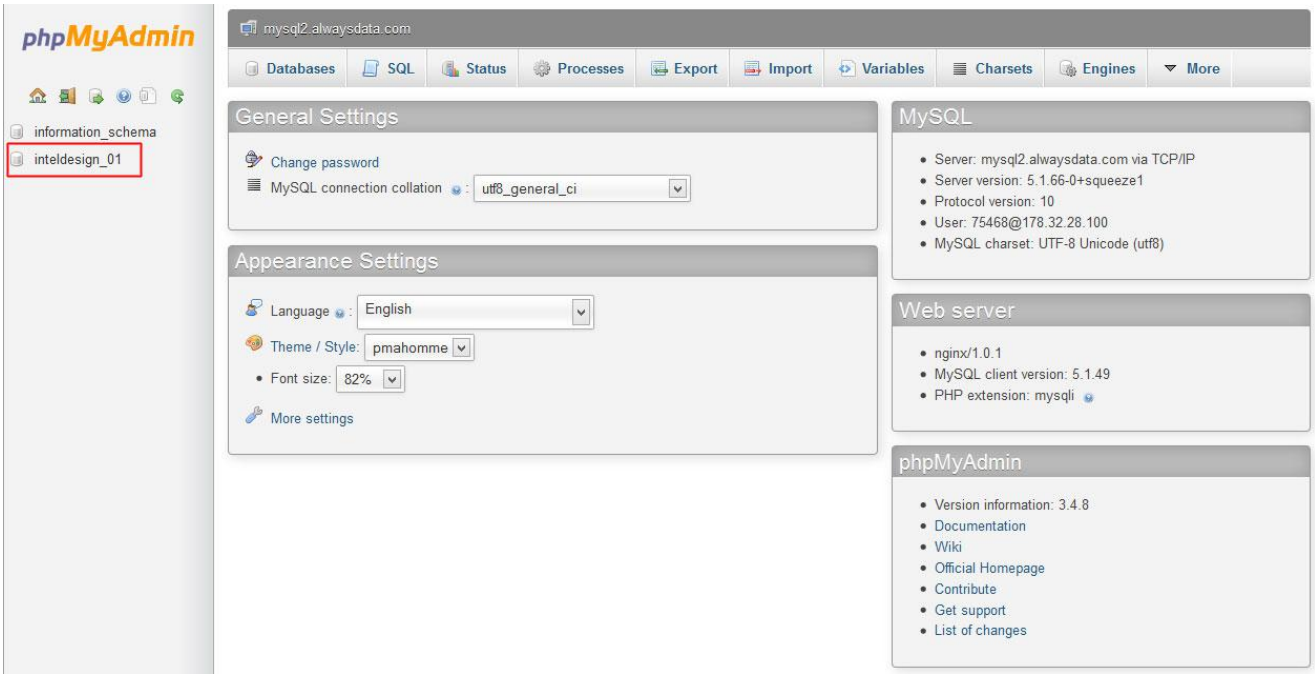
2.3. Install the database

Install the database is quite easy, but you must first create your database. I don't give you any script for that, because in many case, you must do that directly from your web hosting adminstration panel.
The name of your database does not matter and it's possible that your web host decide her name for you (in my case, my database has the name of my domain).

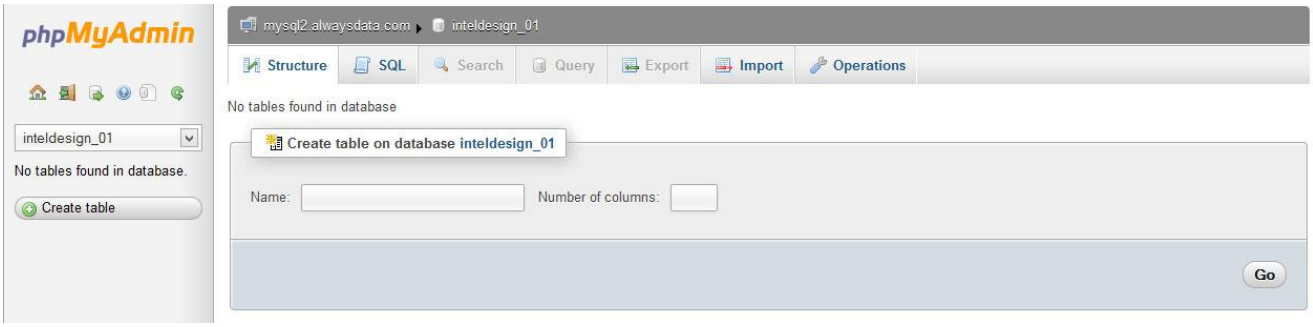
Once your database is created, open PhpMyAdmin.

On the left sidebar you have the list of your databases.

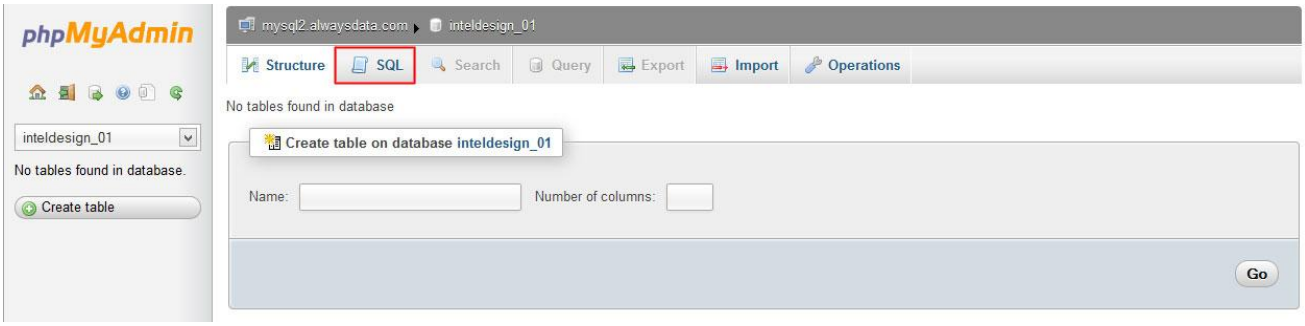
Click on the one you just have created for MultiOnline :



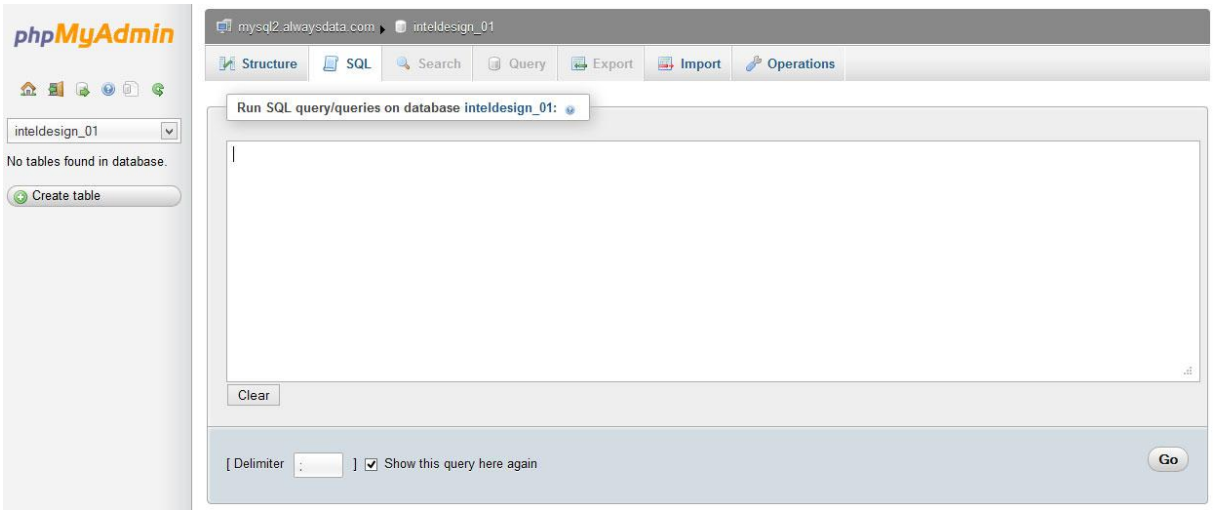
Once you have select your database, you should have something like that :



Click now on the menu "SQL":

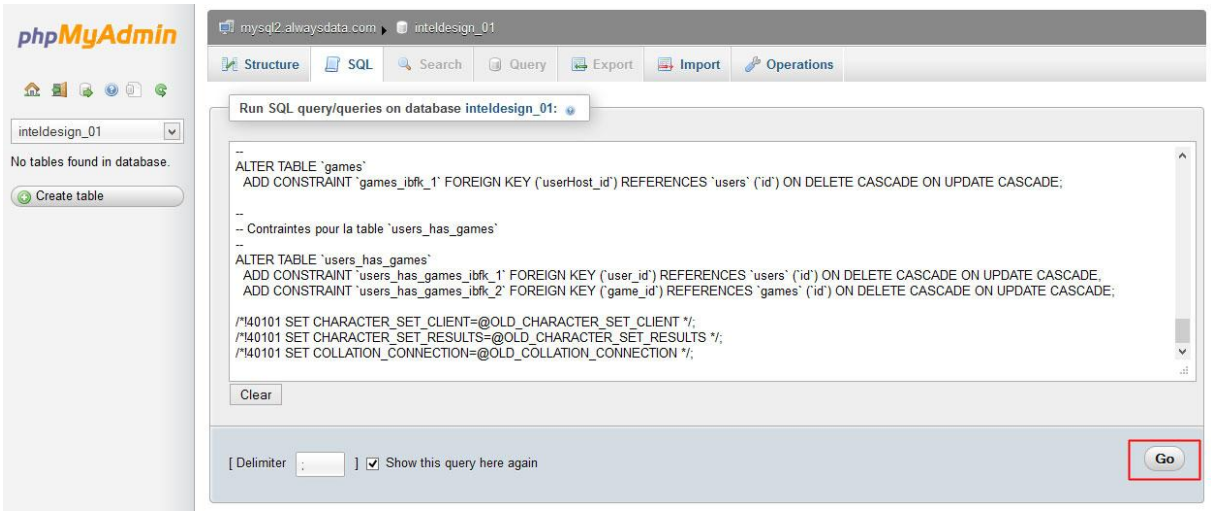


You will have a page with a big textarea, like that :

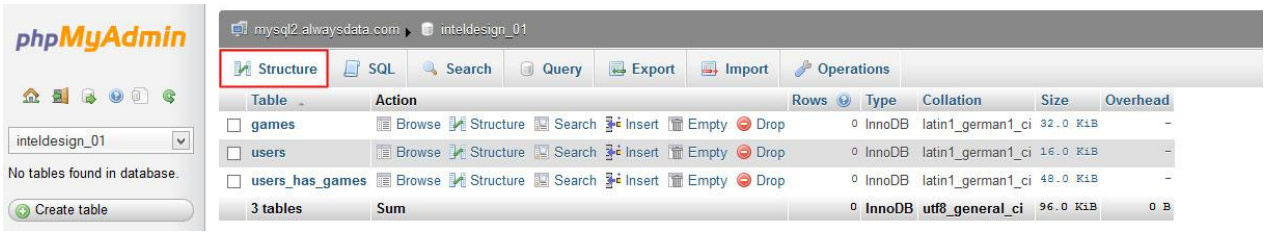


Now, open your file *MySQL build.sql* (on your folder MultiOnline/Web/MySQL build) - you can open it with notepad++, or even with Windows Notepad, it does not matter because we will simply copy the contents of the file.

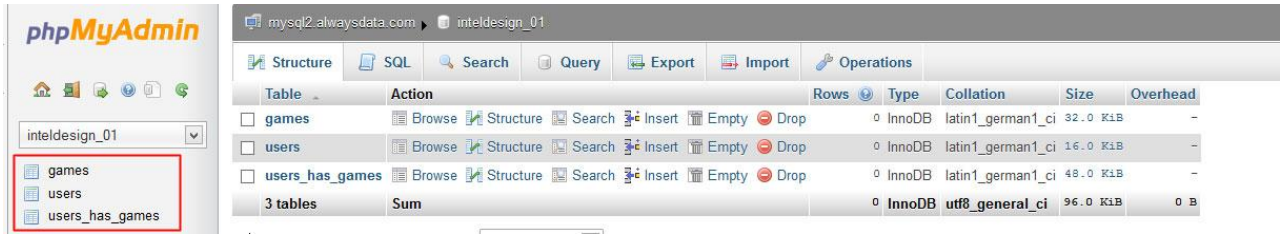
So open it, select all the content and copy it (Ctrl+A, Ctrl+C), go back on PhpMyAdmin, past (Ctrl+V) it on the big textarea, and click on "Go" :



When it's done, click on the menu "Structure", you must see your three MySQL tables : *games*, *users* and *users_has_games* :



And if your refresh your web page, your tables list will appear on the left sidebar too :



That's all for MySQL, your database is now finish, you'll have no more to worry about it !
Now we will see how install and set the PHP files.

2.4. Install the PHP files

Now open your FTP software and connect it with your website (I'll not show you how do that, because it's different for each FTP software).
When it's done, open your "Web pages" folder (in MultiOnline / Web) and put all his content on your website local folder. You can put it on the root or on a specific folder named as you want (personnaly I have put it on a folder that I called "MultiOnline").
Now you'll have to open some PHP files which are on the "*functions*" folder to define some settings.

Note : open and modify the files which are now on your website local folder, not those which are on Unity Asset folder.

First open the "*functions*" folder, and open MO*Db.class.php* with Notepad++ (or the software on your choice).
Line 8 to 17, you have to put your database connexion settings :

```
6   class MODb {
7
8       /*************** YOU MUST COMPLETE THESE PARAMETERS ******************/
9       // Enter here your database username :
10      private $user = 'username';
11      // Enter here your database password :
12      private $pass = 'password';
13      // Enter here your database server (it's often "localhost") :
14      private $host = 'localhost';
15      // Enter here the name of your database :
16      private $dbName = 'database name';
17      /*********************************************************************/
18
```

- Your username and password are the same as those you use to connect on PhpMyAdmin.
- Your database server name will be given by your web host (it's often "localhost", but it can be something else too).
- Your database name is the name you called your database when you have created it just before

When you have fill these four fields, save your file and close it.

Now open *MOUser.class.php*.
Line 6 to 27, you can set the emails which will be send from your game :

```
4  class MOUser extends MODb{
5
6      /**************** YOU CAN CHANGE THESE PARAMETERS ******************/
7      // Sender e-mail :
8      private $mailSender = 'Sender e-mail';
9
10     // Email for account create :
11     // Subject :
12     private $mailRegisterSubject = 'Your account have been created !';
13     // Message (in HTML)
14     private $mailRegisterContent = 'Hello<br/><br />Welcome on SuperGame, you
15
16     // Email for "forgotten username" :
17     // Subject :
18     private $mailUserNameSubject = 'Your userName';
19     // Message (in HTML)
20     private $mailUserNameContent = 'Hello<br/><br />You are registred on Supe
21
22     // Email for "forgotten password" :
23     // Subject :
24     private $mailNewPassSubject= 'Your password';
25     // Message (in HTML) :
26     private $mailNewPassContent = 'Hello<br/><br />This is your new password
27
28     /*************************************************************/
```

- Line 8 : put the e-mail address which will appear as the sender on the automatic e-mail which will be send from your game (for example you can choose an address as no-reply@yourdomain.com)

The other lines contains the content of the differents e-mail which can be send from your game, separate in two variable (the first for the e-mail's subject, the second for the e-mail's content). The e-mail are sent in HTML format, so you can put HTML tags on the e-mail's content.

- Line 12 and 14 : it's the subject and the content of the e-mail which will be send when a player registered on your game for the first time
- Line 18 and 20 : it's the subject and the content of the e-mail which will be send when a player will click on "forgot username"
- Line 24 and 26 : it's the subject and the content of the e-mail which will be send when a player will click on "forgot password"

Note 1 : on the e-mails content, you will see the strings " *$userName*" and " *$newPass*", these strings will be automatically change by their real values (player's username and player's new password) when the e-mail will be send. So you can use them on your e-mails when you want that the player's username or new password will be written.

Note 2 : when a player forgot his password, the script generate automatically a new password. It doesn't send the player's password by e-mail because for more security the players' passwords are hashed before being stored, so it's not possible to give them back.

That's all for this file, when you have finish your can save and close it.

Now, upload all your files on your website.

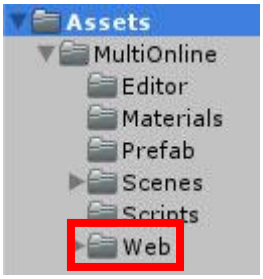
You have a last little thing to do : write the real path of the function folder on the *functions.php* file. First, you need to know the real path of the folder. For that go on the page *realPath.php* (in function file) from your web browser (URL looks like "*www.youdomaine.ch/folderwhere you put scripts/functions/realPath.php*").

This page will display a string : that's the real path of your functions folder (it looks like something as "*/home/www/xxxx/web/folder/functions*").

Copy this line, open the file *functions.php* (on functions folder) and past it on the *$path* variable, line 7, and add a slash just after the word "functions" (as "*/home/www/xxxx/web/folder/functions/*") :

```
1  <?php
2      spl_autoload_register('load_class');
3
4      function load_class($class){
5          // Real path of your functions folder
6          // (go on the page realPath.php with your web browser to know it)
7          $path = 'Real path of your function folder';
8
9          if(file_exists($path.$class.'.class.php') AND is_file($path.$class.'.class.php')){
10             require($path.$class.'.class.php');
11         }
12     }
```

When it's done, save the file, you can delete the file realPath.php (it will no longer serve) and re-upload your website.



That' all, the web install is now finish.

When your are sure that your database is well installed and your website online, you can delete all the Web folder on Unity Editor, you no longer need it).

You can go back on Unity Editor..

2.5. Set scripts parameters

Now you have to define the URL of your website on your Unity C# scripts.
Open the script *MOServer.cs* (MultiOnline/Scripts/MOServer), on line 11 enter the URL of the folder where you have put your web content with a slash in the end.

For example if your have put the scripts on a folder named MultiOnline : "*http://www.yourdomaine.com/MultiOnline/*", or if you have put the scripts at the root : "*http://www.yourdomaine.com/*".

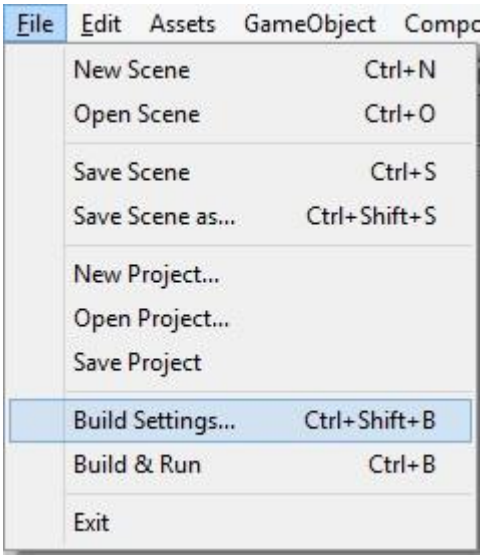
```
7   namespace MultiPlayer {
8       public class MOServer  {
9
10          // Main URL
11          public string url = "Path of your web site";
12          // Register user URL :
13          public string registerUrl = "saveUser.php";
14          // Login user URL :
15          public string logUrl = "logUser.php";
16          // Forgot Login user URL :
17          public string forgotLoginUrl = "forgotLogin.php";
18          // Logout user URL :
19          public string logoutUrl = "logoutUser.php";
20          // Change user profil URL :
21          public string changeProfilUrl = "changeProfil.php";
22          // Register new game URL :
23          public string registerGameUrl = "saveGame.php";
24          // Update game parameters URL :
25          public string updateGameUrl = "updateGame.php";
26          // Search game URL :
27          public string searchGameUrl = "searchGame.php";
28          // Exit game URL :
29          public string exitGameUrl = "exitGame.php";
```

In this script, line 13 to 29, the names of the differents web pages are written. If you change the name of the pages on your website, you'll have to put the new names here.

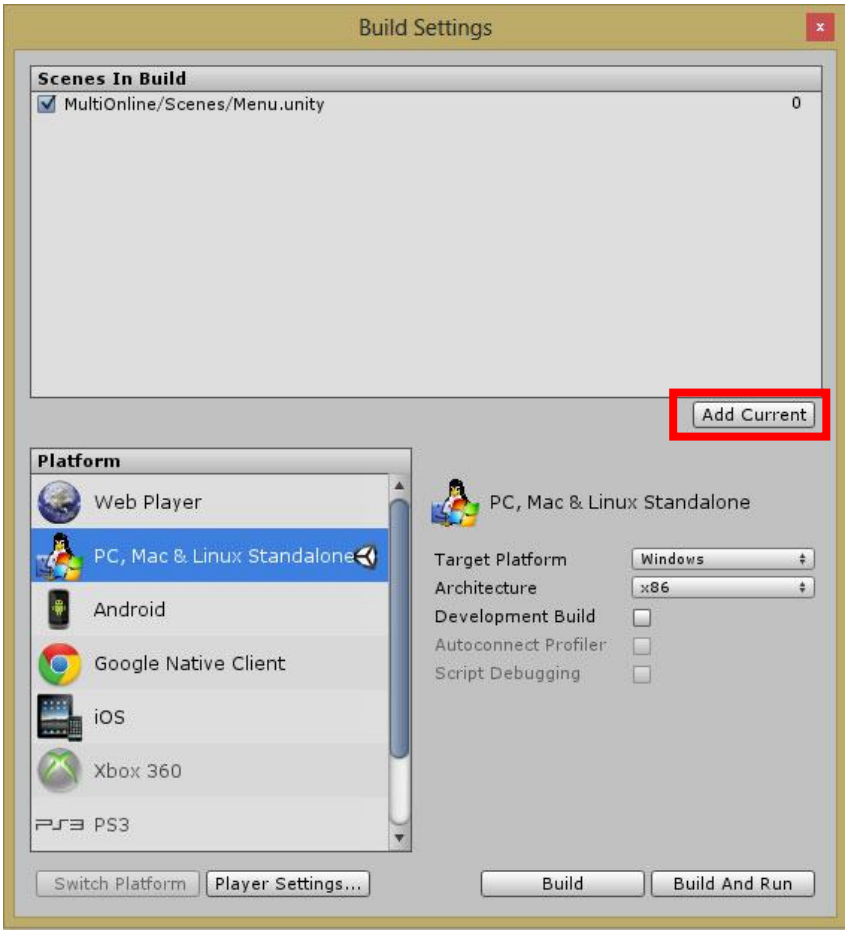
2.6. Building

There is one last step : before try our package, we have to build the differents scenes together, else it's impossible to load a scene from another (so, nothing works).

First, go on the Scenes folder and open "Menu".
Next, click on File/Build Settings... :

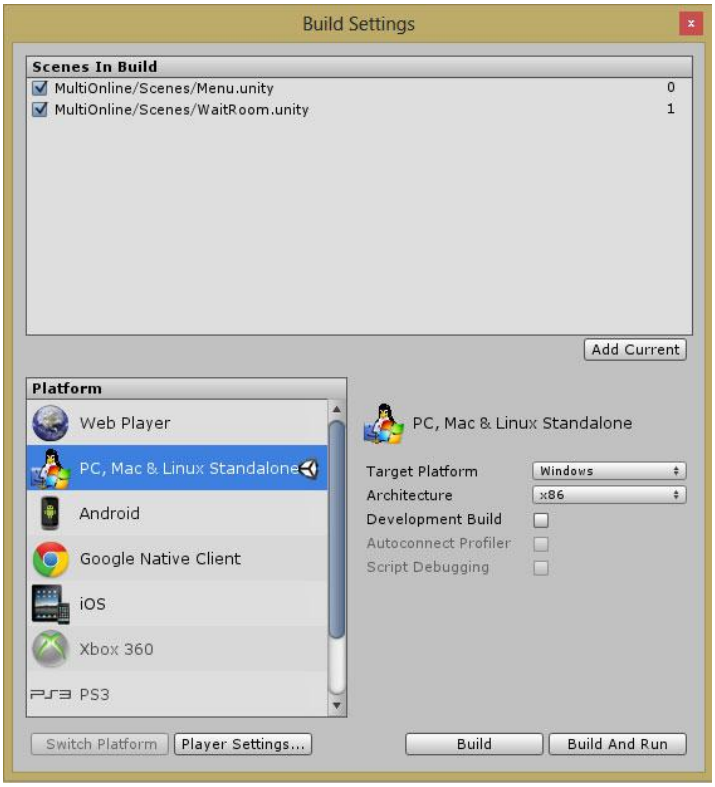


It opens the Build Settings panel. Click on the button "Add Current" in order to add the current scene on the "Scenes in Build" :

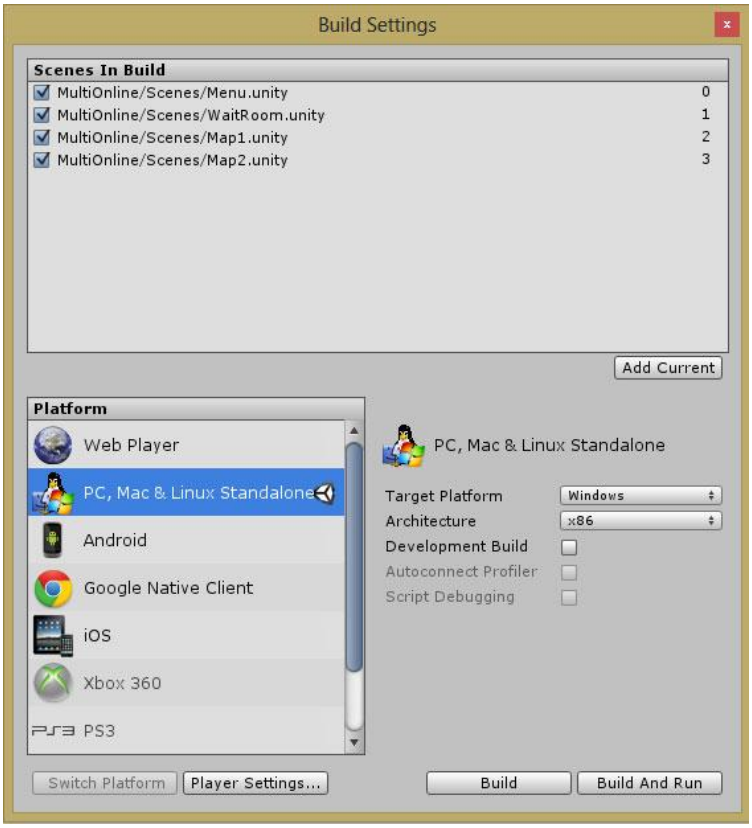


Each scene of the game must be on this list, else they cannot be loaded from the script.

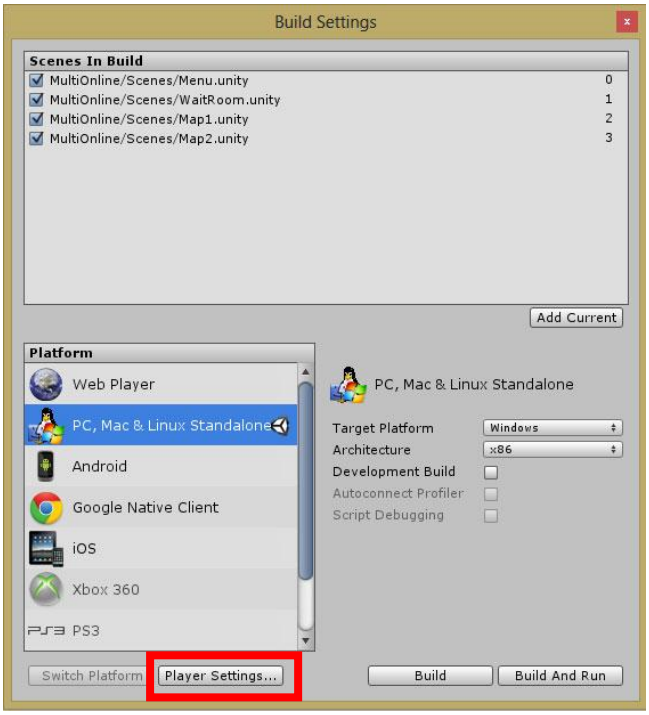
So now, let open the Build Settings panel, and open the WaitRoom scene.
Once the scene is opened, click on "Add Current" :



And do exactly the same thing for the two remaining scenes (Map1 and Map2).
Once you have finished, the Build Settings panel must look like that :



Now, click on the button "Player Settings..." :



The player Settings panel is now open in the inspector.

Check the box "Run in Background" : this will allow you to try the multiplayer with many windows open in the mean time on your computer.
You can also check the box "Resizable Window", I find that's very useful when I work with many windows but it is not mandatory.

Now, you can click on the "Build" button, so that we will try the multiplayer with two windows : an host an a guest.

Yan can name your builded game as you want. When it's done, you get a .exe file and a data folder :

Each time you'll want try your game with many windows, you'll have to build it first. You can go on "File/Build and Run" or use the shortcut "Ctrl+B".

It's now time to see how the multiplayer works.

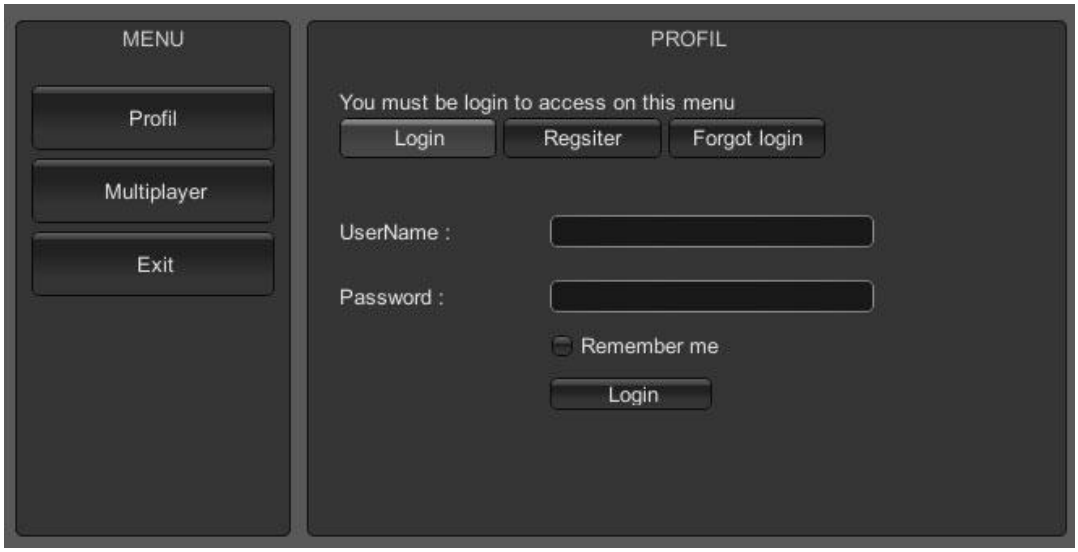
3. Running

3.1. The login menu

You can open directly your .exe file or open the scene "Menu" on your editor and click on "Play".

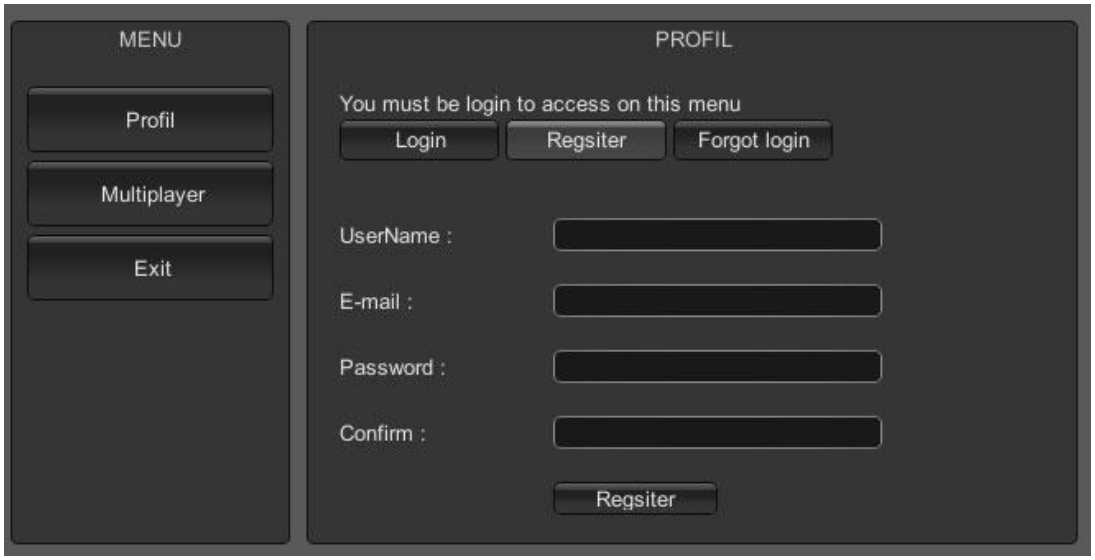
On your left, you have three buttons, but currently, if you try to click on "Profil" or on "Multiplayer", the login panel will appears.
The login panel contains tree menus buttons :

- Login : here you have to login with your username and password. If you check the box "Remember me" your login parameters will be saved and the fields will be prefilled the next time.

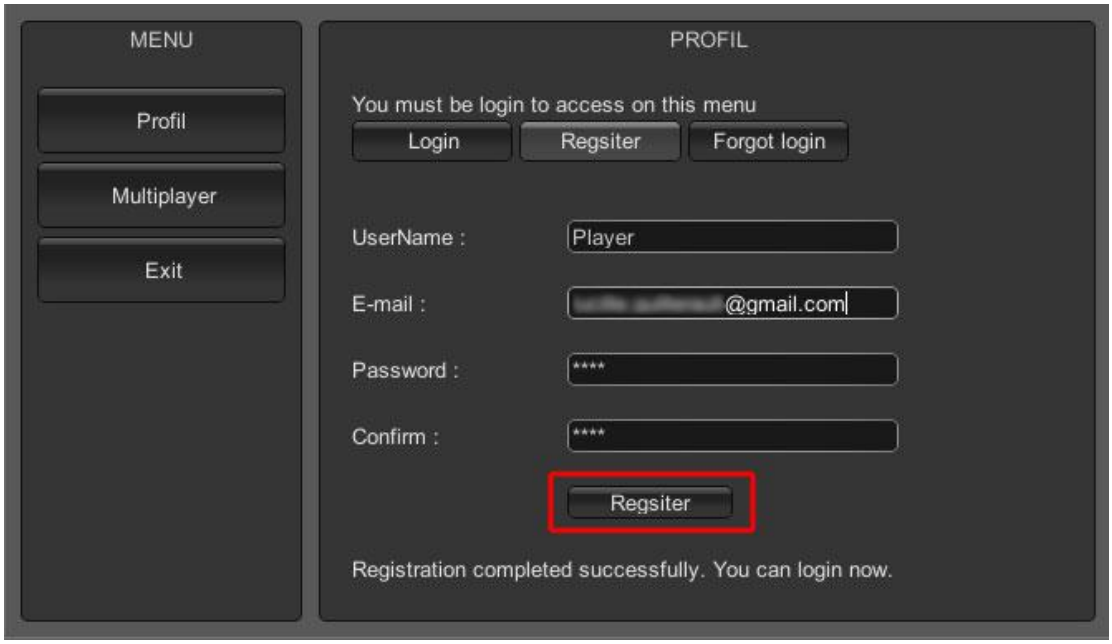


But we have currently no account, so let's click on "Register".

- Register : here you can create a new account. The data of this account are saved on your database, on the users table.

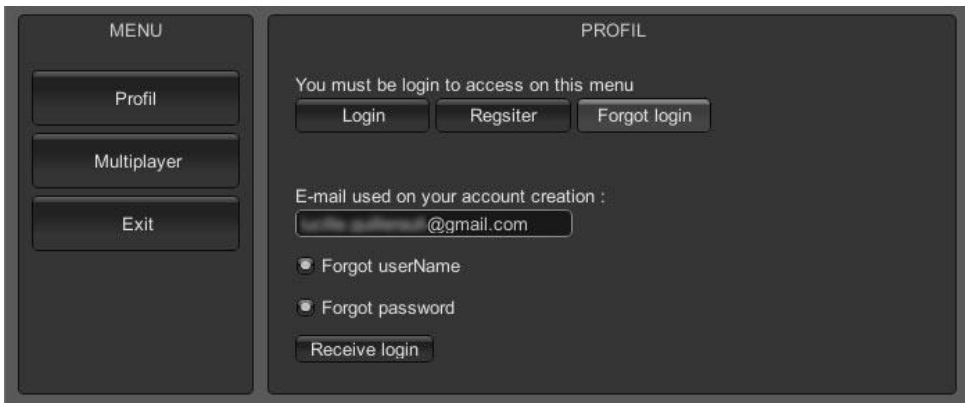


So you can now fill this form and click on register (I advise you to put your real e-mail address, so that you'll can try the automatic messages), when it's done, a confirm message will appear on the bottom of the page :



We could now login, but before let's see the last one login menu.

- Forgot login : if you have forgot your login or password, you can ask it here. Enter your e-mail and check one of the two boxes and click on "Receive login" :



If you have checked the two boxes, you will receive two e-mail.
The first will give you the username you used for create your account :

Your userName

Hello

You are registred on SuperGame with the following userName :
Player

Best regards,
SuperGame team

The second will give you your new password :

Your password

Hello

This is your new password for login on SuperGame :
19359

Once you'll be login, you could change this password from the profil settings.

Best regards,
SuperGame team

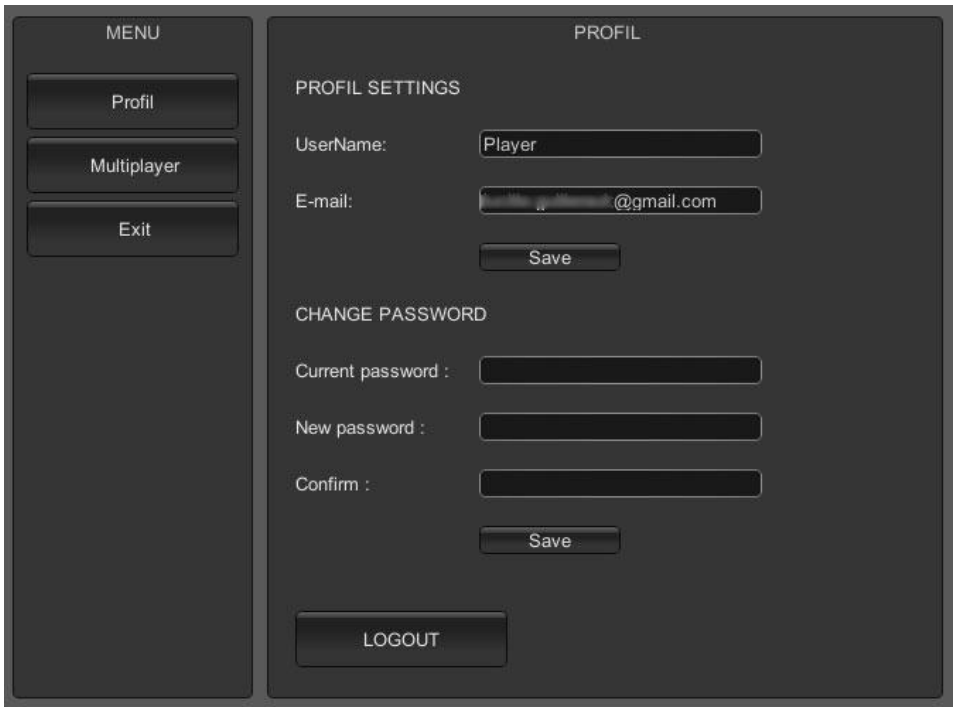
Note : as seen on this documentation part 2.4 (page 8) the content of these e-mail can be changed on the web script *MOUser.php*.

It's now time to login, so click on login, fill the form with your login data and click on login :

3.2. The main menu

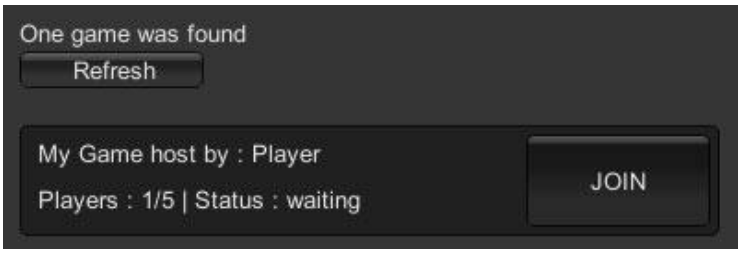
When you are login, you can go on the differents menus.

- Profil : here you can change your username, your e-mail or your password. You have also the "Logout" button at the bottom of the page :

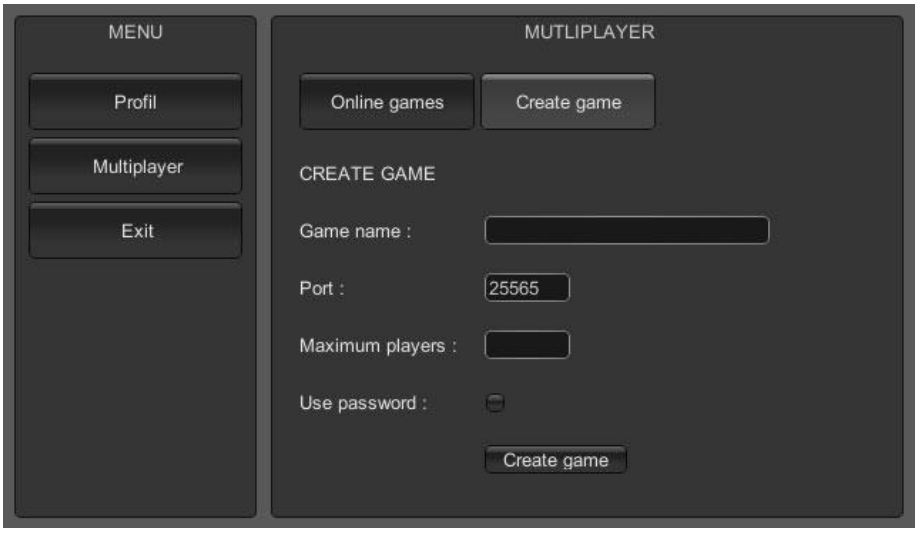


- Multiplayer : you have here two other buttons :
    - Online games : show all open games (so, it doesn't show the full game and started game if we have chosen that we cannot join them).

If there is an open game, the game list looks like that :



- Create game : for create and host a game. You have to fill for fields :



  - Game name : the game's name :P
  - Port : the game's port, 25565 by default, but it can be changed
  - Maximum players : the maximum number of players who can join the game

- Use password : if you check this box, the player have to enter a password to join the game. If the box is checked, it look like that :

- Important note : so that players can join a game, the host has to open the game's port on his router, else, nobody else of the players from his network would can join his game.

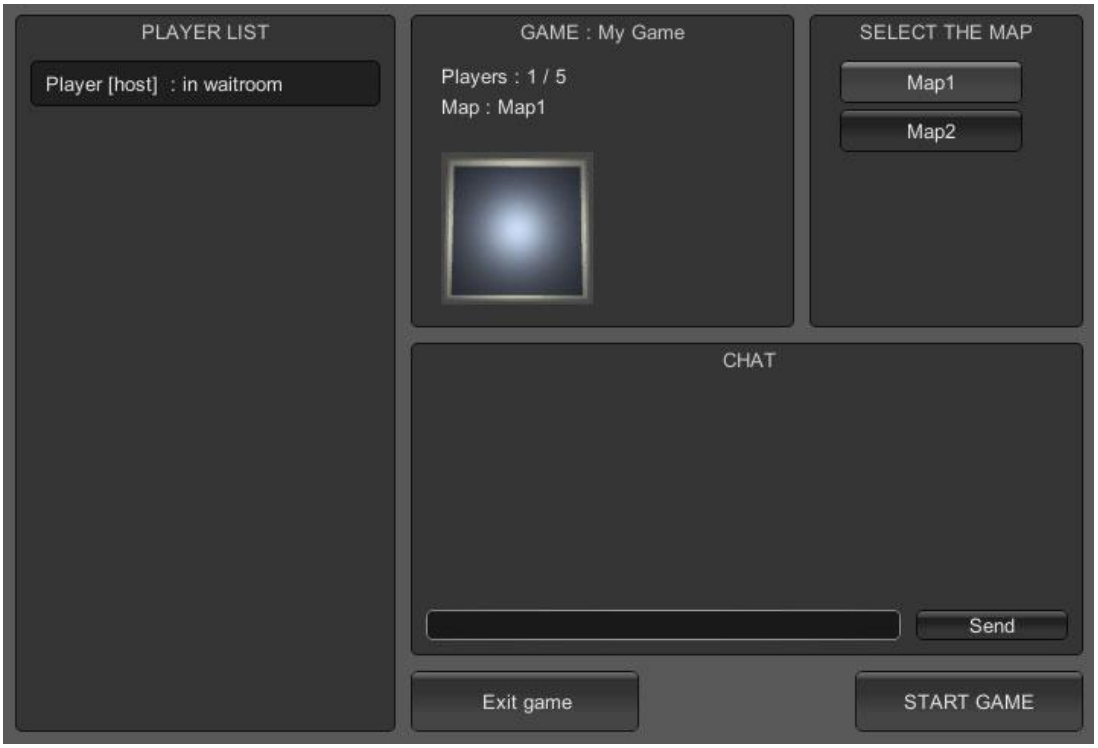
- Exit : the last button, as its name suggests, it's for exit the program

Since we have seen everything about the menu, we're go to create a game.
So click on Multiplayer/Create Game, full the form fields and click on "Create Game" :

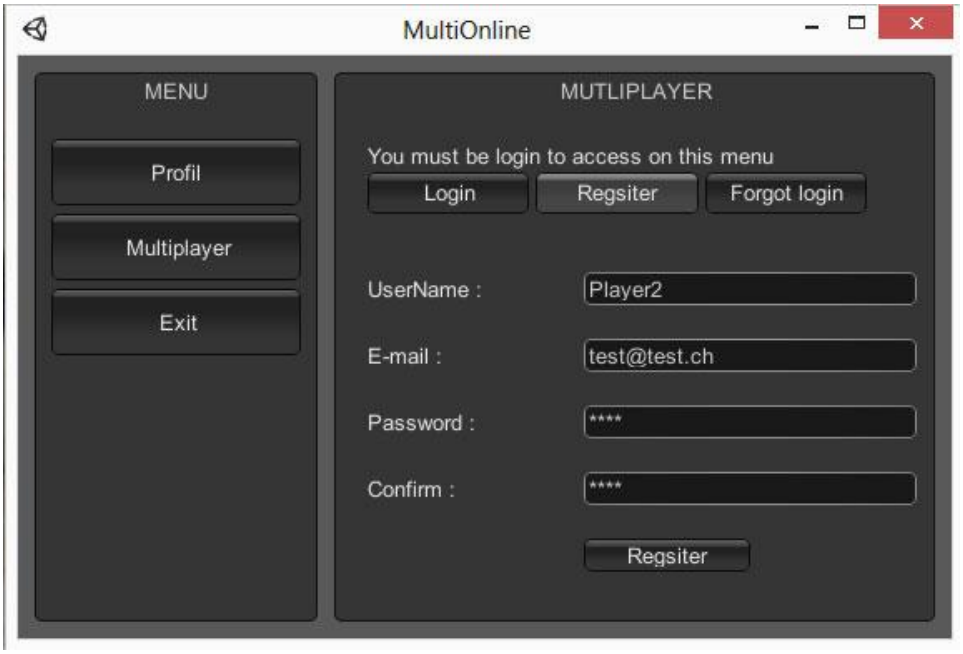
You are now in the waiting room

3.3. The waiting room

The waiting room shows the players list, the game's settings and, since we are the host of the game, we can choose the map here and start the game :

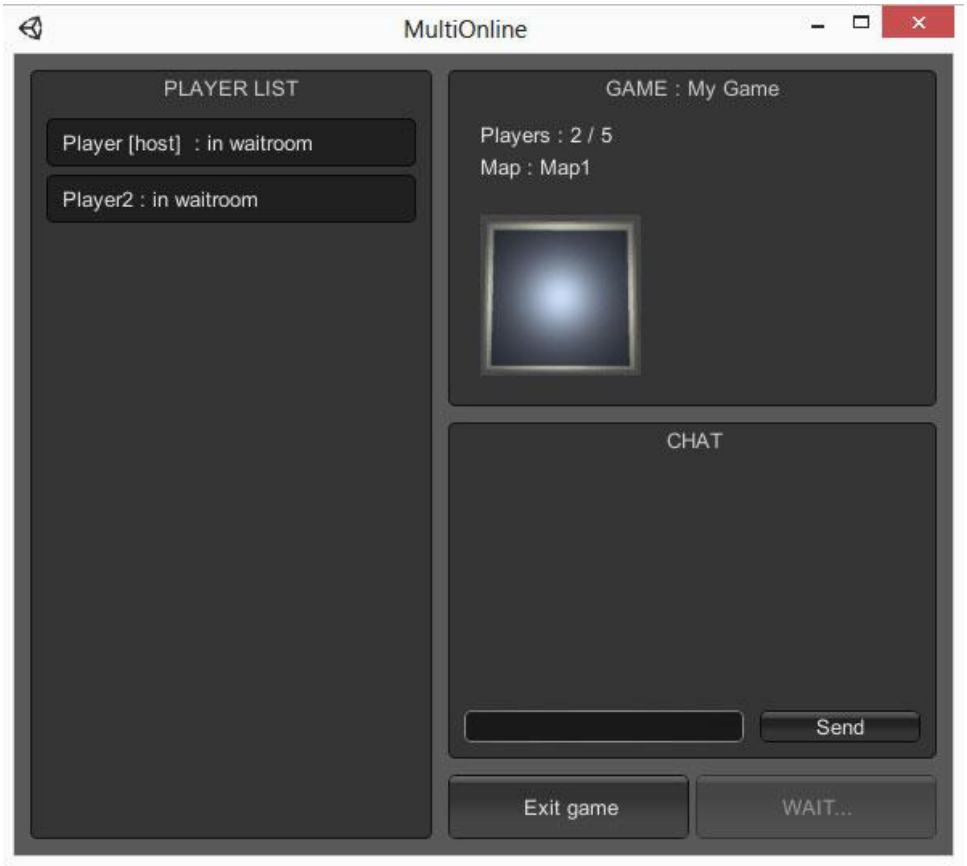


Now, open your .exe file for join your game from another window.
Create a new account and login.

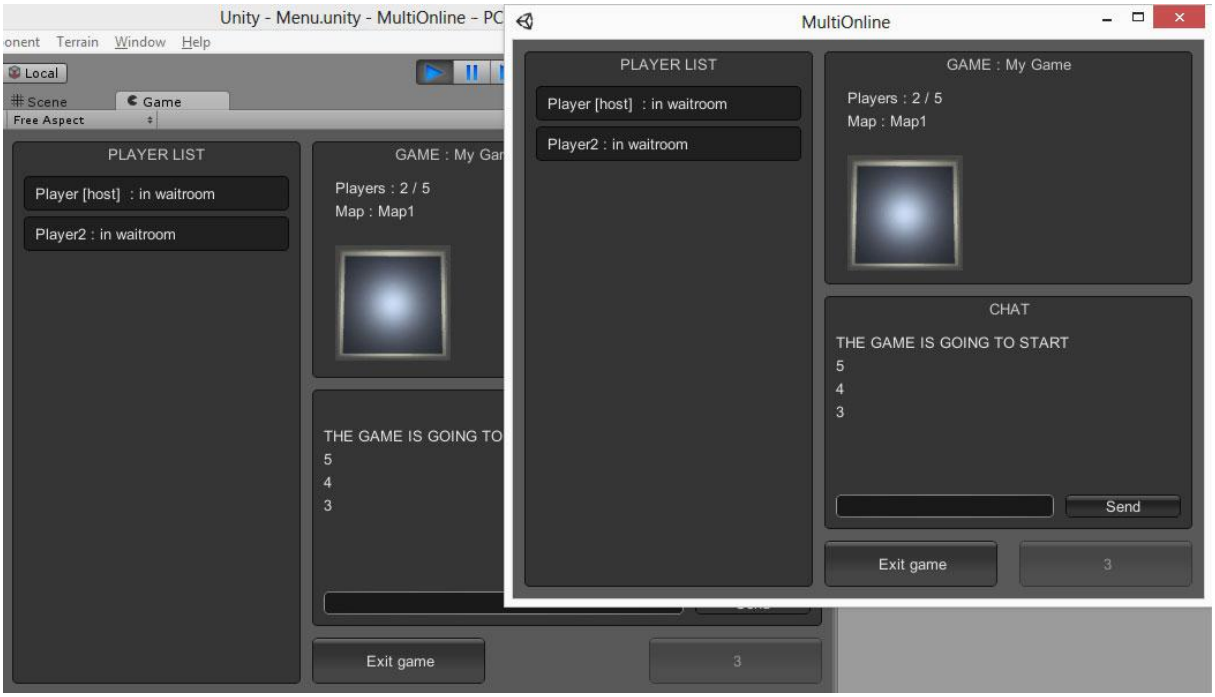


Once you are login, go on Multiplayer/Online game, and you must see the game you have created just before on the list :

Click on join. You can see the waitRoom as a guest. It's quite the same, you just cannot choose the map and cannot start the game:



Now, go on the window which has created the game and click on "START GAME".
A timer of 5 seconds begins and the map is loaded, and the players are spawned.

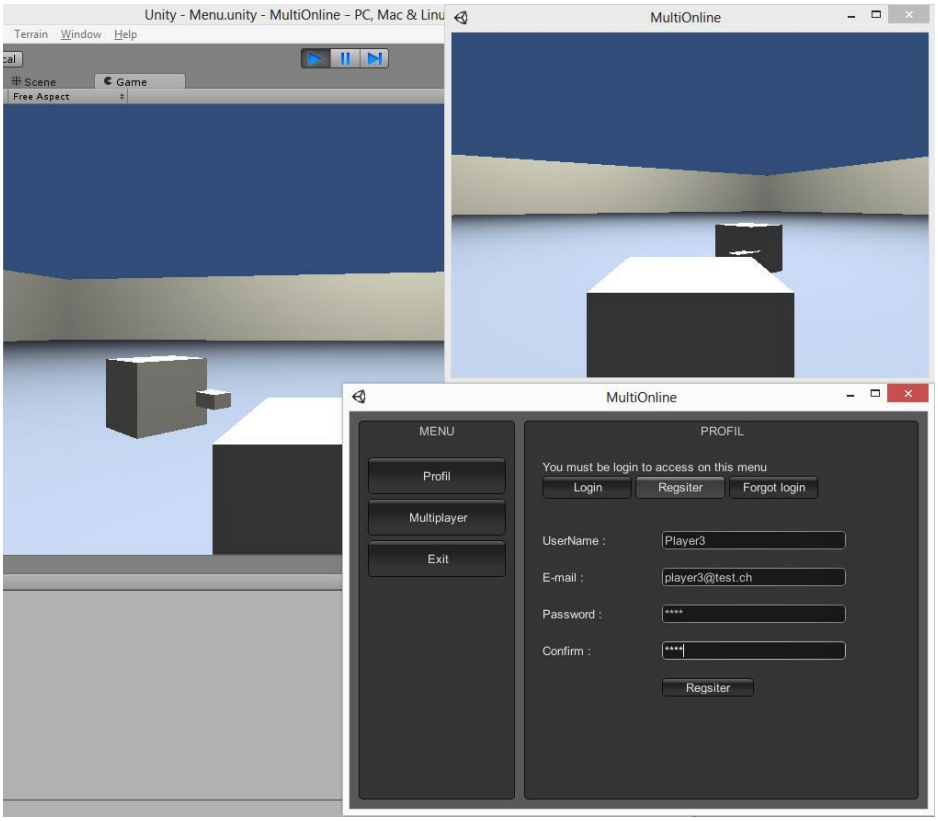


3.4. The map

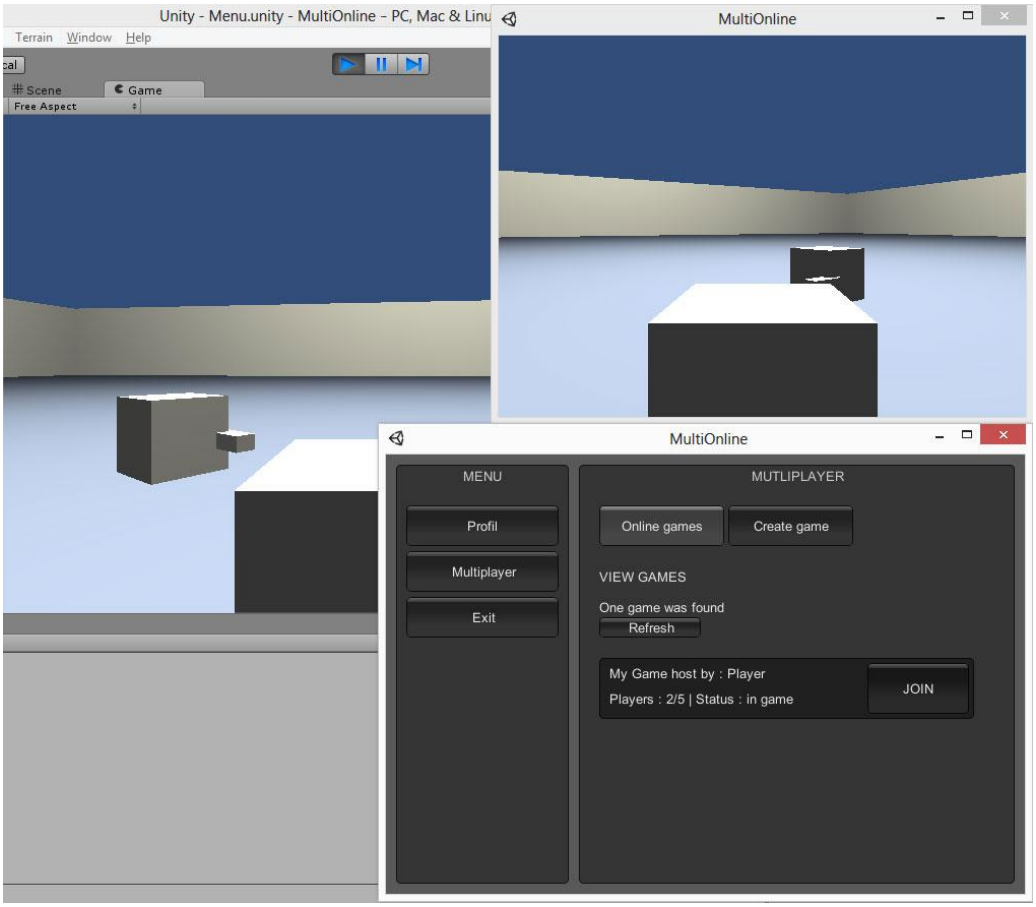
You can now try the multiplayer on a very nice blue map, with two wonderful cubes :P

You can add a third player on your game with opening the game on a new window.

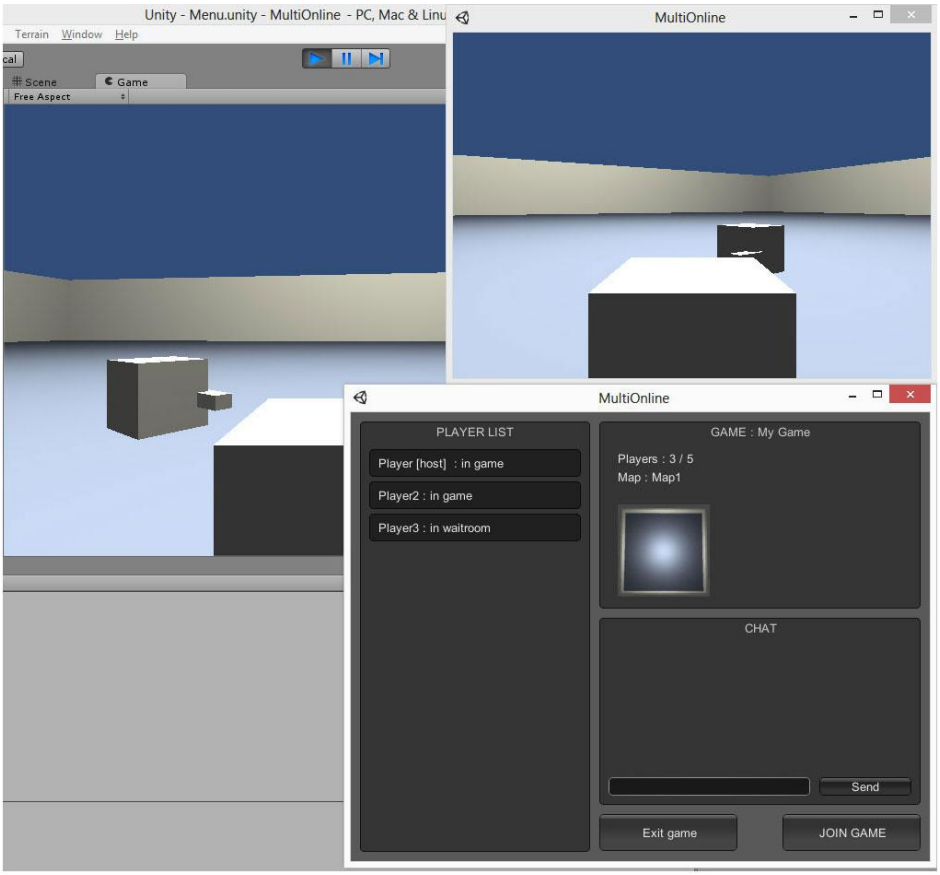
Create an account for this new player :



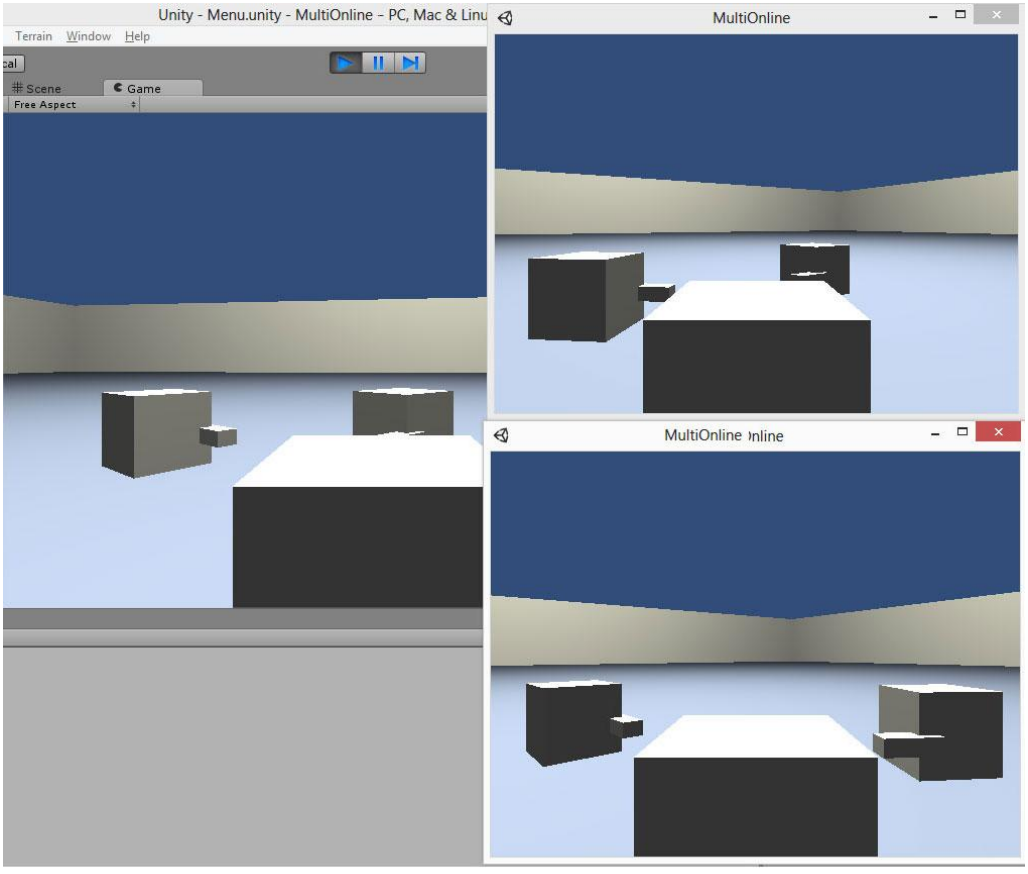
Find the game on the game list (click on "Join") :



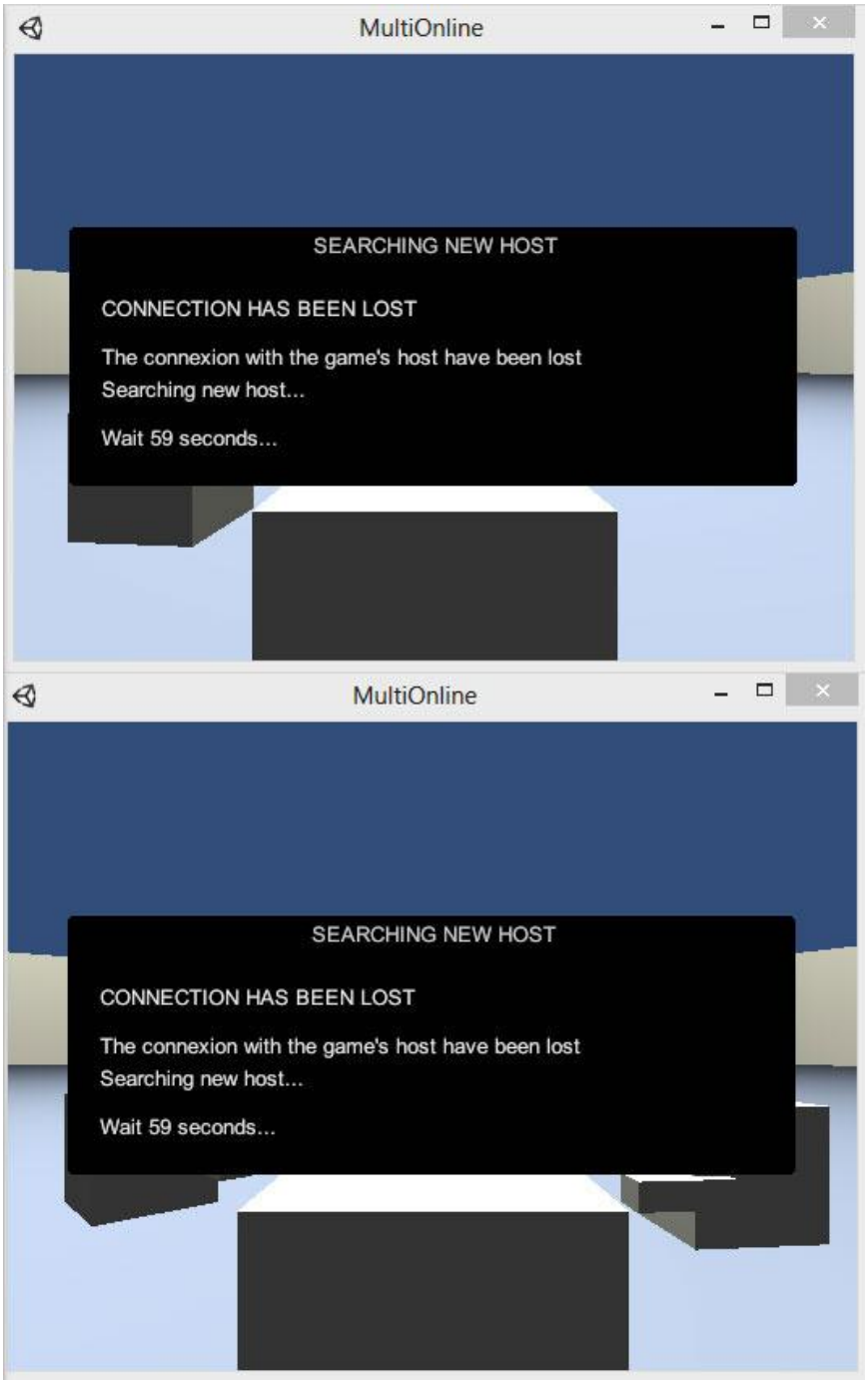
Join the game (click on "Join Game") :



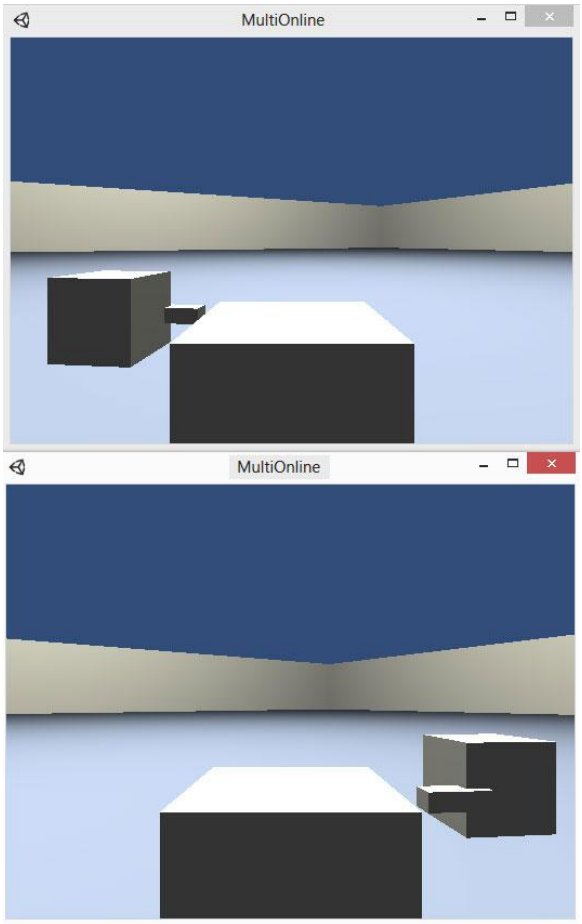
And you've got now three wonderful cubes...



For finish, we'll try the host migration. For that, exit the game on the window which is host. The two other player will receive a message which says that we are searching a new host :



After few seconds, one of the players becomes the new host, and the other is connects on him as guest, and the game plays again with these two players :



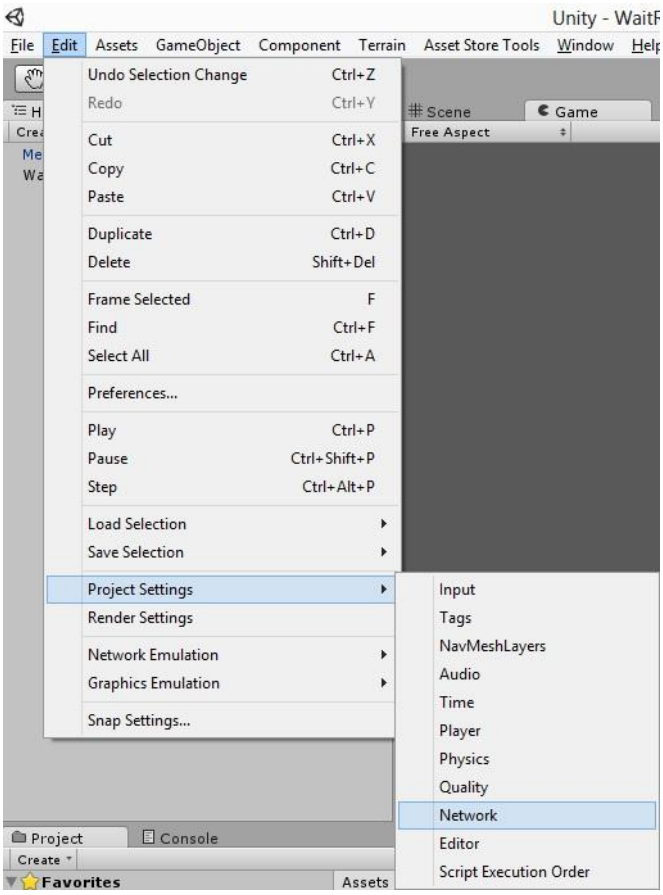
Note : The "Esc" key exits the game for come back on the main menu (it works on the waitroom too).

4. Settings

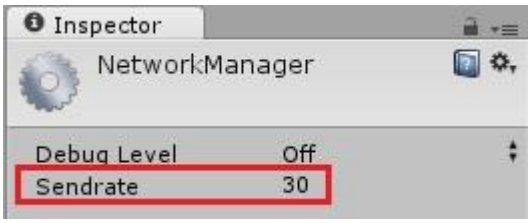
Many things are editable directly from the inspector, and all the scripts' texts can be edit.
So we will see the differents place where we've got a parametrable inspector.

4.1. The sendrate

To modify the sendrate, click on Edit/Project Settings/Network :



This will open the sendrate setting on the inspector. By default Unity set it at 15, but you can put more :

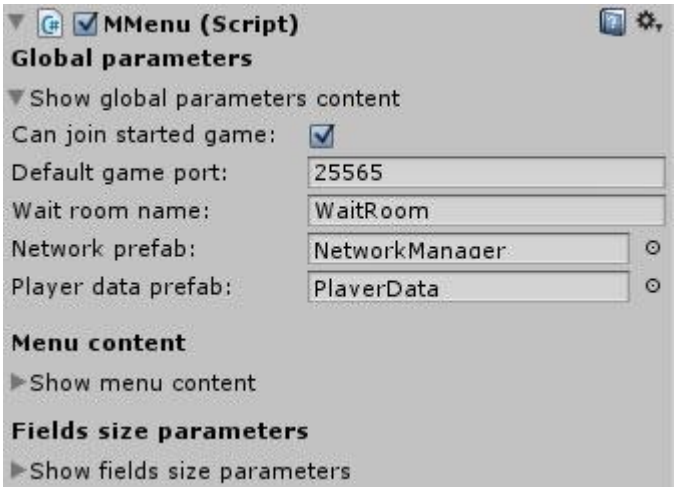


(I have put 30 for the example, but you can put more, in fact I usually put 60 in this field).

4.2. The Menu scene

Open the "Menu" scene and click on the game object named "Menu", you have two scripts attached on this object : MMenu and MOMenu.

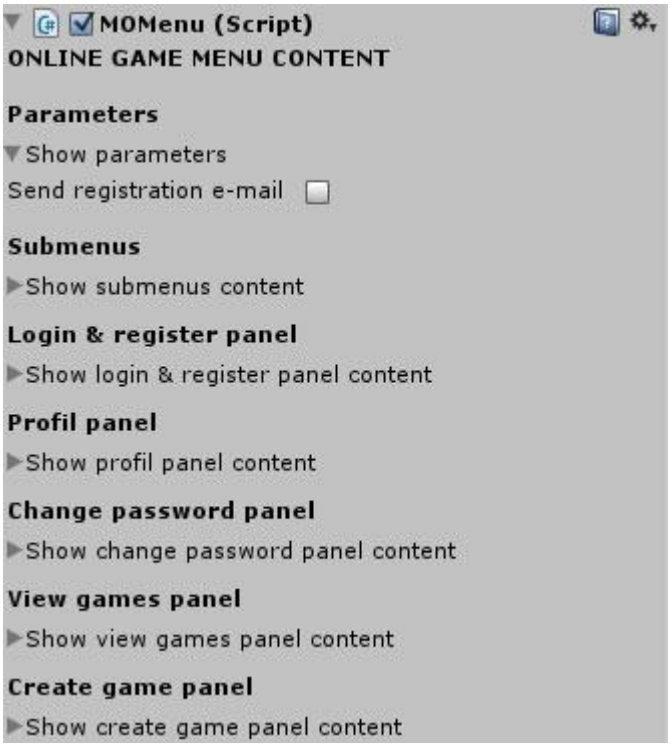
On MMenu, you have the following settings :



- Global parameters :
    - If players can join a started game or not
    - The default game port : this value is used to pre-fill the field "port" in the menu "Create game"
    - The waitroom scene name (in case you would change the waitroom's name, you'll have to fill this field with the new name of the scene)
    - The network prefab (in case you would change it for using your own networking script)
    - The playerData prefab (in case you would change it for using your own script)

- Menu content : the titles, the buttons' text and some global messages
- Fields size parameters : you can here change the size of the buttons, titles, form fields and labels

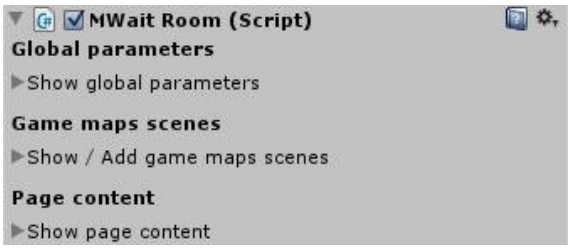
On MOMenu, you have the following settings :



- Parameters : here you can choose if you want send a confirmation e-mail to the players when they create an account
- Submenus : the text of the submenus buttons
- Login & register panel : all the text and messages of the login panel (which contains the menus "Login", "Register" and "Forgot login")
- Profil panel : all the text and messages of "Profil" panel
- Change password panel : all the text and messages of "Change password" panel
- View games panel : all the text and messages of "Online games" panel
- Create game panel : all the text and messages of "Create game" panel

4.3. The WaitRoom scene

Open the "WaitRoom" scene and click on the game object named "WaitRoom", you have these following settings :



- Global parameters:

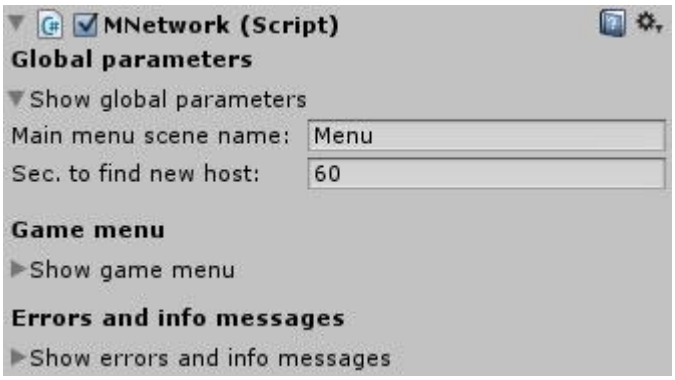
  - If you want to use a timer when the host starts the game. If you check this box, you can also choose :
  - The timer's time
  - If the timer displays on the chat or not (if not, it displays only on the load button)
- Game maps scenes :

  - If you want to display the maps screen on the waiting room, and choose the map screen size.
  - For add a new game map : click on "Add new map", this will create a new field where you'll have to enter the name of your scene and put the image of your map (as a texture - this field is optional). After that, the host could choose your new map from the waitroom
  - For delete a game map : just click on "Delete map x". After that, the map will not more appear on the waitRoom (but it will not destroy your scene)
- Page content : all the texts of the scene

That's all for the scenes. There are no editable things on the maps game scene since the goal is that you change them. The other editable settings are on the prebabs.

4.4. The NetworkManager prefab

Go on the prefab folder and click on the prefab named "NetwokManager". From here, you have these following settings:



- Global parameters
    - The main menu scene name (in case you would change it)
    - The maximum time (on seconds) for find a new host during a host migration (when the host losts his connexion). After this time, the script stop to search a new host and the player receive a message which says that it's not possible to continue the game, since nobody can host it.
- Game menu : all the texts that can appear on game (during host migration or when the player try to exit the game).
- Error and info messages : the text of all errors who come from the network. These text will appear on the main menu scene, but they are send by the network script, that's why they are set here.

4.5. The Spawns prefab

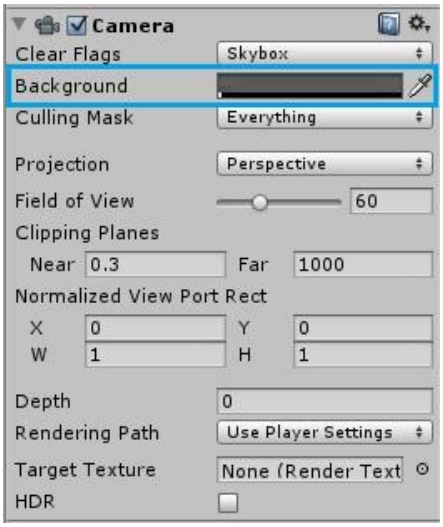
Go on the prefab folder and click on the prefab named "Spawns".
There is only one thing to set here, but that's a very important thing : the player prefab.
So, if you change the player, it's here you have to refer your player's gameObject.

4.6. The GameChat prefab

Go on the prefab folder and click on the prefab named "GameChat".
There is only one little thing to set here : the text of the send button.

4.7. The MenuCamera prefab

Go on the prefab folder and click on the prefab named "MenuCamera".
This prefab is used on the main menu scene and on the waitroom scene so that it's easy to change the background color of these scenes.



5. Customization

For finish, we will see few examples about how customize this package to create your own game.

5.1. Change the player

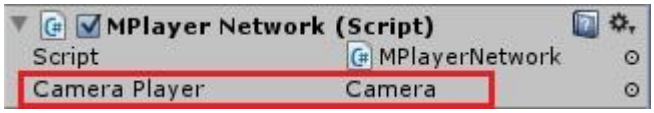
The player is one of the first things you'll want to change.

The step one is to create your own player prefab. There are just some little things he must have for working with the package :

- A network view (Add Component / Miscellaneous / Network view). The network view State Synchronization must be on "Reliable Delta Compressed" :

- The script *MPlayerNetwork* : this script manages the player around the network and disabled the other network players. So you must attach it on your player.

- Your player's camera must be attached on the script *MPlayerNetwork* :



- When your player is finish, you have to save it as a prefab, and click on the "Spawns" prefab. From here, drop your new player gameObject on the field "Player prebab" :

- You can change or remove the script *MPlayerMove*. This script is not necessary for the application's working (provided that you use another move script). But you should look at it and maybe use his "*Start()*" and "*Update()*" functions : these functions disable the player movement in some situations (host migration, exit game panel open...), so you could find it useful. So you could keep this script and change just the function *MovePlayer()* and add your new functions.

Important note : if you change the player movement script, or add new scripts on you player, you must disable these scripts for the others networkviews. For that, go on the function "*Start()*" in *MPlayerNetwork,* line 18 for disable the scripts for the others networkviews. The disable instruction is this one :

```
this.GetComponent<THE NAME OF YOUR SCRIPT>().enabled = false;
```

5.2. Use your own maps

Create new map is very easy.

- The Spawn

  When you map is done, you just have to put the "Spawns" prefab on it.

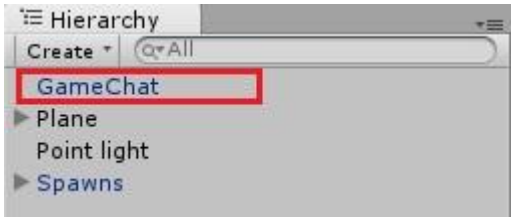
  This prefab contains all the spawns points for your map.
  You can add as much spawn point as you want (and name it as you want too, they just have to be children of the "Spawns" gameObject). You can place them where you want, but slightly higher than the floor of your map (else the player could pass through).

  Note : the differents spawns points can be named as you want, but you must not change the name of the "Spawns" prefab.

- The Game chat

  If you want to have a chat on your map, you just have to put the "GameChat" on it.

That's just an option, you can do as you want.

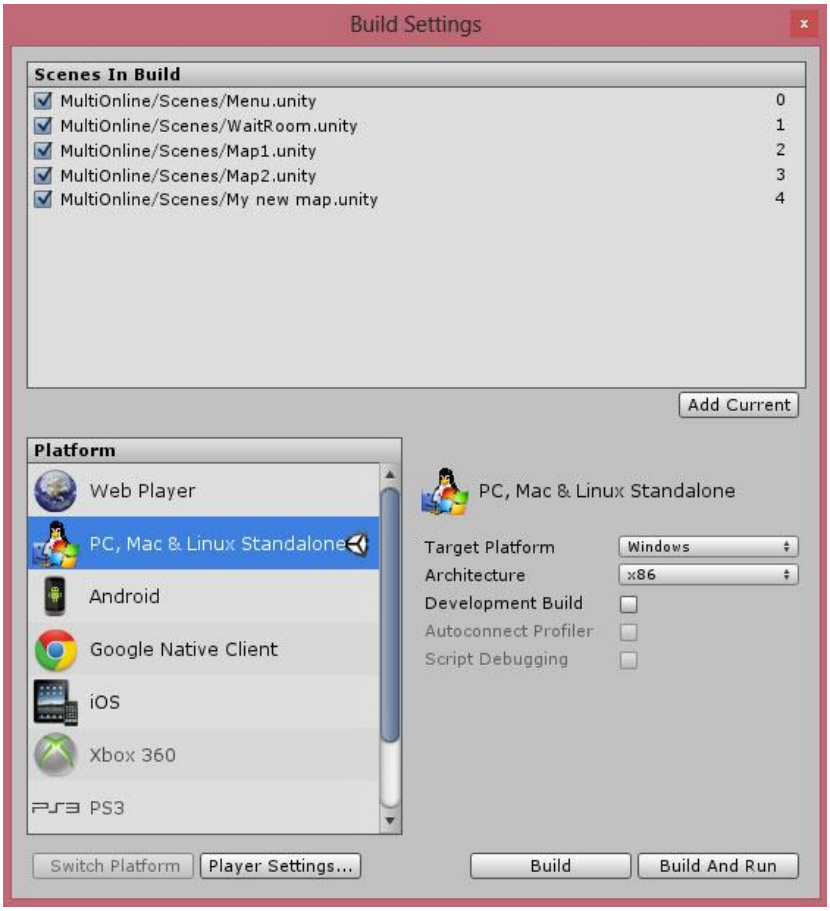


- Add the new map on the game
  Now, your map is ready. You just have to add it on the game.

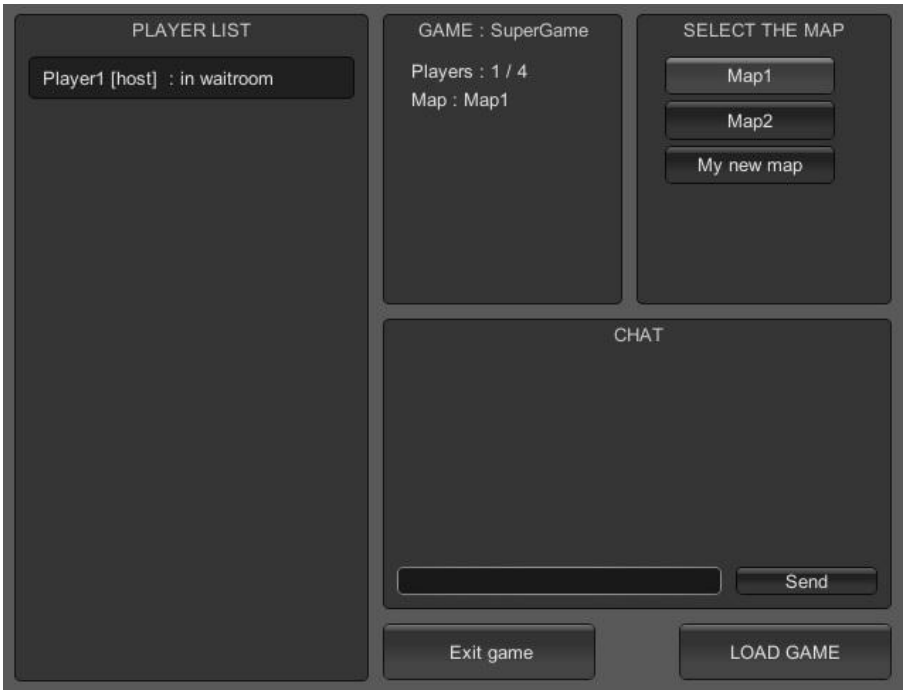
  - First, open the "WaitRoom" scene, click on the "WaitRoom" gameObject and, on the inspector, add your map on the "Game maps scenes", as it's explain on part 4.3 (click on "Add new map", and enter the name of your map's scene and the screen of your map on the new field) :

    Don't forget to save the scene after change something on the inspector.

  - Second, open the build settings on Unity (File/Build Settings...), add your new scene on the "Scenes in Build" (as show on part 2.6) and click on "Build" :



That's all, your new map will now appears on the waitroom and can be choose by the host for use it in game (in our example we didn't put the screen of the map) :



5.3. Change the menu or the waitroom

If you don't like the way how the menu or the waitroom appears, you can go on their scripts and change the onGUI function.

- For the sidebar : open the script *MMenu* and go on line 169
- For the main menu : open the script *MOMenu*, the menu display is separate on different functions (which begin all by "Display"), line 232 to 652
- For the waitRoom : open the script *MWaitRoom* and go on line 89
- For the game panel (panel which appears during host migration or when the player try to exit) : open the script *MNetwork* on go on line 166
- For the game chat (if you use it) : open the script *MGameChat* on go on line 23

For the main menu and the waitRoom, you can also use the "MenuCamera" prefab to change the background color, as show on part 4.7.

6. FAQ

6.1. Players cannot join to my game

If you create a game, don't forget to properly open the game's ports on your server, else, nobody (except the people in your network) can connect to your game.

Check also your antivirus and firewall who will probably send you a warning message when you'll create a game. So, authorize your application on your antivirus and firewall parameters, for private and public communications.

6.2. The host migration doesn't work

When the host leaves the game, all the guest lose the connexion. The script will then try to find a new host among the remaining players in game. If a player have opened the game's port on his server, he'll become the new host, and all the other player will connect on him. But if neither player have his port open, nobody can become the new host, and the game ends.

Note : when the game is already started, the new host must be in game. Players in the waiting room cannot become new host when if game has already been started.

6.3. The game is lagging

If you have some lag problems, you can try to change the Sendrate, as explain on part 4.1. You can also change the field "Observed" on the player's NetworkView component : the "Observed" field is currently "Player(Transform)", you can try to replace it by "Player(Rigidbody)". For that, you just have to go on your player prefab, and drop his

"Rigidbody" component on the field "Observed" of the NetworkView component.

6.4. My animations are not synchronized

When you use animations across a network game, you must call the animations from a RPC function, else, the animation will not be played for the other players.

Example in C# for one animation :

```
// Call this function when the event which triggers your animation happen
void RunAnimation() {
      // Call the RPC function SyncAnimation on all players
      networkView.RPC("SyncAnimation", RPCMode.All);
}

[RPC]
void SyncAnimation() {
      // Play the animation
      animation.Play("the name of your animation");
}
```

Example in C# for many animation :

```
/* Call this function when the event which triggers your first animation
happen */
void RunAnimation1() {
      // Call the RPC function SyncAnimation on all players with parameter 0
      networkView.RPC("SyncAnimation ", RPCMode.All, 0);
}

/* Call this function when the event which triggers your second animation
happen */
void RunAnimation2() {
      // Call the RPC function SyncAnimation on all players with parameter 1
      networkView.RPC("SyncAnimation ", RPCMode.All, 1);
}

[RPC]
void SyncAnimation(int index) {
      if(index == 0){ // If the index parameter is 0 :
            // Play this animation :
            animation.Play("the name of your first animation");
      } else if(index == 1){ // else, if the index parameter is 1 :
            // Play this animation :
            animation.Play("the name of your second animation");
      }
      /* And you can add here as much animation your want by using other
         index numbers */
}
```

Note : on the examples, you can replace *animation.Play()* , by *animation.CrossFade()* if you want to use this function.