

Videojoc per Android

Unity 3D



LageVione

Javier Lage Giménez
Enginyeria Tècnica d'Informàtica de Gestió

Consultors: Jordi Almirall López i Marc Domingo Prieto
11/01/2015



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

GNU Free Documentation License (GNU FDL)

Copyright © 2014 Javier Lage Giménez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (Javier Lage Giménez)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel•lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>LageVione</i>
Nom de l'autor:	<i>Javier Lage Giménez</i>
Nom del consultor:	<i>Jordi Almirall López i Marc Domingo Prieto</i>
Data de lliurament (mm/aaaa):	<i>01/2015</i>
Àrea del Treball Final:	<i>Desenvolupament aplicacions dispositius mòbils (Android)</i>
Titulació:	<i>Enginyeria Tècnica d'Informàtica de Gestió</i>
Resum del Treball (màxim 250 paraules):	
<p>El projecte escollit tracta de fer un video joc de caràcter lectiu per a Android fent servir Unity. El video joc estaria orientat a un públic infantil, justificant així la simplicitat del mateix.</p> <p>Consisteix en fer volar un avió i buscar lletres pel mapa. Un cop trobades les haurem d'encertar mitjançant un formulari. En xocar amb una lletra pretenc reproduir el so de la lletra i mostrar un formulari amb botons per permetre encertar-la.</p> <p>Per tant es tracta d'un joc 3D i hauré de programar l'entorn, l'avió i donar-li les propietats per desplaçar-se i xocar.</p> <p>El joc es connectarà a un servidor MySQL mitjançant un Web Service per enregistrar usuaris i les lletres que va encertant.</p>	
Abstract (in English, 250 words or less):	
<p>The object of the project is to make a video game for Android using Unity software. The video game would be aimed at the children, justifying the simplicity of it.</p> <p>It consists in flying a plane reaching some letters which are in the air. When the plane crashes with a letter, the sound of this letter is reproduced and it is shown three different answers to make easier to the user to find the correct answer.</p> <p>Therefore it is a 3D game and I had to set the environment, the air and give the properties to move and collide.</p> <p>The game is connected to a MySQL and web server in order to allow the user's registration and storing the letters that the users find.</p>	
Paraules clau (entre 4 i 8):	
Joc, avió, llegir, aprendre, lectiu	

Contingut

1.	Introducció	6
2.	Objectiu del projecte	7
3.	Unity 3D	9
	Una mica d'Història.....	9
	Jocs desenvolupats amb Unity.....	10
	Components de unity: Organització del projecte	11
	Les escenes.....	12
	Els GameObjects	12
	Els Components	13
	Els Scripts	14
	La Càmera	15
4.	El meu video joc.....	16
	menu_scene.....	16
	login_scene	19
	island_scene.....	25
	El terreny.....	26
	L'avió	28
	La esfera	32
	Els sons de les lletres.....	38
5.	Sobre el servidor	39
	Objectiu.....	39
	Com és el servidor virtual	39
	SSH: Putty i WinSCP	40
	Apache: Web server.....	41
	MySQL Base de dades	42
	Disseny de la base de dades	43
	Els fitxers html i php.....	44
	config.php	46
	register.php.....	47

login.php	48
saveletters.php	49
askletters.php	51
6. Conclusions	53

1. Introducció

Aquest projecte consisteix en fer un video joc 3d amb Unity per a dispositius Android. Android és el sistema operatiu per telèfons mòbils per excel·lència. Ens dona un mercat potencial molt gran i eines per a distribuir el nostre joc com el Google Play.

Amb aquest projecte pretenc crear un video joc senzill per familiaritzar-me amb el programari i el mètode.

Un dels aspectes molt importants d'aquest projecte és que vull connectar el joc amb un servidor mysql per emmagatzemar dades rellevants així com usuaris i puntuacions. D'aquesta manera pur dinàmicament afegir contingut al joc sense crear una nova versió de la aplicació.

Punts objectius del projecte:

- Aprendre a crear jocs 3d emprant el programari Unity.
- Aprendre a donar i gestionar el moviment dels objectes (sprites) del joc mitjançant el sistema de scripting de Unity.
- Connectar el joc amb un servidor mysql per permetre el registre, la validació d'usuaris i la informació rellevant per puntuacions etc.
- Muntar i configurar el servidor Apache i mysql emprant algun sistema de hosting online.
- Aprendre a generar el .apk i executar-lo a terminals mòbils.
- Publicar una aplicació al Google Play.



2. Objectiu del projecte

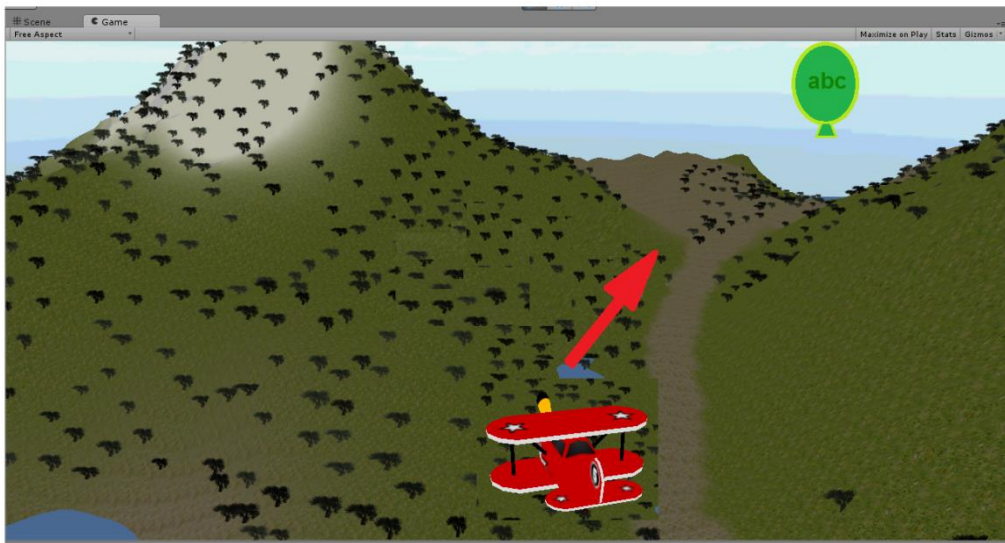
Donat que és el primer videojoc que faig, l'he dissenyat per a un públic infantil. La seva finalitat és que, sota supervisió dels pares, els nens puguin aprendre les lletres de l'abecedari jugant.

En qualsevol moment els pares poden accedir a un registre de les lletres que els nens han anat encertant durant partides anteriors. D'aquesta manera poden confirmar que les han après y les recorden.

El mètode consisteix en fer volar un avió per un escenari en 3 dimensions per anar trobant lletres que estan amagades, penjant en l'aire. Per controlar l'avió emprarem un joystick situat al centre de la pantalla.

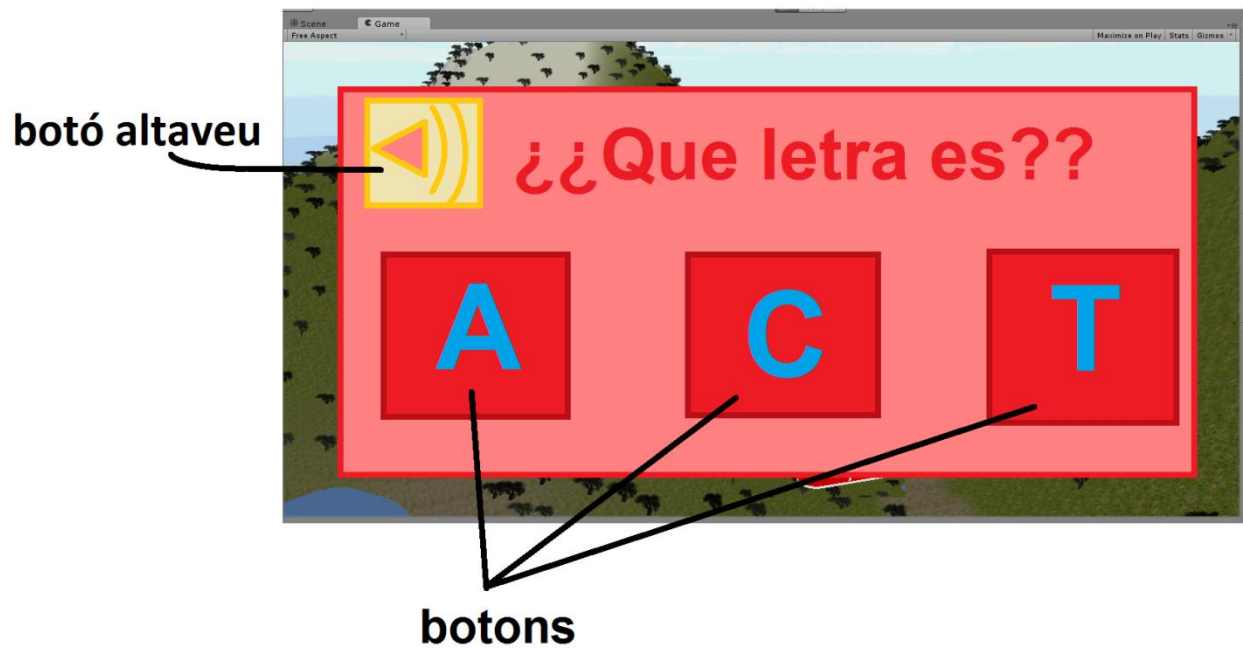


En trobar una lletra mostraré una finestra des d'on podem reproduir un so per encertar la lletra en qüestió entre unes quantes proposades.



Aquest és el concepte original de la finestra que emprerà el nen per encertar la lletra que acaba de trobar jugant.

- El botó en forma d'altaveu serveix per reproduir el so de la lletra que tenim que encertar.
- Amb els botons en forma de lletra podem encertar la lletra que reproduïm.



- Els botons s'han de poder generar aleatòriament.
- Un d'ells ha de contenir la lletra del globus.
- En encertar la lletra correcta l'hem d'emmagatzemar a base de dades.
- Un cop encertada la lletra s'ha de consultar a la base de dades quina serà la següent lletra a buscar i tornar a posicionar el globus en una posició aleatòria del mapa per continuar el joc.

Tot això o orquestrarem emprant Unity 3d.

3. Unity 3D



Unity 3D es el programari que he emprat per a la creació del video joc. Em proporciona les eines necessàries per a gestionar tots els elements que conformen el joc.

Per a unity els jocs son un conjunt d'escenes i objectes que fem interactuar entre ells mitjançant l'ús de scripts. Els objectes són el personatge, la càmera, el terreny, els objectes del joc com poden ser els arbres, les cases o portes, etc.

Els scripts son els que donen vida al que passa al joc. Els fem per detectar els inputs de l'usuari i programar accions. També podem programar menús, botons, o disparadors per detectar per exemple col·lisions entre objectes i provocar alguna acció al respecte.

Una mica d'Història

Unity Technologies va ser fundada al 2004 a Copenhague després de crear el seu primer joc GooBall per a MAC OS X.



El joc no va tenir èxit però els 3 fundadors van reconèixer el valor i potencial del motor gràfic i les eines de desenvolupament que van emprar per a la creació del video Joc. Per tant els fundadors es van proposar alliberar-lo creant un motor gràfic que tots poguéssim emprar a un preu assequible.

Van llançar la primera versió de Unity a la conferència per a desenvolupadors d'Apple al 2005 per a funcionar i produir jocs exclusivament a la plataforma Mac OS.

Actualment Unity està disponible com a plataforma de desenvolupament per Windows i Mac OS X i permet crear jocs per a Windows, OS X, Linux, Xbox 360, PlayStation 3, Playstation Vita, Wii, Wii U, iPad, iPhone, Android y Windows Phone. També es poden desenvolupar videojocs de navegador per a Windows i Mac.

La versió actual es compatible amb 3ds Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D, Cheetah3D, Adobe Photoshop, Adobe Fireworks y Allegorithmic Substance.

Existeixen 2 versions, la gratuïta i la professional . Amb la versió professional disposem de suport, més eines, efectes i textures.

La versió gratuïta és suficient per iniciar-se i crear jocs fantàstics i molt complets.

Jocs desenvolupats amb Unity

Aquests són alguns dels jocs desenvolupats amb unity un cop es va alliberar el software.

Battlestar Galactica Online (2011)

Family Guy Online (2011)

Dead Lab (2012)

Endless Space (2012)

Prime World (2012)

Guns of Icarus Online (2012)

DOT - Space Hero (2013)

Temple Run (2012)

Game of Thrones: Seven Kingdoms(2013)

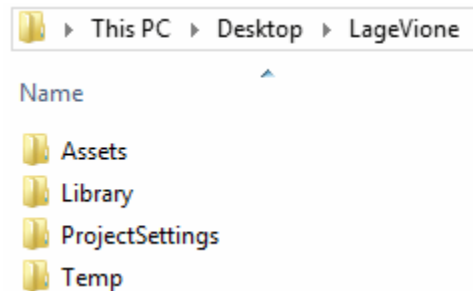
Into The Gloom (2014)

The Forest (2014)

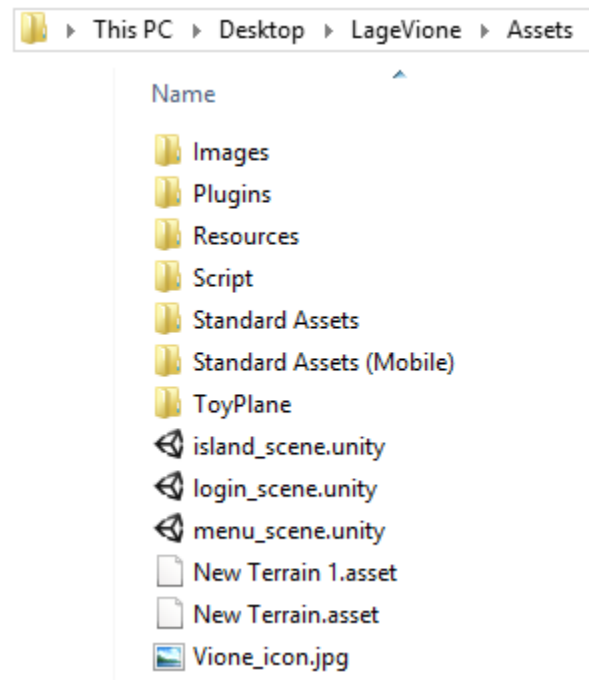
Font: [http://es.wikipedia.org/wiki/Unity_\(software\)#Software_desarrollado_con_Unity](http://es.wikipedia.org/wiki/Unity_(software)#Software_desarrollado_con_Unity)

Components de unity: Organització del projecte

Els projectes d'Unity no són com els projectes de Visual Studio o Eclipse. Quan obrim un projecte d'Unity el que fem en comptes d'obrir un arxiu del projecte determinat o un tipus de fitxer de la solució és apuntar Unity a una estructura de carpetes formada per les carpetes Assets, Library, ProjectSettings i Temp.



La carpeta Assets conté tots els seus actius: material gràfic, codi, àudio; cada arxiu que inclogui en el seu projecte va aquí. Aquesta és sempre la carpeta de nivell superior d'Unity Editor.



La **carpeta Library** és la memòria cau local per als actius importats; manté totes les metadades dels actius.

La **carpeta ProjectSettings** emmagatzema les opcions que configuri des Edit | Project Settings. La carpeta Temp s'usa per als arxius temporals de Mono i Unity durant el procés de compilació.

Les escenes

Un video joc amb unity està organitzat per **escenes**. Tot el que s'executa al joc existeix a una escena. El joc serà una col·lecció d'escenes.

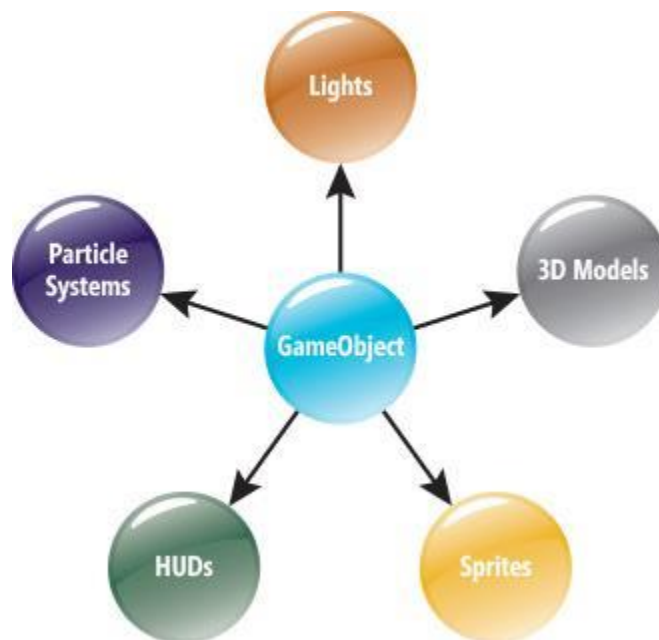
Un projecte pot tenir tantes escenes com vulguem. Tot i que podria no ser així, les escenes acostumen a ser les diferents pantalles i/o nivells d'un vídeo joc.

Les escenes estan formades per diferents **game objects**.

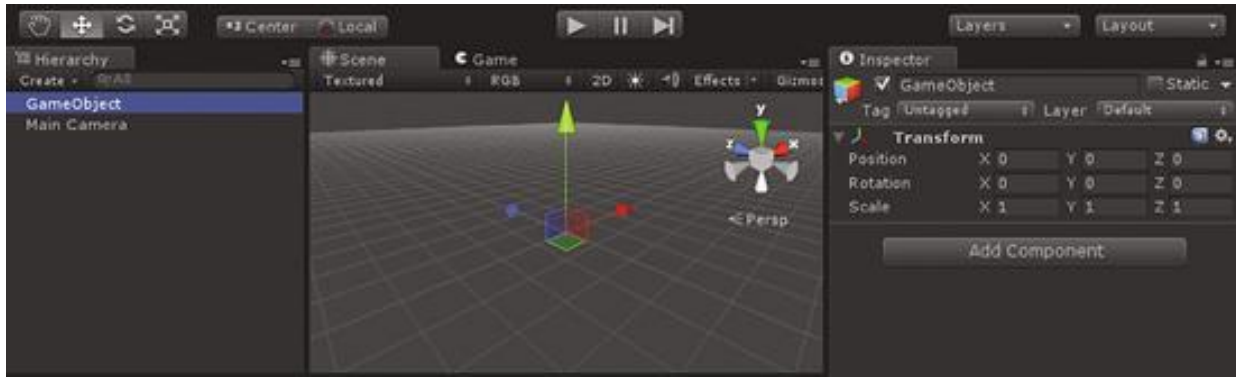
Els GameObjects

Casi tot en una escena serà un **GameObject**. És el mateix concepte que s'aplica en .Net o Java, totes les classes provenen de System.Object. És la classe bàsica.

Els objectes poden ser de molts tipus, no necessàriament gràfics. La càmera per exemple seria un objecte, les llums, el terreny, els formularis, o també podem emprar objectes per guardar dades i accedir a ells des d'altres objectes, etc.



Quan creem un objecte a Unity apareix a la finestra d'escena i Inspector però amb quasi totes les seves propietats buides.



És mitjançant l'ús de componenets que els **Game Objects** adquireixen propietats. Per defecte només hereten les propietats de **Transform** que són aquelles que permeten establir la posició de l'objecte, la rotació i l'escala.

Fins aquest punt encara no hem assignat una textura, imatge o model 3d a aquest objecte. Això o podrem fer emprant l'ús de **components**.

Els Components

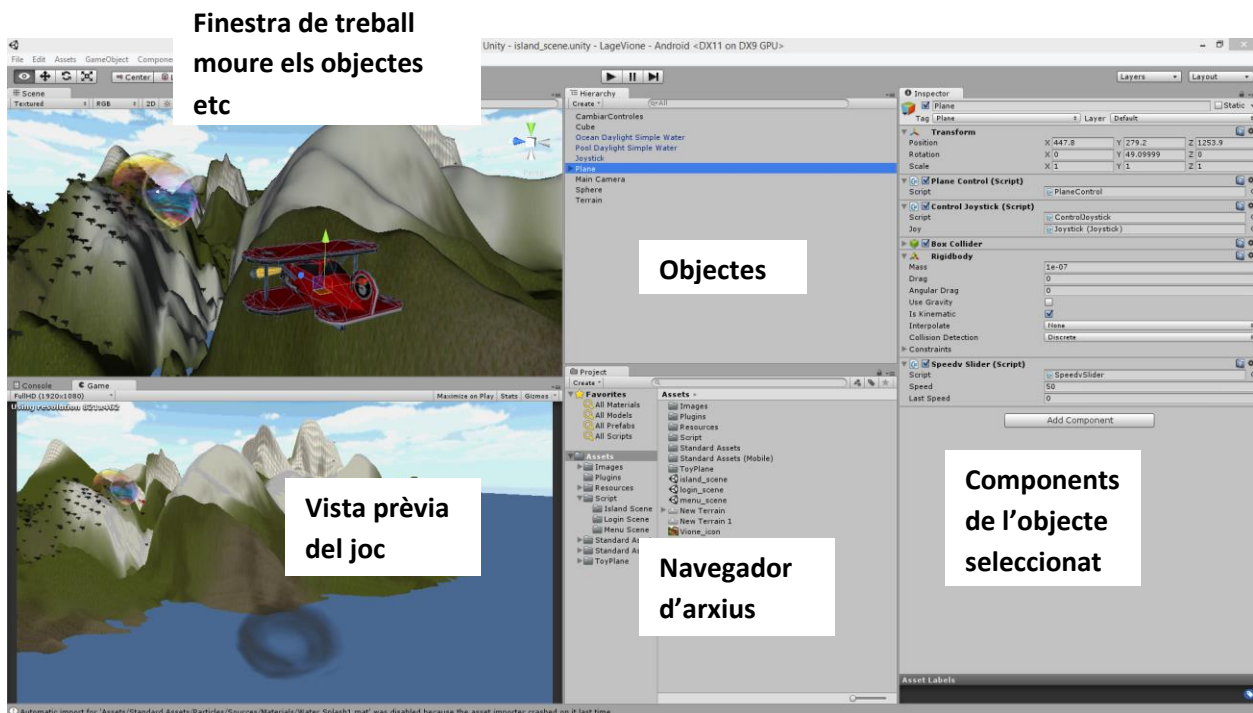
Els **Components** apareixen a la finestra Inspector quan seleccionem un **Game Object**.

Unity ens proveeix de components de tot tipus. Quan fem un component amb un GameObject estem fent un tipus **implement**. Aquest objecte adquireix les propietats del component que estem implementant.

- Unity ens dona components com els MeshRender i SpriteRender per dotar l'objecte de la capacitat de renderitzar-se i mostrar-se per pantalla.
- Hi ha components per a la funcionalitat d'àudio i càmera.
- Components relacionats amb la física (col·lisionadors i cossos rígids per gestionar automàticament la gravetat d'un objecte, etc)

Per assignar codi a un objecte hem d'utilitzar un component de tipus script.

Els components són els que donen vida als **GameObjects** ja que serveixen per afegir les funcionalitats desitjades. Mitjançant els scripts podem detectar les accions de l'usuari i moure **GameObjects** o activar-los. Llançar accions com reproduir sons, canviar propietats etc.



Els Scripts

Unity està basat en C++ natiu però els escrits poden ser escrits en C#, JavaScript o Boo Script. El codi dels escrits s'executa en Mono o Microsoft.Net Framework i que generalment es compila contínuament.

La estructura dels scripts per defecte està formada per una funció Start i una funció Update Però en podem crear tants mètodes com vulguem Dintre d'un script.

```

using UnityEngine;
using System.Collections;

public class lage_test : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

    private void algo () {
        //hacer algo
    }
}

```

El mètode **Start** es crida automàticament en instanciar-se un **GameObject** que te afegit un script com a component.

El mètode **Update** es crida automàticament a cada frame durant el joc. El número de frames per segon i per tant, el número de vegades que s'executi el mètode Update dependrà del rendiment del dispositiu on s'executi el joc.

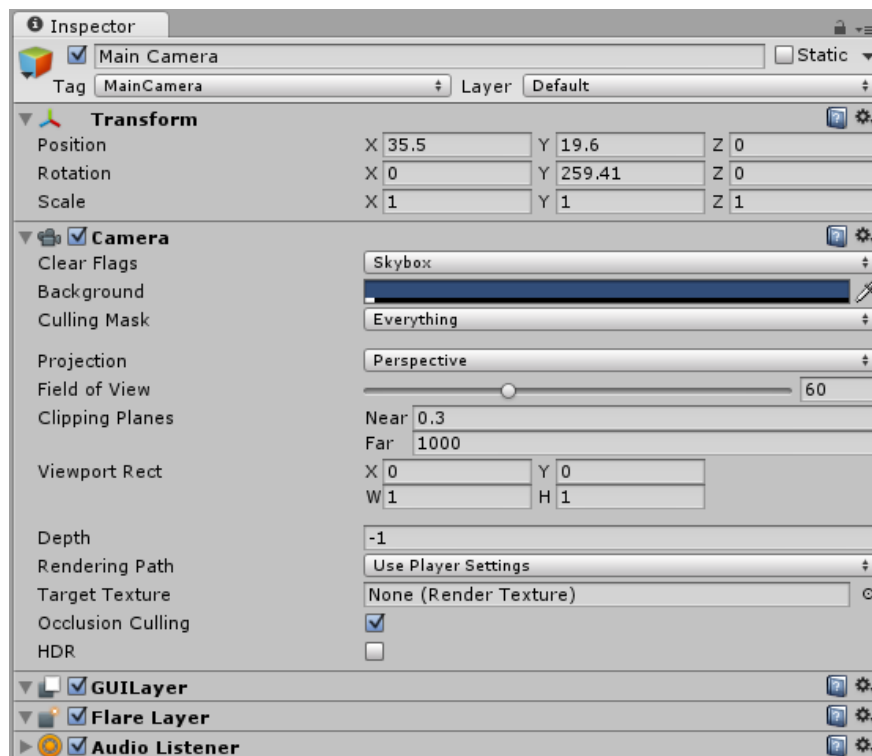
Això es molt important perquè hi accions que voldrem que s'executin en un moment determinat, no a cada frame però en canvi la comprovació per saber si hem d'executar una determinada funció si que ho haurem d'examinar a cada frame.

Per això utilitzarem molt els flags d'estats. Pot ser una variable de tipus booleà al mateix script o pot ser un valor determinat de qualsevol variable com per exemple, la posició de l'avió, etc.

La Càmera

La càmera és un objecte més del joc amb unes determinades propietats. Com son les Transform per establir la posició i rotació.

A la escena menú la càmera està fixa, no està animada i mostra el que te davant però a la escena de la illa la càmera s'ha animat amb un script fet en C# que obliga a la càmera a perseguir l'avió per el mapa.



4. El meu video joc

El meu video joc està format per 3 escenes.

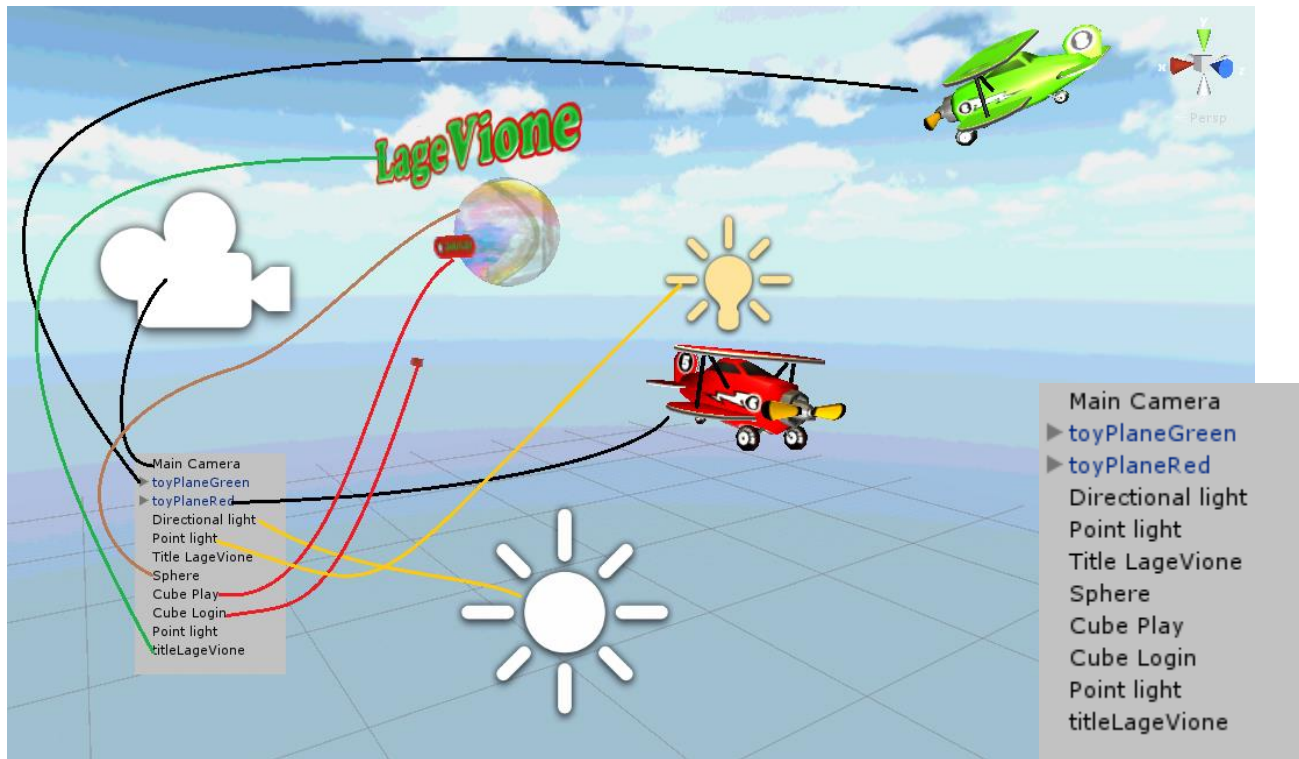
- island_scene.unity
- login_scene.unity
- menu_scene.unity

menu_scene

menu_scene és l'escena que apareixerà en obrir la aplicació. Ens permetrà anar a jugar directament si som un nen o anar a la escena per registrar-nos si som un adult.



Els objectes d'aquesta escena:



Tant la esfera com els avions tenen un script per rotar. La esfera rotaria sobre els eixos X i Y, y els avions només sobre els eixos Y. (son scripts diferents).

L'script de la esfera seria:

```
using UnityEngine;
using System.Collections;

public class gameobject_rotation_3d : MonoBehaviour {

    // called once per frame
    void Update () {
        // rotate the object around its X axis
        transform.Rotate(Time.deltaTime, 0, 1);
        // rotate the object around its Y axis
        transform.Rotate(1, Time.deltaTime, 0, Space.World);
    }
}
```

L'script més interessant d'aquesta escena es el que detecta quin objecte ha estat clicat amb el ratolí o tocat mitjançant la pantalla tàctil del dispositiu.

Aquest script s'anomena TouchAction.cs i està afegit al Gameobject: Main Camera.

```
using UnityEngine;
using System.Collections;

public class TouchAction : MonoBehaviour {
    string gameObjectName = null; //objecte tocat

    // Update is called once per frame
    void Update () {

        /*Code for mouseClick*/
        if (Input.GetMouseButtonDown (0)) {
            Ray toMouse = Camera.main.ScreenPointToRay (Input.mousePosition);
            RaycastHit raycastInfo;

            if(Physics.Raycast(toMouse, out raycastInfo, 500.0f)) {
                //Debug.Log(raycastInfo.collider.name);
                gameObjectName = raycastInfo.collider.name;
            } else {
                Debug.Log(" empty space");
            }
        }

        /*Code for touch screen*/
        int touchCorrection = 1;
        RaycastHit hit = new RaycastHit();
        for (int i = 0; i+touchCorrection < Input.touchCount; ++i) {
            if (Input.GetTouch(i).phase.Equals(TouchPhase.Began)) {
                // Construct a ray from the current touch coordinates
                Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(i).position);
                if (Physics.Raycast(ray, out hit)) {
                    //Debug.Log(hit.collider.name);
                    gameObjectName = hit.collider.name;
                }
            }
        }

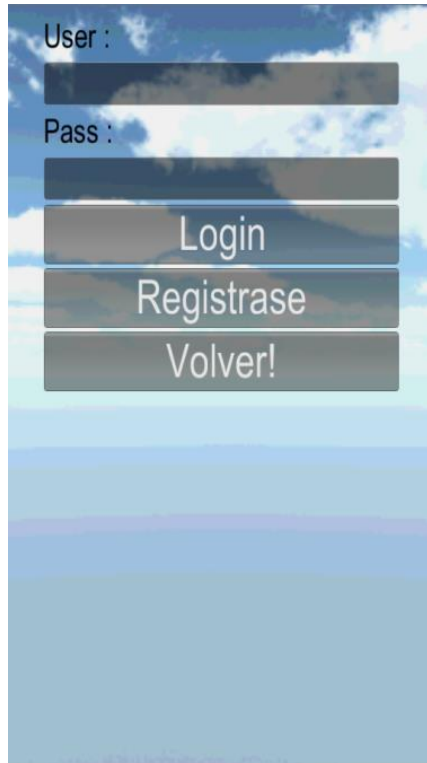
        /*Quina escena carregarem*/
        if(gameObjectName == "Cube Play") {
            Application.LoadLevel ("island_scene");
        } else if(gameObjectName == "Cube Login") {
            Application.LoadLevel ("login_scene");
        } else {
            //Nothing
        }
    }
}
```

El que faig resumidament és captar quina coordenada s'ha clicat o tocat i preguntar a la càmera que està mostrant en aquesta posició.

En funció de l'objecte clicat o tocat carreguem la següent escena.

login_scene

login_scene: és l'escena per al registre i validació d'usuaris orientada als pares dels nens que jugaran.



The screenshot shows a login form with a blue sky background. It includes a 'User:' label followed by a text input field, a 'Pass:' label followed by a text input field, and three buttons: 'Login', 'Registrase', and 'Volver!'.

El primer que veiem en carregar la escena es aquest formulari per a entrar com a usuari registrat.

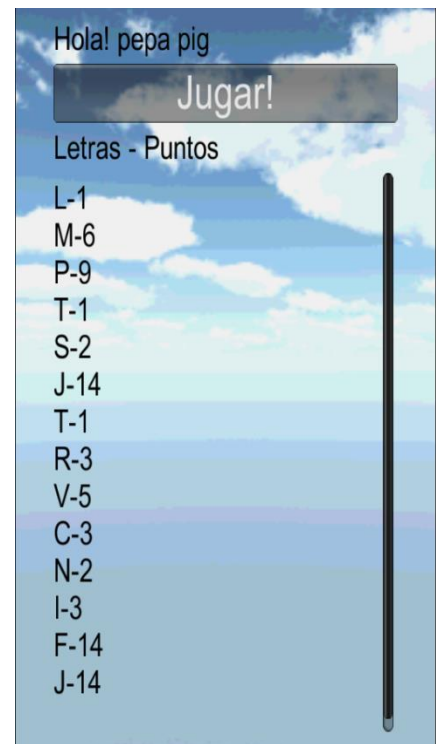
L'he fet amb javascript. Es pot rotar el terminal que el formulari es mostra correctament.

He aplicat un full d'estil per fer les lletres mes grans i de diferent color al per defecte.

En clicar a Login es crida per webservice al fitxer **login.php** del servidor enviant els paràmetres corresponents. El servidor valida si la informació es correcta o no.

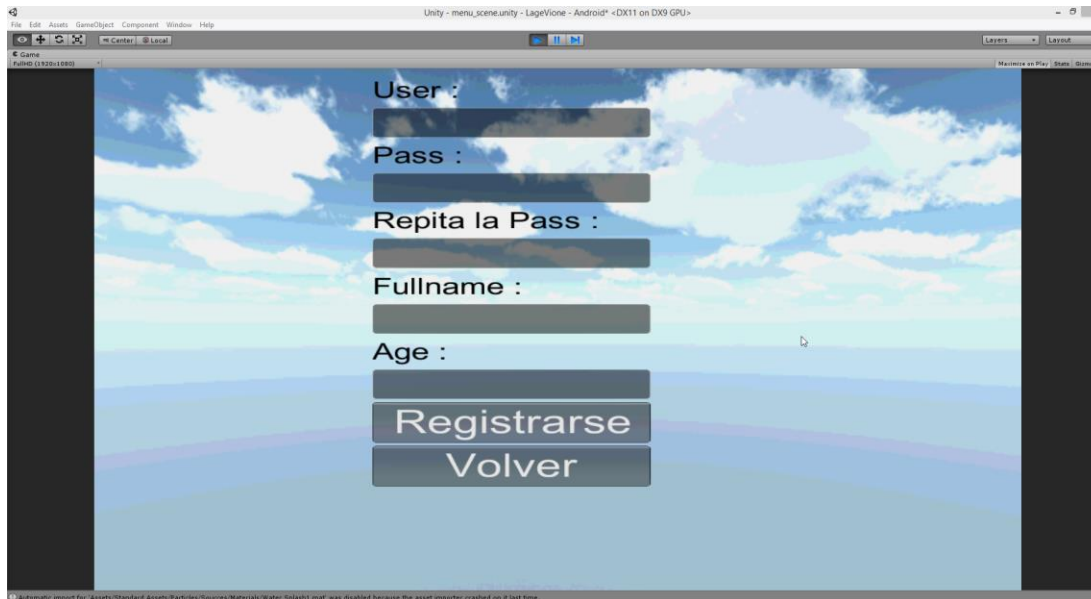
Un cop ens hem validat amb el nostre usuari correctament veurem el llistat de les lletres que hem encertat anteriorment. Per a mostrar les lletres es crida el fitxer **askletters.php** del servidor tot enviant l'usuari que s'ha validat.

El servidor retorna una cadena de text que un cop processada podem mostrar en un panell amb autoscroll.



The screenshot shows a game interface with a blue sky background. At the top, it says 'Hola! pepa pig' and has a 'Jugar!' button. Below that, it says 'Letras - Puntos' and lists the following items: L-1, M-6, P-9, T-1, S-2, J-14, T-1, R-3, V-5, C-3, N-2, I-3, F-14, and J-14. A vertical scrollbar is visible on the right side of the list.

En clicar a **Registrarse** s'amplia el formulari. Al tornar a clicar en Registrarse es crida per web Service al fitxer register.php del servidor enviant els paràmetres corresponents.



El script `login_unity_script.js` d'aquesta escena l'he fet amb javascript:

```
var User : String;
var Pass : String;
var RePass : String;
var Fullname : String;
var Age : String;
private var Msg : String;

var Host : String = "http://146.185.148.96/unity/";
var PHPUser : String = "login.php";
var PHPRegister : String = "register.php";
var PHPSaveLetters : String = "saveletters.php";
var PHPAskLetters : String = "askletters.php";
private var Texto : String;
private var Temp : String = "";
private var isNewUser : boolean;
private var isUser : boolean;
private var once : boolean = true;

//redimensionar el GUI usando un skin personalizado
var MySkin : GUISkin;

function OnGUI () {
    //añado skin
    GUI.skin = MySkin;
```

```

    if (Screen.width > Screen.height) {

        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new
Vector3(Screen.width / 1000.0f, Screen.height / 800.0f, 1));
        GUILayout.BeginArea(Rect(300,10,300,800));
    } else {
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new
Vector3(Screen.width / 600.0f, Screen.height / 800.0f, 1));
        GUILayout.BeginArea(Rect(50,10,500,800));
    }

    if(!isUser){
        GUILayout.Label("User :");
        User = GUILayout.TextField(User);
        GUILayout.Label("Pass :");
        Pass = GUILayout.PasswordField(Pass,"*" [0]);
        if(isNewUser){
            GUILayout.Label("Repita la Pass :");
            RePass = GUILayout.PasswordField(RePass,"*" [0]);

            GUILayout.Label("Fullname :");
            Fullname = GUILayout.TextField(Fullname);

            GUILayout.Label("Age :");
            Age = GUILayout.TextField(Age);

            if(GUILayout.Button("Registrarse")){
                if(RePass == Pass && Pass != "" && User != "" && Fullname
!= "" && Age != ""){
                    if(RePass.Length > 1){
                        PerformNewUser(User,Pass,Fullname,Age);
                    }else{
                        Msg = "2 caracteres como mínimo";
                    }
                }else{
                    Msg = "Revisa la contraseña o los campos vacíos";
                }
            }
            if(GUILayout.Button("Volver")){
                isNewUser = false;
            }
        }else{
            if(GUILayout.Button("Login")){
                if(Pass != "" && User != ""){
                    PerformLogin(User,Pass);
                }else{
                    Msg = "Revisa los campos";
                }
            }
            if(GUILayout.Button("Registrarse")){
                isNewUser = true;
            }
            if(GUILayout.Button("Volver!")){
                once = true;
                Application.LoadLevel("menu_scene");
            }
        }
    }

```

```

    }
    GUILayout.Label (Msg);
}else{
    GUILayout.Label (Texto);
    if(MiEstado.miEstado != null){
        MiEstado.miEstado.setUser (User);
    }
    if(GUILayout.Button("Jugar!")){
        Application.LoadLevel("island_scene");
    }
    if(isUser){
        //here the info about points and pending letters
        if (once) {
            once = false;
            this.askDiscoveredLetters (User);
        }
        GUILayout.Label ("Letras - Puntos");
        var scrollPosition : Vector2;
        if (Screen.width > Screen.height) {
            scrollPosition = GUILayout.BeginScrollView
(scrollPosition, GUILayout.Width (300), GUILayout.Height (600));
        }else{scrollPosition = GUILayout.BeginScrollView
(scrollPosition, GUILayout.Width (490), GUILayout.Height (600));}
        GUILayout.Label (Temp);
        GUILayout.EndScrollView ();
    }
}
GUILayout.EndArea ();
}
}

/*Funcion per efectuar el login*/
function PerformLogin(user : String,pass : String){
    var form : WWWForm = new WWWForm();
    form.AddField("user",user);
    form.AddField("pass",pass);
    var Enviar : WWW = new WWW(Host+PHPUser,form);
    yield Enviar;
    if(Enviar.text == "wrong-user-or-pass"){
        Msg = "Usuario o contraseña incorrectos";
    }else{
        Msg = "Login correcto";
        Texto = "Hola! " + Enviar.text;
        isUser = true;
    }
}
}

```

```

function PerformNewUser(user: String,pass : String,fullname : String, age:
String){
    var form : WWWForm = new WWWForm();
    form.AddField("user",user);
    form.AddField("pass",pass);
    form.AddField("fullname",fullname);
    form.AddField("age",age);
    var Enviar : WWW = new WWW(Host+PHPRegister,form);
    yield Enviar;
    if(Enviar.text == "user-already-exist"){
        Msg = "The user already exist.";
    }else if (Enviar.text == "fatal-error"){
        Msg = "Error 404. Intentalo mas tarde";
    }else{
        Msg = "Registrado correctamente";
        Texto = Enviar.text;
        isNewUser = false;
    }
}

/*
* Function to retrieve the already discovered letters for the registered user
*/
function askDiscoveredLetters (user: String) {
    var form : WWWForm = new WWWForm();
    form.AddField("user",user);
    var Enviar : WWW = new WWW(Host+PHPAskLetters,form);
    yield Enviar;
    //Debug.Log ("Enviar.text: " + Enviar.text);
    if(Enviar.text == "0"){
        Temp = ""; //do nothing
    }else if (Enviar.text == "fatal-error") {
        Temp = ""; //do nothing
    }else{
        Temp = Enviar.text;
        Temp = Temp.Replace('#','\n').ToUpper();
    }
}

/*Function to save the new discovered letter for the registered user*/
public function saveDiscoveredLetter (user: String, letter : String) {
    var form : WWWForm = new WWWForm();
    form.AddField("user",user);
    form.AddField("letra",letter);
    var Enviar : WWW = new WWW(Host+PHPSaveLetters,form);
    yield Enviar;

    if(Enviar.text == "ok"){
        Texto = Enviar.text;
    }else{
        Msg = "Error 404. Intentalo mas tarde";
    }
}
}

```

La part més important d'aquest script és potser el fet d'enviar la request **WWWForm** amb el yield. Yield el que fa es enviar aquesta petició amb un altre thread d'aplicació. D'aquesta manera el procés principal continua executant la funció OnGUI().

La funció OnGUI() funciona com la funció Update. Un cop per frame es processa tot actualitzant el contingut.

Quan rebem la resposta del servidor es pintada just a la següent execució del mètode OnGUI().

Altre punt a destacar d'aquesta escena es que els terminals mòbils tenen diferent mida de pantalla y sobre tot diferent resolució. Això implica que tinguin també diferent relació d'aspecte entre uns terminals i uns altres.

Això fa molt difícil dissenyar la mida dels elements del formulari. Per aquest motiu utilitzo les següents línies al començament del script:

```
GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new  
Vector3(Screen.width / 600.0f, Screen.height / 800.0f, 1));
```

D'aquesta manera el que estic fent es que la pantalla del terminal funcioni com si tingues una resolució de 600x800. Independentment de la resolució nativa del terminal. Això simplifica molt la feina alhora de dissenyar els elements tot i que encara es poden deformar si la relació d'aspecte d'un terminal es molt diferent.

Amb les línies

```
if (Screen.width > Screen.height) {
```

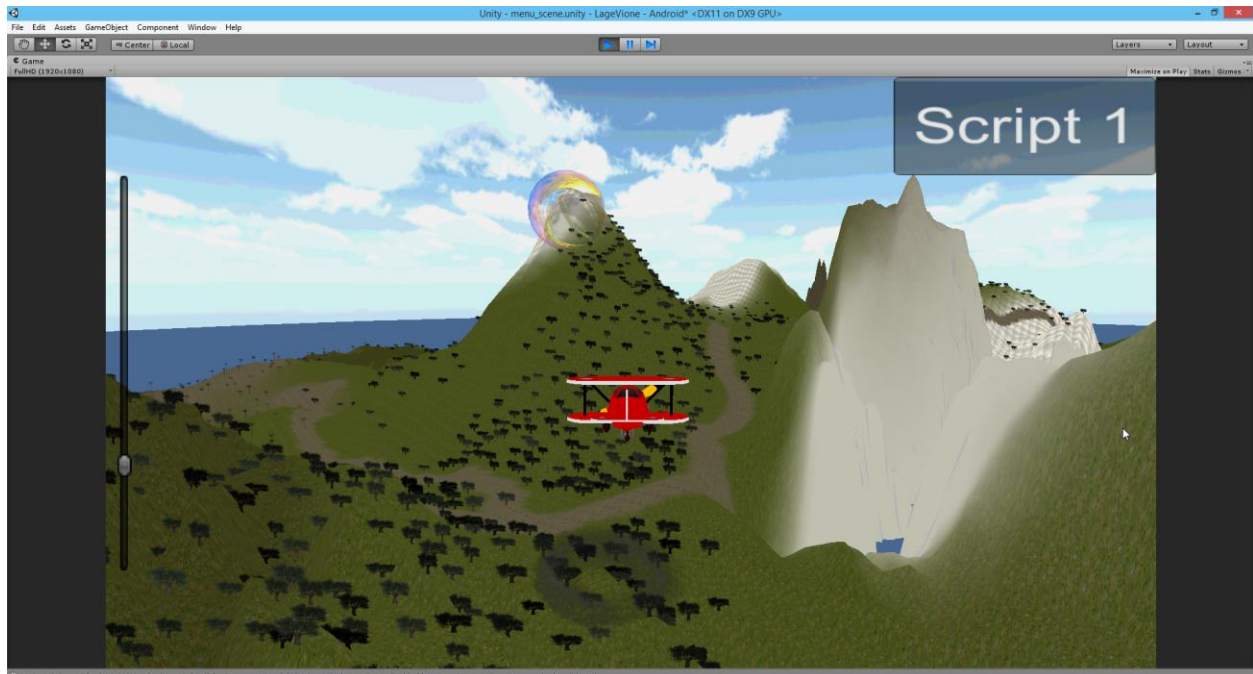
El que estic mirant es si el mòbil està en posició vertical o horitzontal. És molt més eficient que el mètode natiu que ho retorna

```
if ((Input.deviceOrientation == DeviceOrientation.LandscapeLeft) ||  
(Input.deviceOrientation == DeviceOrientation.LandscapeLeft))
```

I el mateix per la posició vertical anomenada Portrait.

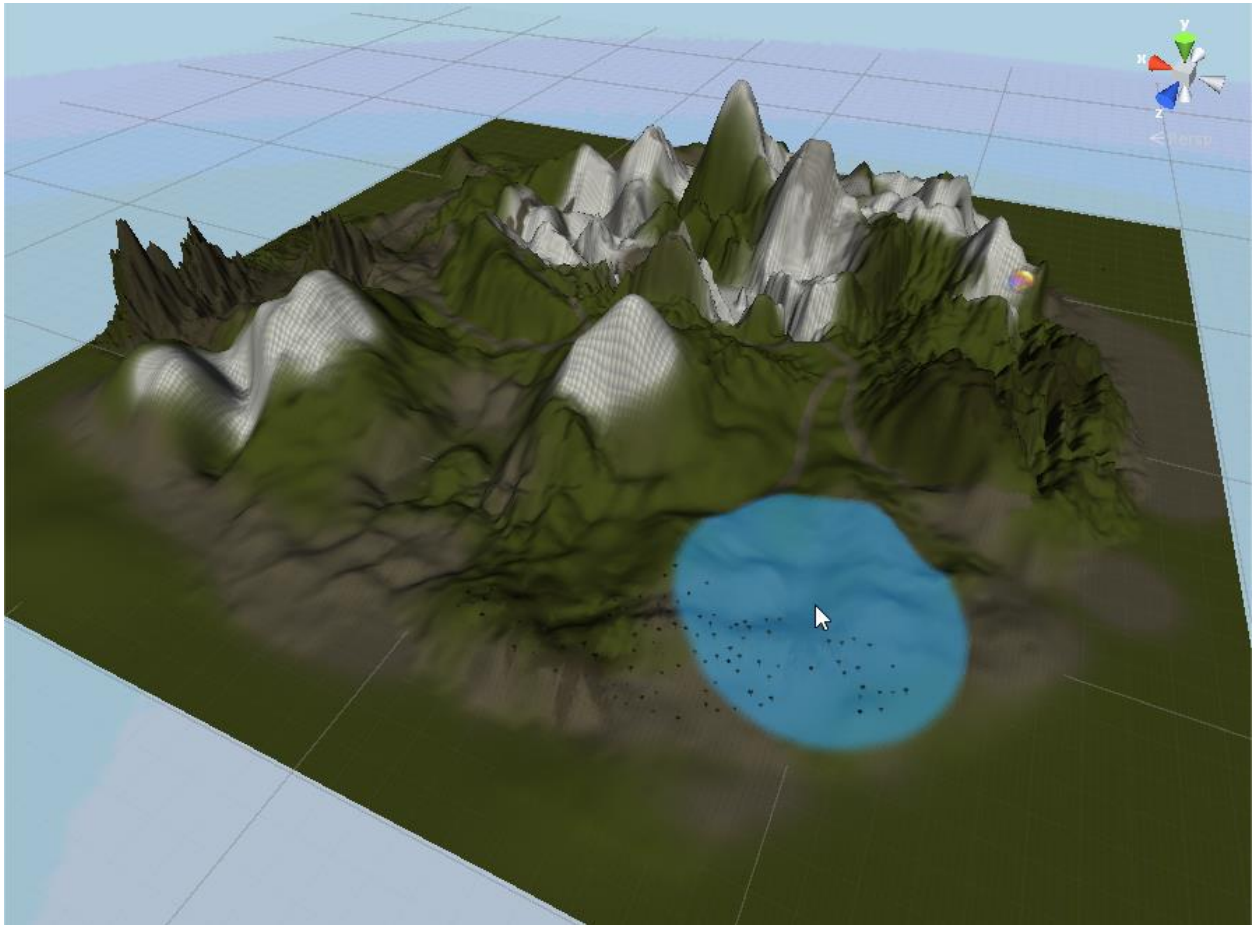
island_scene

island_scene: és l'escena que conté el mapa, l'avió i la lògica del joc que permetrà cercar i encertar les lletres.

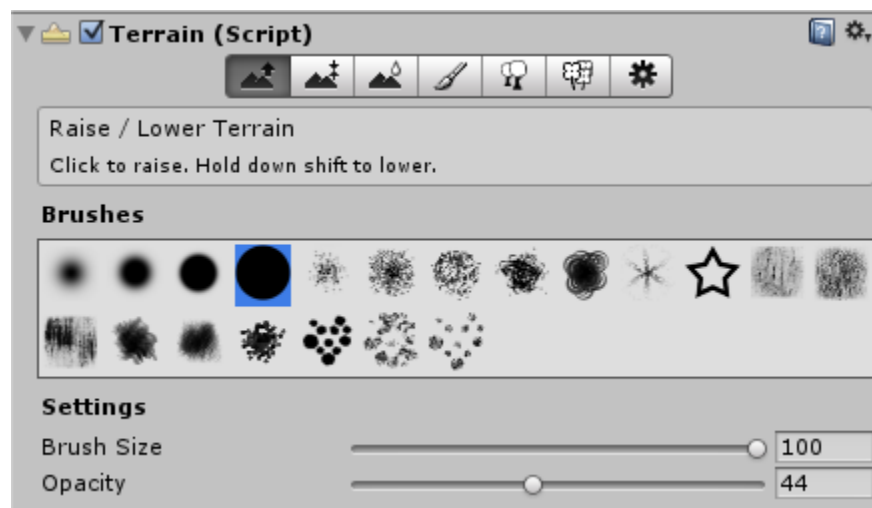


El terreny

El terreny es crea literalment pintant sobre una matriu x,y,z.

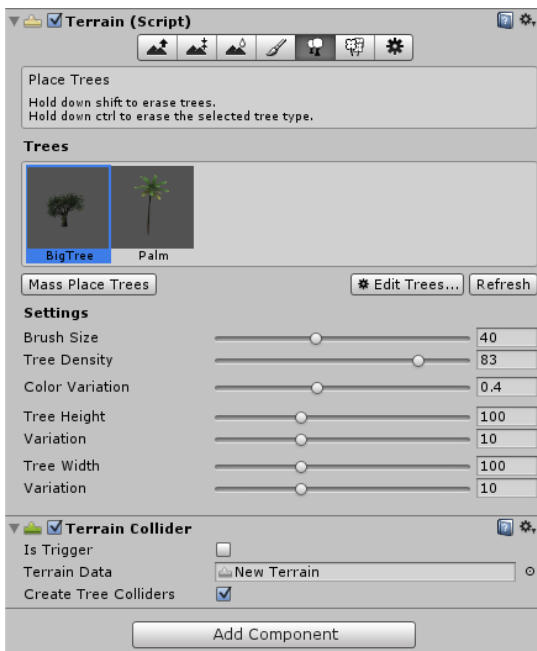
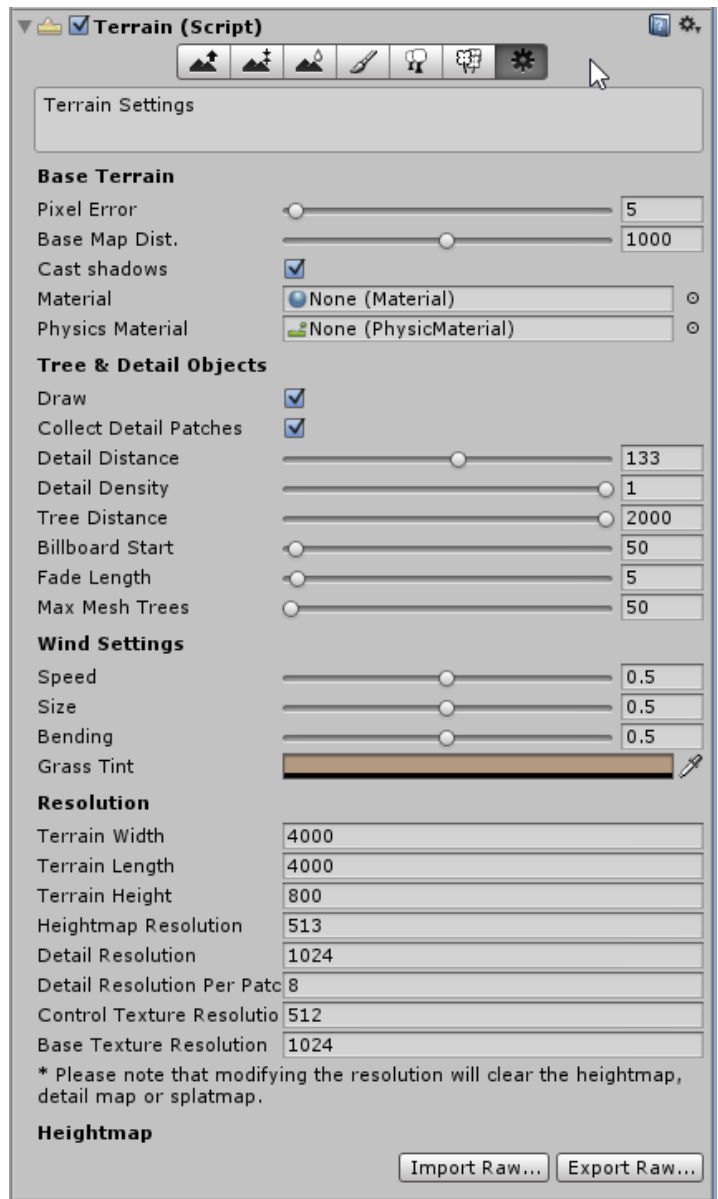


Disposen d'unes eines per fer créixer les muntanyes, amb diferents formes, aplicar textures, arbres flors etc.



El terreny te una sèrie de paràmetres dels que podem destacar la mida o la distancia del detall.

Però també la velocitat del vent, la resolució de la textura etc.



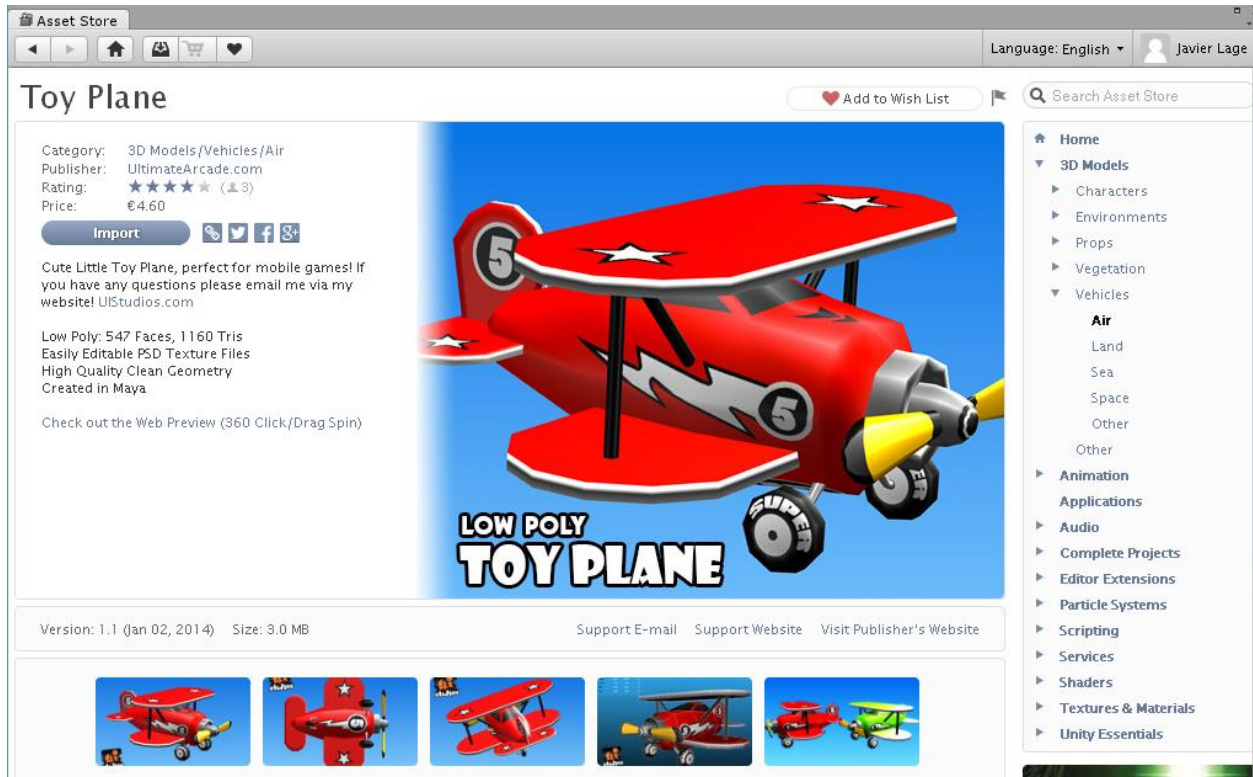
Si volem afegir arbres al terreny els hem d'importar prèviament a Unity.

Un cop importats els podem pintar mes grans o petits, amb mes densitat o menys.

L'avió

Tot i que vaig intentar dissenyar-lo amb Blender 3D Studio vaig decidir ràpidament que no era una opció. No es objectiu el projecte aprendre modelatge 3D. Els rol del desenvolupador de videojocs està clarament diferenciat del rol de dissenyador gràfic.

Buscant alternatives vaig trobar la botiga virtual d'Assets o Asset Store de Unity.



Per menys de 5€ vaig poder comprar-lo i importar-lo directament a Unity.

El problema consistia en dotar-lo de moviment. Per animar l'avió pretenia investigar l'ús del Giròscop del terminal però va ser molt complicat i vaig acabar optant per l'ús d'un Joystick.



Aquesta taca que veiem en pantalla fa la funció de Joystic. Al joc trobem un **GameObject** de tipus Joystic amb un script associat, joystick.cs fet amb C# que detecta les accions de l'usuari. Es a dir, cap a quina direcció estem accionant el Joystic.

Script 2

Aquesta informació es transmet a l'avió. L'avió te associat un altre escript també amb C# on termino com vull que reaccioni.

```
using UnityEngine;
using System.Collections;

public class ControlJoystick : MonoBehaviour {
    //public float speed = 50.0f;
    public Joystick joy;

    // Update is called once per frame
    void Update () {
        float Vertical = -joy.position.y;
        float Horizontal = joy.position.x;

        Vector3 moveCamTo = transform.position - transform.forward * 30.0f +
Vector3.up * 10.0f;
        float bias = 0.96f;
        Camera.main.transform.position = Camera.main.transform.position *
bias + moveCamTo * (1.0f - bias);
        Camera.main.transform.LookAt (transform.position + transform.forward
* 50.0f);

        transform.position += transform.forward * Time.deltaTime *
GameObject.FindGameObjectWithTag ("Plane").GetComponent<SpeedvSlider>().speed;

        transform.Rotate (Vertical, 0.0f, -Horizontal);
        float AlturaPies = Terrain.activeTerrain.SampleHeight
(transform.position);

        if (AlturaPies > transform.position.y) {
            transform.position = new Vector3 (transform.position.x,
AlturaPies, transform.position.z);
        }
    }
}
```

Aquest script te diferents propòsits. Un dels propòsits es donar velocitat a l'avió. La velocitat que aplica a l'avió la aconsegueix de la barra vertical Slider situada a la esquerra de la pantalla. Es un altre script afegit com a component a l'avió.

```
transform.position ... ( "Plane").GetComponent<SpeedvSlider>().speed;
```

El script de joystick.cs afegit també a l'avió recull els Inputs de l'usuari i ho transforma a eix vertical y horitzontal amb aquestes línies de codi faig girar l'avió.

```
transform.Rotate (Input.GetAxis ("Vertical"), 0.0f, -
Input.GetAxis ("Horizontal")*4);
```


Una altra de les funcions d'aquest script és la de forçar a la càmera a perseguir l'avió. Ho aconseguixo amb les línies.

```
Camera.main.transform.position ...etc
```

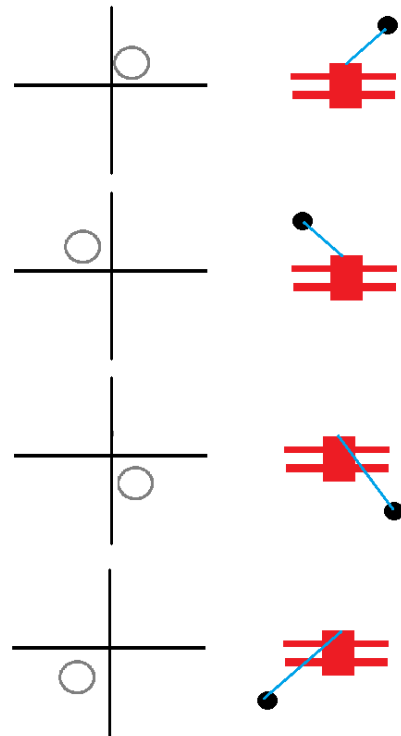
Script 1

Aquesta es la primera versió que vaig fer en quant al moviment de l'avió però no em quedava satisfet i volia investigar més.

Seguint el concepte de que la càmera va seguint l'avió el que vaig fer es crear un objecte invisible anomenat 'Cubo' sense textura ni component per renderitzar-lo (per tant, no visible), i transmetre el moviment del joystick al Cubo.

Un cop podia moure el cub per la pantalla forçava que l'avió el seguis.

El resultat va ser força agradable per tant vaig crear un botó per poder canviar d'un script a l'altre.



Velocitat variable

La velocitat es pot regular amb la barra slider de la esquerra de la pantalla. Es un GameObject de tipus GUI. El script que permet fer això esta afegit.

Té una variable pública de tipus float per preestablir la velocitat.

La velocitat de l'avió pot anar de 0 a 200.

La variable es pública perquè la utilitzo amb la resta de Scripts de moviment i col·lisió amb l'esfera.

```
using UnityEngine;
using System.Collections;

public class SpeedvSlider : MonoBehaviour {
    public float speed = 50.0f;
    public float lastSpeed;
    private int SliderLong = 200;
    private bool isShown = true;

    /*Draw slider in GUI*/
    void OnGUI () {
        if (isShown) {
            GUI.matrix = Matrix4x4.TRS (Vector3.zero, Quaternion.identity,
new Vector3 (Screen.width / 900.0f, Screen.height / 300.0f, 1));
            if (Screen.width > Screen.height) {
                speed = GUI.VerticalSlider (new Rect (10, 50, 50,
SliderLong), speed, 200.0f, 0.0f);
            } else {
                GUI.matrix = Matrix4x4.TRS (Vector3.zero,
Quaternion.identity, new Vector3 (Screen.width / 300.0f, Screen.height /
300.0f, 1));
                //speed = GUI.VerticalSlider (new Rect (50, 100, 50,
SliderLong/2), speed, 200.0f, 0.0f);
            }
            speed = GUI.VerticalSlider (new Rect (10, 50, 50, SliderLong),
speed, 200.0f, 0.0f);
        }
    }

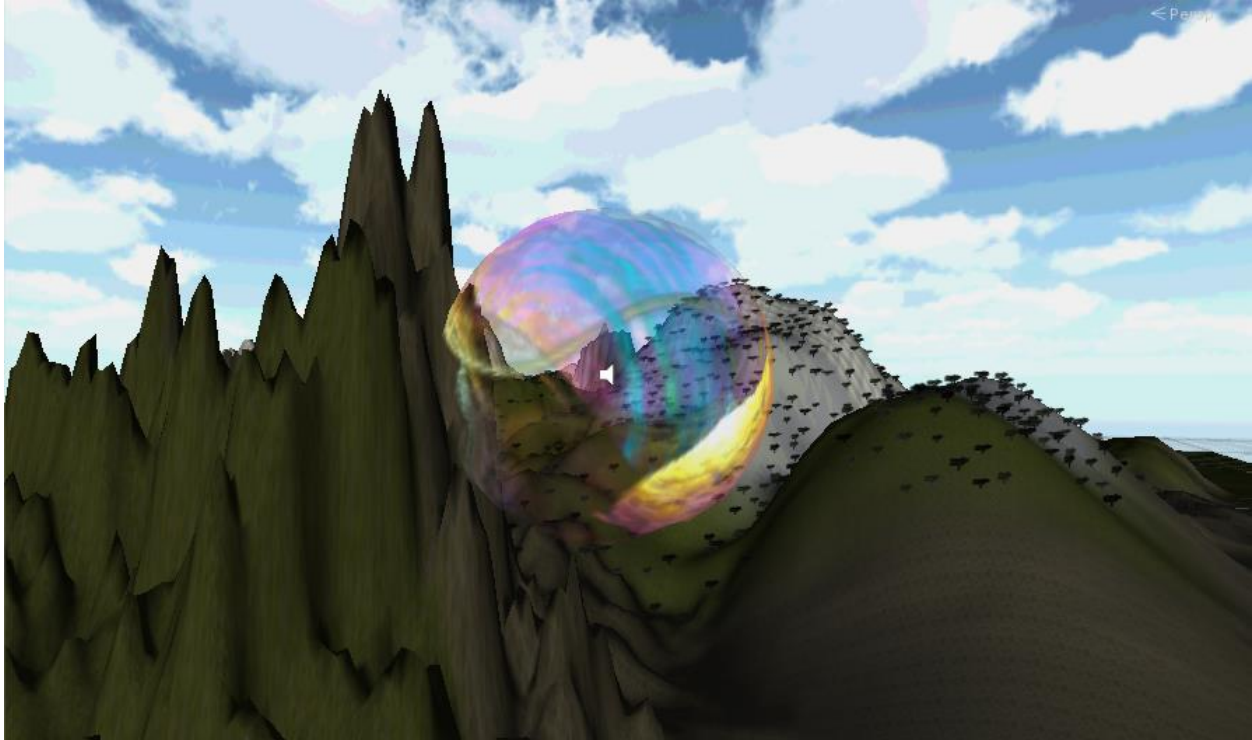
    /*Stop de movement of the plane (for all scripts)*/
    public void StopMovement () {
        lastSpeed = speed;
        speed = 0.0f;
        isShown = false;
    }

    /*Continue Movement using lastSpeed var.*/
    public void ContinueMovement () {
        speed = lastSpeed;
        isShown = true;
    }
}
```

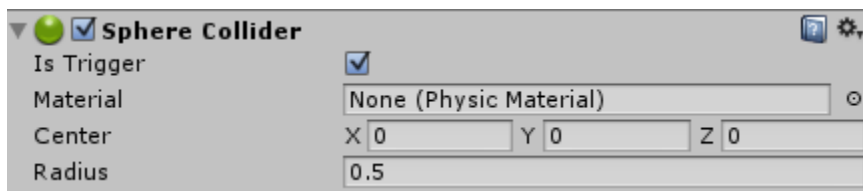
La esfera

La esfera es va col·locant aleatòriament sobre el mapa. Amb l'avió l'hem de trobar i xocar-nos.

La esfera es un GameObject de tipus esfera amb una textura de bombolla de sabó. Te un script per rotar sobre els eixos X i Y.

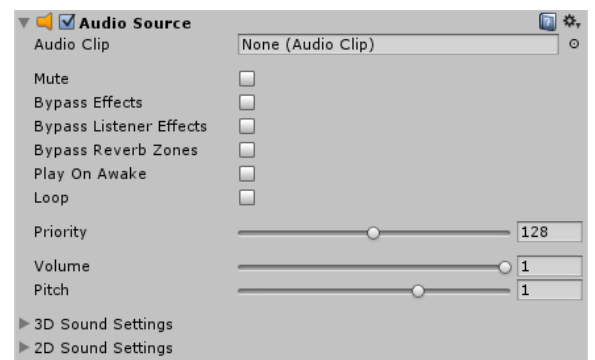


La esfera te activat el component de Collider que ens permetrà des de l'escript programar accions quan l'avió es col·lisió amb ella.



La esfera també te un component associat per a la gestió dels àudios que reproduïrem.

No te cap audio associat. Això o gestionarem de forma dinàmica amb un altre script.




```

using UnityEngine;
using System.Collections;
public class SphereCollision : MonoBehaviour {
    public bool Mostrar;
    public string laLetra = "A";
    public string texto;
    public string button1;
    public string button2;
    public string button3;
    private Texture speakerPicture; //imagen del sonido
    private GUISkin MySkin;
    private AudioClip sonido;
    private AudioSource myAudioSource;
    private bool acertado = false; //flag para cuando acierte la letra
    private bool controlCubo = false;
    // Use this for initialization
    void Start () {
        speakerPicture = Resources.Load<Texture>("iconosonido");//cargamos la imagen
de la carpeta resources
        myAudioSource = this.GetComponent<AudioSource>();
        myAudioSource.pitch = 1;
        MySkin = Resources.Load<GUISkin>("SkinEsferas");
        this.MoveSphereRandomly (); //mueve la esfera a una posicion random del mapa
        this.laLetra = this.nuevaLetraRandom().ToString().ToUpper(); //assignar letra
        this.reassignButtons (); //asigna valores a los botones
    }

    // Update is called once per frame
    void Update () {
        //no need
    }
    void OnTriggerEnter(Collider other)
    {
        AsignarSonido();
        myAudioSource.Play();
        GameObject.FindGameObjectWithTag ("Plane").GetComponent<SpeedvSlider>
().StopMovement ();
        Mostrar = true;
        this.ChangeAutoRotateStatusTo(false);
        GameObject.FindGameObjectWithTag
("CambiarControles").GetComponent<CambiarControles> ().mostrarBotonCambiarScript =
false;
    }
    void OnGUI(){
        float fcx; //factor corrector de eje X de la relacion de aspecto de la
pantalla
        float fcy; //factor corrector de eje Y de la relacion de aspecto de la
pantalla
        int offsetY; //desplazamiento de y;
        GUI.skin = MySkin;

        if(Mostrar == true) {
            //if screen is landscape
            if (Screen.width > Screen.height) {
                fcx = 1.0f;
                fcy = 1.0f;
                offsetY = 0;
                GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity,new
Vector3(Screen.width / 800.0f, Screen.height / 600.0f, 1));
            }
            //if screen is portrait define instead:
            else {
                fcx = 1.5f;

```

```

        fcy = 1.5f;
        offsetY = 100;
        //Debug.Log ("else fcx=" + fcx);
        GUI.matrix = Matrix4x4.TRS(Vector3.zero, Quaternion.identity, new
Vector3(Screen.width / 800.0f, Screen.height / 800.0f, 1));
    }
    //if screen is landscape
    //Debug.Log ("Screen is Landscape {Width:" + Screen.width + " Height:"
+ Screen.height + "}");
    //interfaz de las letras
    GUI.Box(new Rect(200/fcx,50+offsetY,400*fcx,410),"");
    //boton reproducir, el audio source debe tener quitado la opcion loop
para que solo suene una vez
    if(GUI.Button(new Rect(210/fcx,60+offsetY,380*fcx,80/fcy),speakerPicture))
{
    myAudioSource.Play();
    //Debug.Log ("Volume on play: " +
myAudioSource.volume.ToString());
}
    //label ¿que letra es?
    GUI.Label(new Rect(280/fcx,140+offsetY,350*fcx+offsetY,100/fcy),texto);
    //Botones, con el if se recoge la pulsacion en el boton
    if(GUI.Button(new Rect(350,200+offsetY,100*fcx,80/fcy),button1)) {
        if(button1 == laLetra) {
            acertado = true;
        } else {
            texto = button1 + ": NOOOO!!!";
        }
    }
    if(GUI.Button(new Rect(350,(200+80+5)+offsetY,100*fcx,80/fcy),button2)) {
        if(button2 == laLetra) {
            acertado = true;
        } else{
            texto = button2 + ": NOOOO!!!";
        }
    }
    if(GUI.Button(new
Rect(350,(200+80+80+10)+offsetY,100*fcx,80/fcy),button3)) {
        if(button3 == laLetra) {
            acertado = true;
        } else{
            texto = button3 + ": NOOOO!!!";
        }
    }
}
if(acertado == true)
{
    Mostrar = false; acertado = false;
    if(MiEstado.miEstado != null){ //that line represents an user that
have been properly logged into our game.

        //Debug.Log ("MiEstado.miEstado != null: " +
MiEstado.miEstado.getUser());
        StartCoroutine(
this.saveDiscoveredLetter(MiEstado.miEstado.getUser(),laLetra.ToLower() ));
    }
    this.MoveSphereRandomly();

GameObject.FindGameObjectWithTag("Plane").GetComponent<SpeedvSlider>().ContinueMovemen
t();

    laLetra = this.nuevaLetraRandom().ToString().ToUpper();
    this.reassignButtons (); //asigna valores a los botones
    texto = "¿Que letra es?";

```

```

        this.ChangeAutoRotateStatusTo (true);
        GameObject.FindGameObjectWithTag
("CambiarControles").GetComponent<CambiarControles> ().mostrarBotonCambiarScript =
true;

    }
}

/*
 * Asignacion del sonido a reproducir
 */
void AsignarSonido(){
    sonido = Resources.Load<AudioClip>("Letras/"+laLetra);
    myAudioSource.volume = 1.0f;
    myAudioSource.minDistance = 500.0f;
    myAudioSource.pitch = 1;
    myAudioSource.clip = sonido;
}

/*
 * mueve la esfera a una nueva posicion
 */
void MoveSphereRandomly(){
    float x = Random.Range (50, 4000);
    float y =
Terrain.activeTerrain.SampleHeight(transform.position)+Random.Range(100,800);
    float z = Random.Range(50,4000);
    transform.position = new Vector3(x,y,z);
}

/*
 * Generar letra random
 */
private char nuevaLetraRandom() {
    // This method returns a random UpperCase letter.
    int num = Random.Range(0, 26); // Zero to 25
    char let = (char)('a' + num);
    return let;
}

/*
 * Asigna letras random a los botones
 */
private void reassignButtons(){
    button1 = nuevaLetraRandom().ToString().ToUpper();
    button2 = nuevaLetraRandom().ToString().ToUpper();
    button3 = nuevaLetraRandom().ToString().ToUpper();

    if ((button1 + button2 + button3).Contains (laLetra)) {
        //do nothing
    } else {
        int i = Random.Range(1, 4);
        if (i==1) {
            button1 = laLetra;
        } else if (i==2) {
            button2 = laLetra;
        }else{
            button3 = laLetra;
        }
    }
}
}
/*

```

```

    * Lock screen autorotation functionality
    */
    private void ChangeAutoRotateStatusTo(bool status) {
        Screen.autorotateToLandscapeLeft = status;
        Screen.autorotateToPortraitUpsideDown = status;
        Screen.autorotateToLandscapeLeft = status;
        Screen.autorotateToLandscapeRight = status;
    }
}

```

D'aquest script cal destacar diferents funcionalitats importants.

```

void Start () {
    speakerPicture = Resources.Load<Texture>("iconosonido");//cargamos la imagen
de la carpeta resources
    myAudioSource = this.GetComponent<AudioSource>();
    myAudioSource.pitch = 1;
    MySkin = Resources.Load<GUISkin>("SkinEsferas");
    this.MoveSphereRandomly (); //mueve la esfera a una posicion random del mapa
    this.laLetra = this.nuevaLetraRandom().ToString().ToUpper(); //assignar letra
    this.reassignButtons (); //asigna valores a los botones
}

```

La funció Start en aquest cas es molt important. Carrega el component d'àudio, assigna una nova lletra random a la esfera i col·loca la esfera en una nova posició aleatòria de la pantalla. També carrega el CSS MySkin per a ser emprat al formulari d'encertar la lletra.

```

void OnTriggerEnter(Collider other)

```

Aquesta funció es un Trigger. Salta quan la esfera ha detectat que l'avió a col·lisionat amb ella i a partir d'aquí es fan moltes accions.

Es canvia el flag 'mostrar' a true per a que la funcion onGUI comenci a mostrar el formulari per encertar la lletra.

Es reproduueix el so.

El joc continua quan s'encerta la lletra.

```

onGUI ()

```

Mostra el formulari per encertar la lletra.

```
private void reassignButtons () {
```

Un cop la lletra que hem de trobar i encertar aquest torn és assignada a la esfera, aquesta funció te la finalitat de donar valor als botons del formulari. Un dels botons ha de contenir la lletra que hem d'encertar i els altres una lletra aleatòriament cadascun.

```
void AsignarSonido () {
```

Un cop la lletra que hem de trobar i encertar aquest torn és assignada a la esfera anem a buscar el so de LaLletra.mp3 a la carpeta resources del projecte per carregar-lo de forma dinàmica. Això va ser un bon problema perquè si no es coneix el funcionament i com Unity treballa amb la estructura de carpetes no es poden carregar imatges o sons dinàmicament.

```
void MoveSphereRandomly ()
```

Un cop encertada la lletra la esfera es col·loca en una nova posició aleatòria del mapa, es torna a assignar una altre lletra i comença a de nou a jugar.

```
GameObject.FindGameObjectWithTag ("Plane").GetComponent<SpeedvSlider> ().StopMovement  
();
```

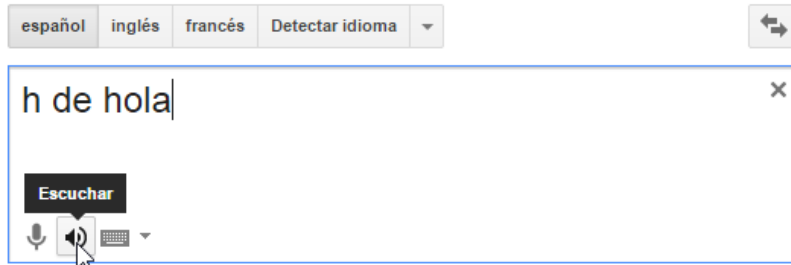
Amb aquesta funció paro el moviment de l'avió i oculto els botons de canviar el script de moviment de l'avió i de regular la velocitat de vol.

És un script bastant llarg, m'agradaria haver-lo modularitzat mes distribuint les funcions en altres fitxers.

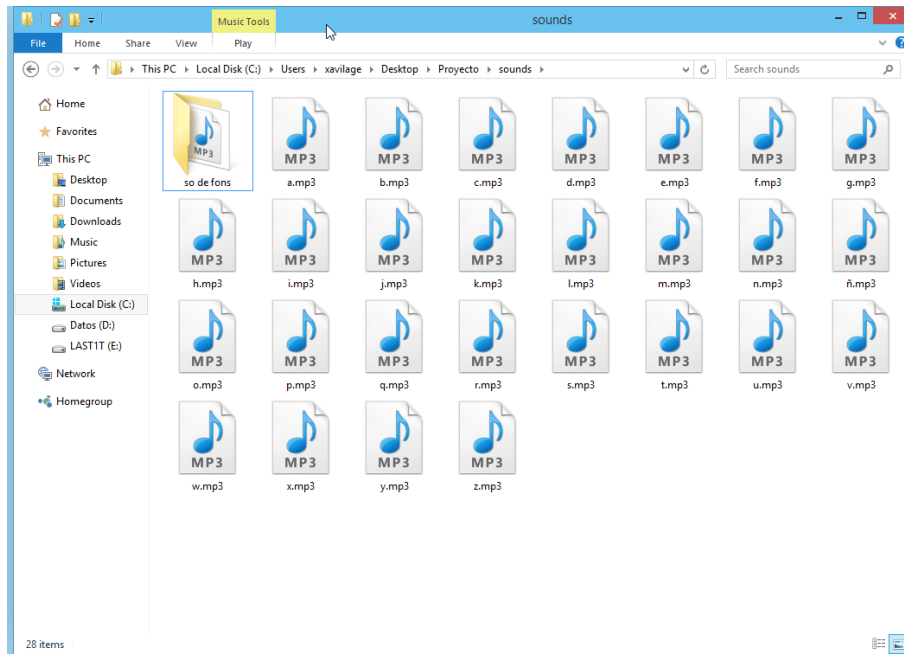
Els sons de les lletres

Els sons els he generat amb google translàtor. El so que es genera al clicar l'altaveu de google translàtor es poden descarregar automàticament mitjançant aquesta url.

href=http://translate.google.com/translate_tts?tl=es&q=h%20de%20hola



He fet un so per cada lletra i els he agut de ubicar a la carpeta \Assets\Resources del projecte. És l'única carpeta que Unity llegeix dinàmicament, en temps d'execució.



S'assignen al component d'àudio de la esfera. Quan l'avió topa amb ella es reproduïxen. Estan gestionats per l'escript SphereCollision.cs també en C#.

5. Sobre el servidor

Objectiu

El joc incorpora la possibilitat de registrar-se com a usuaris i validar-se abans de començar a jugar. Això permet a l'usuari accedir a un registre de les lletres que hem trobat i la puntuació obtinguda.

Sense dubte es una part que m'agradaria desenvolupar molt més però a falta de temps, fer que funcioni de forma bàsica m'ha resultat força interessant.

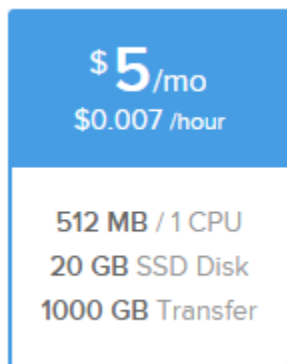
En un futur, realitzant alguns canvis al codi del joc, la connectivitat amb un servidor online em permetria moltes coses. Per exemple afegir nou contingut al joc simplement actualitzant una base de dades o crear rankings d'usuaris.

L'enfocament que li dono per a aquest projecte es que els pares puguin consultar quines lletres a trobat el seu fill i així poder-li preguntar si les recorda.

Com és el servidor virtual

He contractat un servidor virtual a www.digitalocean.com. Es tracta d'un servidor dedicat amb Ubuntu 14.04 de 64 bits on puc instal·lar els serveis que necessiti.

El cost és de 5\$ al més i dispeno de 512MB de memòria RAM, 20GB de disc dur SSD i 1TB de transferència de dades mensuals. Realment no necessito tot això però es la versió més econòmica que ofereixen.



\$5 /mo
\$0.007 /hour


512 MB / 1 CPU
20 GB SSD Disk
1000 GB Transfer

Public Network

IP Address: 146.185.148.96

Gateway: 146.185.148.1

Netmask: 255.255.255.0

Image	Name	IP Address	Status	Memory	Disk	Region
	lagetest	146.185.148.96	Active	512MB	20GB	ams1

SSH: Putty i WinSCP

Digitalocean anomena Droplets als seus servidors virtuals. En crear un droplet puc connectar-me des de la seva web mitjançant un servei VNC i un plug-in de browser que donen preinstal·lat.



No era gaire còmode ja que hi havia coses que no podia fer bé com copiar i enganxar text o accedir visualment als fitxers.

Per això vaig instal·lar i configurar OpenSSH server amb `#sudo apt-get install openssh-server`. Ara ja podia fer servir el software putty com a client ssh y el winscp per accedir als fitxers del servidor de forma molt més còmode.



Putty.exe



WinSCP.exe

```
root@lagetest: ~
login as: root
root@146.185.148.96's password:
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Sat Jan 10 12:24:29 EST 2015

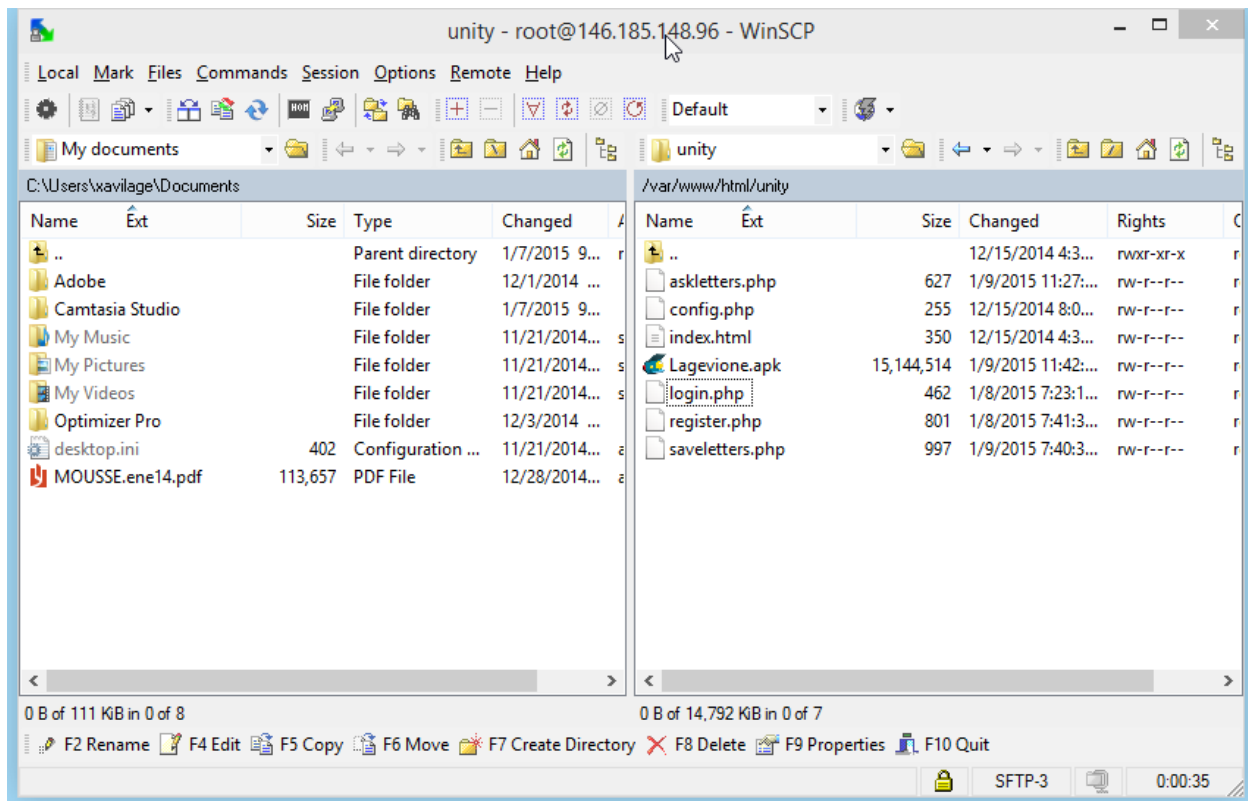
System load:  0.02                Processes:    80
Usage of /:   12.9% of 19.56GB     Users logged in:  0
Memory usage: 34%                IP address for eth0: 146.185.148.96
Swap usage:   0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

90 packages can be updated.
54 updates are security updates.

-----
Thank you for using DigitalOcean's LAMP Application.
Your web root is located at /var/www/html and can be seen from http://146.185.148.96/
The details of your PHP installation can be seen at http://146.185.148.96/info.php
Your MySQL root user's password is h5dUigA7nW
You are encouraged to run mysql_secure_installation to ready your server for production.

-----
To delete this message of the day: rm -rf /etc/motd.tail
Last login: Sat Jan 10 12:24:29 2015
root@lagetest:~#
```

Apache: Web server

El següent va ser instal·lar Apache i el mòdul de PHP.

Apache es un servidor Web HTTP de codi obert. El seu desenvolupament va començar al 1995. És el software emprat al 70% dels servidors web. Les alternatives son varies com nginx, o ISS de Microsoft.



Comprendré Apache per modificar-lo o crear noves funcionalitats és molt complex però instal·lar-lo i i emprar-lo per al propòsit del joc no havia de ser tan complicat.

La idea era crear un servei Web que cridarem des de el joc per a satisfer diferents funcionalitats.

- Registrar usuaris nous del joc.
- Validar usuaris existents.
- Guardar les lletres que un usuari validat ha trobat durant la partida.
- Consultar totes les lletres que ha trobat anteriorment un usuari.

MySQL Base de dades

Des de 2009 Oracle desenvolupa i distribueix MySQL com a sistema de gestió de base de dades relacional com a software lliure.

MySQL es utilitza per grans marques com Wikipedia, Facebook, etc. Cal destacar que aquestes grans marques no l'utilitzen per a cerques.

Te un sistema de llicenciament complex que permet mantenir copyright de la major part del codi i cobrar llicència si es pretén emprar amb productes privats. Tot i així es distribueix sota llicència GNU GPL.

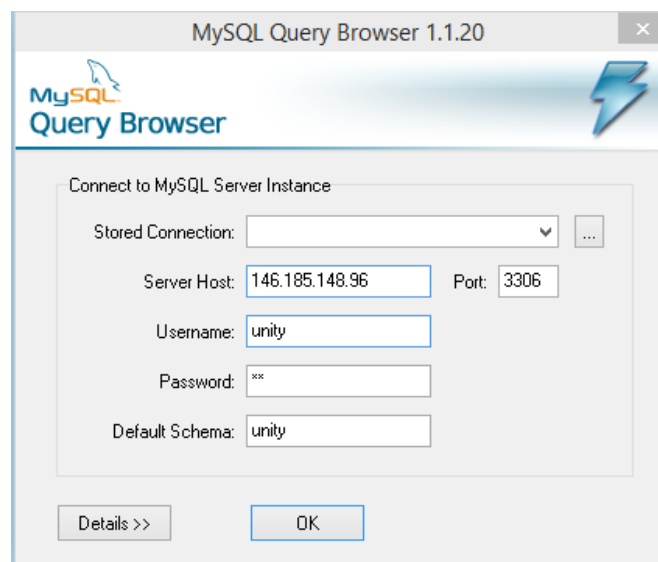
És multi usuari, multi plataforma i també l'utilitzen molts gestors de continguts com Drupal, Joomla, Wordpress, etc.

La instal·lació de la versió 5.6 de MySQL al servidor virtual Ubuntu 14 ha sigut senzilla amb la comanda `#apt-get install mysql-server-5.6.` i seguint els passos.

Vaig crear un schema amb linia de comandes i un usuari amb els respectius permisos.

- Usuari: **unity**
- Contrasenya: **3d**

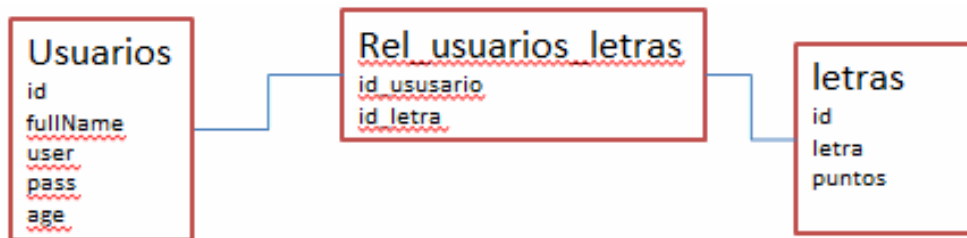
Com a client de MySQL he fet servir MySQL Query Browser. Un software també distribuït sota llicència GNU.



Disseny de la base de dades

La base de dades s'ha fet molt simple. Consisteix només en una taula per guardar usuaris. Una altra taula per a guardar lletres, i finalment una taula per a guardar la relació entre usuaris i lletres, es a dir, les lletres que ha trobat cada usuari durant el joc.

No és la finalitat del projecte crear un sistema complex de base de dades sino fer un video joc. Tot i així he volgut donar sentit a les bases apreses durant la enginyeria i encaixar-les de la manera correcta amb el vídeo joc.



La taula **usuarios** conté els usuaris que s'han enregistrat al joc.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id	INT(10)	✓	✓	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
fullname	VARCHAR(25)	✓		<input type="checkbox"/> BINARY	NULL
user	VARCHAR(15)	✓		<input type="checkbox"/> BINARY	NULL
pass	VARCHAR(20)	✓		<input type="checkbox"/> BINARY	NULL
age	INTEGER	✓		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
enable	TINYINT(1)	✓		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	1

Query d'exemple:

```
INSERT INTO usuarios (user,pass,fullname,age) values ('$User','$Pass','$Fullname','$Age');
```

La taula **letras** conté les lletres que seran mostrades pel joc.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id	INTEGER	✓	✓	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
letra	CHAR(1)	✓		<input type="checkbox"/> BINARY <input type="checkbox"/> ASCII <input type="checkbox"/> UNIC	
puntos	INTEGER	✓		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	1

Query d'exemple.

```
SELECT id FROM letras l where l.letra ='$Letra';
```

La taula **rel_usuarios_letras** conté la relació entre usuaris i lletres. Guardarem un registre cada cop que un usuari encerti una lletra durant una partida.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id_usuario	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0
id_letra	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	0

Per fer això primer hem de trobar l'id de l'usuari i l'id de la lletra. Un cop obtinguts els guardem en aquesta taula.

Query d'exemple.

```
insert into rel_usuarios_letras (id_usuario, id_letra) values ('$ id_usuario', '$ id_letra');
```

Els fitxers html i php

Amb el propòsit definit i clar em dispenso a crear la base de dades comentada i els fitxers php que es connectaran a aquesta per consultar o guardar informació.

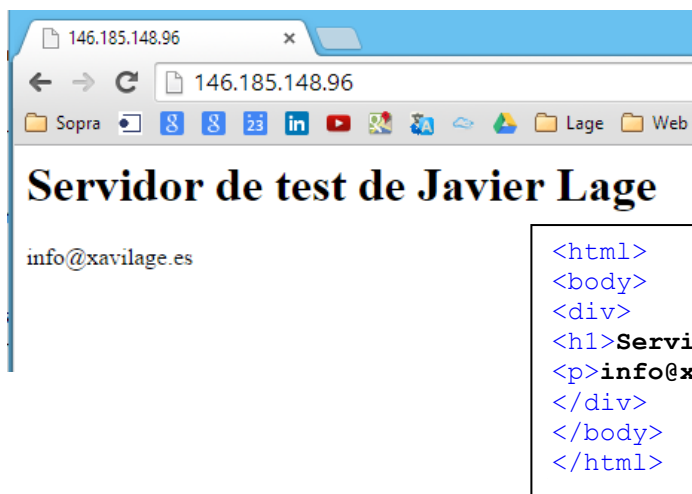
Apache té un directori per defecte on hem de localitzar els fitxers que seran visibles online. Aquest directori es podria canviar i es diferent segons el sistema operatiu.

En anar a <http://146.185.148.96/> en el nostre cas estariem accedint als fitxers del següent directori:

`\var\www\html` → directori arrel d'Apache.

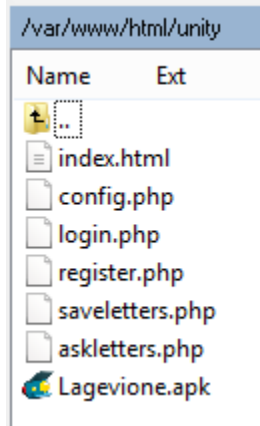
En aquest directori he localitzat només un fitxer `index.html` ja que serà el que es mostrarà automàticament per el nostre servidor Web Apache. Si Apache no trobés aquest fitxer mostraria automàticament un llistat dels fitxers que conté el directori a través del navegador a qualsevol visitant. Això vulneraria una mica la privacitat.

El fitxer `index.html` del directori arrel d'Apache conté només:



El directori que conté els fitxers respectius al joc és:

`\var\www\html\unity` → directori amb els fitxers del projecte.

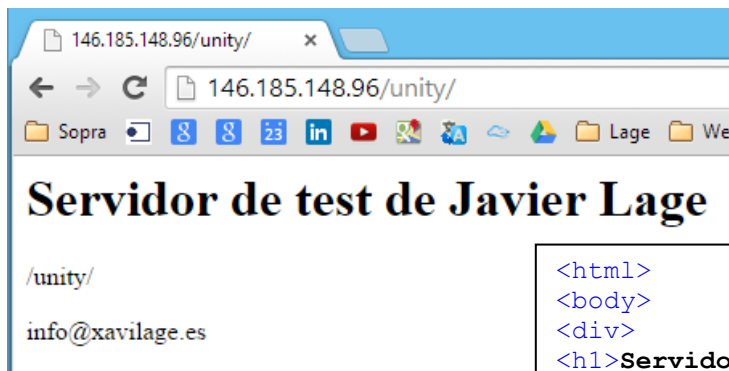


Accedeixo als fitxers directament amb WinSCP connectant-me com a root a la direcció IP estàtica del servidor virtual.

Dintre d'aquesta carpeta `\unity\` he creat també un fitxer `index.php` que serà el retornat automàticament per el servidor web Apache en cridar la url <http://146.185.148.96/unity/>.

Sense aquest fitxer es mostraria un llistat dels fitxers del directori vulnerant la privacitat desitjada.

`\var\www\html\unity\index.php`



```
<html>
<body>
<div>
<h1>Servidor de test de Javier Lage</h1>
<p>/unity/</p>
<p>info@xavillage.es</p>
</div>
</body>
</html>
```

Els fitxers php contenen codi que es visible només per al servidor. Imprimeixen per pantalla mitjançant el navegador el contingut que escrivim rere la comanda **echo**. La resta es contingut privat que ho executarà l'Apache.

Aquest motiu es suficient per quedar-me tranquil en quant a la privacitat de les dades tot hi que no he securitzat el joc contra SQL injection.

Els fitxers php son:

- **config.php**: proveeix la connexió a base de dades MySQL.
- **register.php**: Permet el registre de nous usuaris a la base de dades.
- **login.php**: permet la validació d'usuaris existents a la base de dades.
- **saveletters.php**: permet guardar les lletres que va encertant un jugador durant la partida.
- **askletters.php**: permet consultar les lletres que ha encertat un usuari registrat.

Els fitxers php estan directament relacionats amb la base de dades pel que faré aquí una pausa per continuar explicant-los conjuntament amb MySQL.

config.php

El fitxer config.php te la funció i el codi necessari per establir la connexió contra la base de dades mysql. Serà emprat per la resta de fitxers php emprant la línia de comanda @`include('config.php');`

Aquest fitxer, si la connexió contra la base de dates mysql del host indicat, amb l'user y pasword indicat funciona be, no ens retornarà res.

En cas que la connexió no es pugui realitzar per algun motiu ens retornarà el missatge `("Data base error connection.")`.

```
<?PHP
    $host = "146.185.148.96";
    $User = "unity";
    $Pass = "3d";

    $con = mysql_connect($host,$User,$Pass) or die("Data base error
connection.");
?>
```

register.php

El fitxer register.php te la funció i el codi necessari per permetre el registre d'usuaris nous mitjançant una comanda **insert** a la base de dades. La base de dades especificada al fitxer config.php.

```
<?php
    @include('config.php');

    $User = $_REQUEST['user'];
    $Pass = $_REQUEST['pass'];
    $Fullname = $_REQUEST['fullname'];
    $Age = $_REQUEST['age'];

    $insert = "INSERT INTO usuarios (user,pass,fullname,age) values
('{$User}', '{$Pass}', '{$Fullname}', '{$Age}')";
    $query = "SELECT * FROM usuarios WHERE user='{$User}'";
    $query_response = mysql_query($query) or die("fatal-error");
    $ensure = mysql_num_rows($query_response);
    if($ensure == 0){
        $query_response = mysql_query($insert) or die("wrong-user-or-pass");
        $query2 = "SELECT * FROM usuarios WHERE user='{$User}' AND
pass='{$Pass}'";
        $query_response2 = mysql_query($query2) or die("fatal-error");
        $ensure2 = mysql_num_rows($query_response2);
        if($ensure2 == 1){
            echo "" . $Fullname;
        }
    }else{
        echo "user-already-exist";
    }
?>
```

Cridarem directament a aquest fitxer register.php mitjançant una url. Per funcionar correctament la URL ha de portar correctament especificats els 4 paràmetres.

- 'user': usuari amb el que farem login posteriorment.
- 'pass': contrasenya amb la que farem login posteriorment.
- 'fullname': nom complet
- 'age': edat

Un exemple de la url que hauríem de cridar per donar d'alta un usuari nou a la base de dades mysql és:
<http://146.185.148.96/unity/register.php?user=lage&pass=123456&fullname=LageFullName&age=17>

Possibles sortides a aquesta crida:

- Si l'usuari ja existeix a la base de dades es retornarà el missatge: "user-already-exist".
- Si hi ha un error realitzant alguna de les consultes rebrem el missatge: "fatal-error".
- Si tot funciona correctament retornarem el nom complet especificat a la variable 'fullname'.

login.php

El fitxer login.php te la funció i el codi necessari per permetre la validació d'usuaris al joc. Aquesta validació es fa comprovant un usuari i una contrasenya mitjançant una comanda **select** a la base de dades. La base de dades es la especificada al fitxer config.php.

```
<?php
@include('config.php');

$user = $_REQUEST['user'];
$pass = $_REQUEST['pass'];

$query = "SELECT * FROM usuarios WHERE user='$User' AND pass='$Pass'";
$query_response = mysql_query($query) or die("fatal-error");
$check = mysql_num_rows($query_response);
if($check == 1){
    if($ln = mysql_fetch_array($query_response)){
        $user = $ln['user'];
        $fullname = $ln['fullname'];
        echo "" . $fullname;
    }
} else{
    echo "wrong-user-or-pass";
}
?>
```

Cridarem directament a aquest fitxer login.php mitjançant una url. Per funcionar correctament la URL ha de portar correctament especificats els 2 paràmetres.

- 'user': usuari amb el que iniciarem sessió al joc.
- 'pass': contrasenya amb la que iniciarem sessió al joc.

L'inici de sessió es fa simplement validant l'existència d'aquestes dades a la base de dades mysql.

Un exemple de la url que hauríem de cridar per validar-nos com a usuari existent a la base de dades mysql és:

<http://146.185.148.96/unity/login.php?user=lage&pass=123456>

Possibles sortides a aquesta crida:

- Si l'usuari i la contrasenya no coincideixen conjuntament amb cap registre a la base de dades es retornarà el missatge: "wrong-user-or-pass".
- Si hi ha un error realitzant alguna de les consultes rebrem el missatge: "fatal-error".
- Si tot funciona correctament retornarem el nom complet especificat a la variable 'fullname'.

saveletters.php

El fitxer saveletters.php te la funció i el codi necessari per permetre guardar cadascuna de les lletres que anem trobant durant la partida sempre i quan siguem un usuari que ha iniciat sessió al joc prèviament.

Aquest registre es fa mitjançant una comanda **insert** a la base de dades que relaciona l'usuari que s'ha identificat amb la lletra que acaba d'encertar. La base de dades és la especificada al fitxer config.php.

```
<?php
    @include('config.php');

    $User = $_REQUEST['user'];
    $Letra = $_REQUEST['letra'];

    $query_id_usuario = "SELECT id FROM usuarios u where u.user = '$User'";
    $query_id_letra = "SELECT id FROM letras l where l.letra = '$Letra'";

    $response_query_id_usuario = mysql_query($query_id_usuario) or
die("fatal-error-usuario");
    $response_query_id_letra = mysql_query($query_id_letra) or die("fatal-
error-letra");

    if ((mysql_num_rows($response_query_id_usuario) > 0)
        &&(mysql_num_rows($response_query_id_letra) > 0)) {

        $row_id_usuario = mysql_fetch_array($response_query_id_usuario);
        $row_id_letra = mysql_fetch_array($response_query_id_letra);

        $insert = "insert into rel_usuarios_letras (id_usuario, id_letra)
values ('$row_id_usuario[0]', '$row_id_letra[0]')";
        $insert_response = mysql_query($insert) or die("fatal-error-insert");

        echo "ok";

    } else {
        echo "fatal-error-insert";
    }
?>
```

Quan un usuari registrat encerta una lletra, això ho guardem a la base de dades en una taula específica anomenada **rel_usuarios_letras**.

Aquesta relació entre usuaris i lletres es fa mitjançant el id de l'usuari que està jugant amb l'id de les lletres que va encertant.

Per guardar aquesta relació primer hem d'obtenir primer l'id de l'usuari i l'id de la lletra. Un cop obtinguts els guardem en aquesta taula.

Cridarem directament a aquest fitxer saveletters.php mitjançant una url. Per funcionar correctament la URL ha de portar correctament especificats els 2 paràmetres.

- 'user': usuari amb el que iniciarem sessió al joc.
- 'letra': és la lletra que l'usuari acaba d'encertar jugant.

Un exemple de la url que hauríem de cridar per enregistrar aquesta relació a la base de dades mysql és:

<http://146.185.148.96/unity/saveletters.php?user=lage&letra=z>

Amb aquest enllaç guardariem a la taula rel_usuaris_letras un nou registre amb l'id de l'usuari i l'id de la lletra.

Possibles sortides a aquesta crida:

- Si no trobem l'usuari tractant d'obtenir el seu id retornarem el missatge: "fatal-error-usuario".
- Si no trobem la lletra tractant d'obtenir el seu id retornarem el missatge: "fatal-error-letra".
- Si hi ha un error realitzant l'insert dels 2 id a la taula rel_usuarios_letras retornarem el missatge: "fatal-error".
- Si tot funciona correctament retornarem el nom complet especificat a la variable "fatal-error-insert".
- Si tot va be rebrem: "ok".

askletters.php

El fitxer askletters.php te la funció i el codi necessari per consultar les lletres que ha trobat un usuari que ha iniciat sessió al joc.

Farem diferents **selects** a la base de dades per obtenir totes les lletres que ha trobat ja que a la taula rel_usuarios_letras tenim relacionats als usuaris amb les lletres mitjançant l'ús dels seus id's. La base de dades és la especificada al fitxer config.php.

```
<?php
    @include('config.php');

    $User = $_REQUEST['user'];
    $query_usuario_letras = "SELECT l.letra, l.puntos FROM
rel_usuarios_letras ul, letras l
    where id_usuario in (SELECT u.id FROM usuarios u where
u.user='$User')
    and ul.id_letra = l.id;";
    $response_query_usuario_letras = mysql_query($query_usuario_letras) or
die ("fatal-error");

    if (mysql_num_rows($response_query_usuario_letras) > 0) {

        while($row = mysql_fetch_array($response_query_usuario_letras)) {
            $temp .= $row['letra'];
            $temp .= '-';
            $temp .= $row['puntos'];
            $temp .= '#';
        }
        echo $temp;
    } else {
        echo "0";
    }
?>
```

Cridarem directament a aquest fitxer askletters.php mitjançant una url. Per efectuar correctament la consulta, la URL ha de portar correctament especificat l'usuari.

- 'user': usuari amb el que iniciarem sessió al joc.

Un exemple de la url que hauríem de cridar per fer aquesta consulta a la base de dades mysql és:

<http://146.185.148.96/unity/askletters.php?user=pepa>

Possibles sortides a aquesta crida:

- Si la query no funciona perquè no troba l'usuari o falla la connexió rebrem: "fatal-error".
- Si l'usuari encara no ha encertat cap lletra rebrem el missatge: "0".
- Si tot va be rebrem una cadena de text amb totes les lletres que ha trobat i la respectiva puntuació assignada a cada lletra separades per '#': " 1-1#m-6#p-9#t-1#s-2#j-14#t-1#r-3#v-5#c-3#n-2#i-3#f-14#j-14#".

Problemes de connexió al servei MySQL

Bind-address mysql

Per defecte el mysql està configurat per permetre només connexions des de el mateix servidor i no des de peticions externes com podríem ser nosaltres des de el joc, o des de el navegador o emprant un software com mysql query browser.

He comentat la línia **bind-address 127.0.0.1** del fitxer de configuració de mysql: **/etc/mysql/my.cnf** per permetre connexions remotes però ara ja no puc accedir al phpmyadmin.

Haig de fer servir **mysql query browser**.

6. Conclusions

Estic bastant content del resultat obtingut. Vaig començar el projecte sense tenir idea de com fer un video joc. Vaig valorar la possibilitat de fer-ho sense emprar cap framework, únicament amb Java.

Valorar fer un dominó, un joc d'escacs o un billar amb gràfics 2d assignant dibuixos amb Java però això no es el que vull.

Vull dedicar-me al mon dels video jocs. Per això haig de conèixer les eines actuals i haver fet alguna cosa que al menys sigui visualment agradable.

Vull destacar la dificultat que suposa aprendre a emprar un framework tan complex com es Unity. El resultat és molt senzill i està clar que no el posaré a la venta però com a primera experiència estic molt satisfet. Cada petita implementació que volia fer em suposava hores de llegir documentació.

La corba d'aprenentatge d'un software com Unity és molt elevada. Ara em trobo capacitat per fer desenvolupaments molt mes complexos. M'agradaria treballar en equip amb rols diferenciats, i sobre tot despreocupar-me del disseny gràfic.

Treballar amb Unity ha sigut una experiència difícil però finalment gratificant tot i que el joc no vagi a ser un súper ventes.

Respecte al servidor, també ha sigut una experiència molt interessant. Des de la contractació del servei, configurar-lo i fer-lo funcionar. Ara dispo de d'una base de dades online amb els usuaris que es vulguin registrar i podria fer estadístiques de qui juga, des de on juguen, a quina hora, quina edat tenen, etc.

Podria augmentar contingut del joc via online. Emmagatzemant les traduccions dels textos online i afegint caràcters en altre idioma podria estendre el joc a altres abecedaris.

Puc sino aprofitar els scripts per a crear un joc d'avions multijugador, dotar-los de la funció de disparar. O aterrar i poder continuar a peu.

Sense dubte aquest projecte em dona la possibilitat de fer moltes coses que abans o emprant únicament eclipse i Java no m'hages sigut possible.