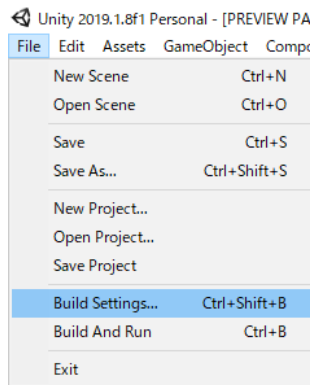


PREPARACIÓN Y COMERCIALIZACIÓN

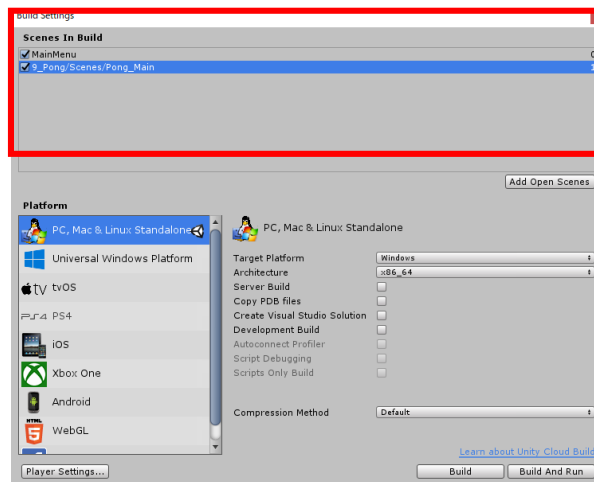
Exportación a PC:

A) Pasos para generar el ejecutable para Windows

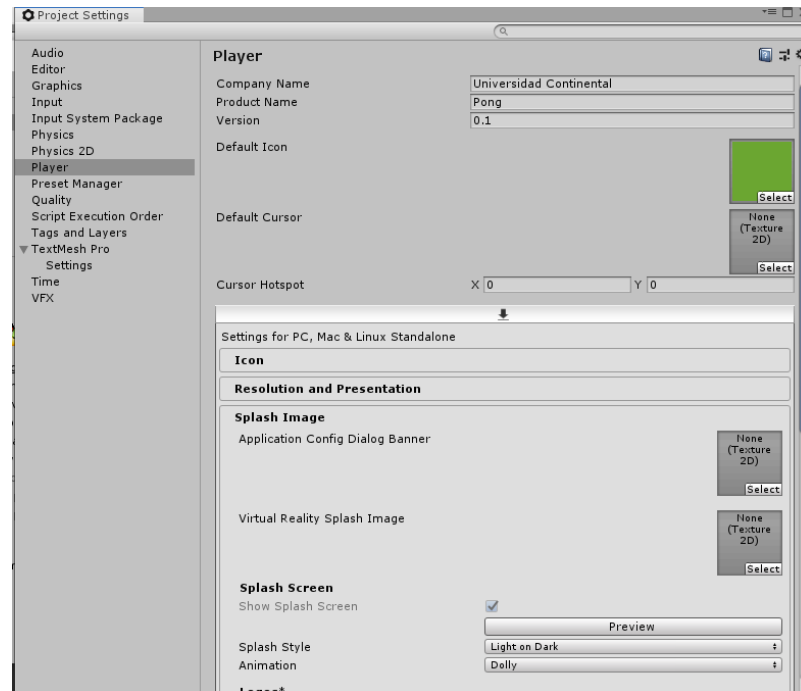
1. Abrimos la ventana de despliegue: File > Build Settings



2. Unity de forma automática instala los módulos necesarios para construir el ejecutable para la plataforma standalone de la computadora: Windows / MAC / LINUX. Tenemos que tener en cuenta agregar las escenas que queremos incluir en el build.



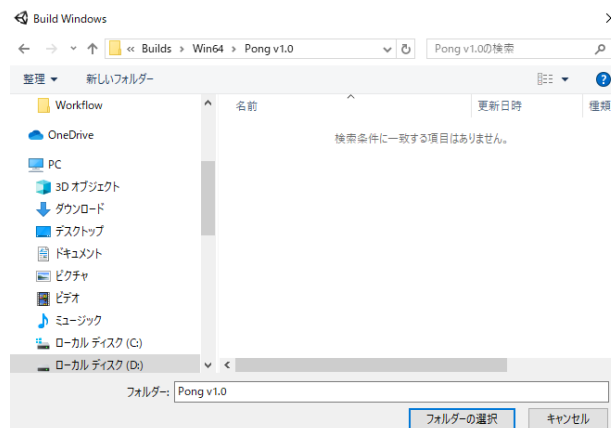
3. Antes de construir el ejecutable, podemos cambiar algunas opciones. Para ello presionamos la opción "Player Settings"



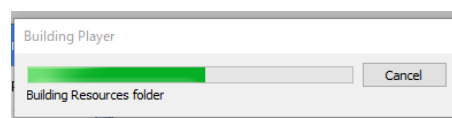
Podemos elegir un ícono, un cursor, e incluso un Splash, que es una secuencia de imágenes animadas mostrando los créditos, que se muestran al iniciar el ejecutable.

Para información adicional: <https://docs.unity3d.com/Manual/class-PlayerSettings.html>

4. Presionamos el botón “Build” y elegimos una ubicación de destino. En este caso se creó una carpeta dentro del proyecto pero fuera de Assets: <ProjectFolder>/Builds/Win64/Pong v1.0



El tiempo que toma dependerá de los recursos de la máquina, pero Unity te mostrará información acerca del build:



5. Una vez completado, podemos ejecutarlo y probarlo:

Unity_Fundamentals_Examples > Builds > Win64 > Pong v1.0				Pong v1.0の検索
名前	更新日時	種類	サイズ	
MonoBleedingEdge	2019/06/28 10:07	ファイル フォルダー		
Pong_Data	2019/06/28 10:07	ファイル フォルダー		
Pong.exe	2019/06/28 10:07	アプリケーション	636 KB	
UnityCrashHandler64.exe	2019/06/13 5:31	アプリケーション	1,421 KB	
UnityPlayer.dll	2019/06/13 5:30	アプリケーション拡張	23,399 KB	

Sub tema 2.2 Exportación a Android / IOS

A) Pasos para generar el ejecutable para Android

1. Para generar el APK para Android se necesita instalar módulos adicionales, para ello si no lo hicimos antes, tenemos que abrir Unity HUB y agregar el módulo de Android.



モジュールを加える			
Unity 2019.1.8f1 にモジュールを追加 : 使用可能スペースの合計 100.1 GB - 必要なスペースの合計 0 B			
Dev tools		ダウンロードサイズ	インストールサイズ
<input type="checkbox"/>	Microsoft Visual Studio Community 2017	1.0 GB	1.3 GB
Platforms			
<input checked="" type="checkbox"/>	Android Build Support	インストール済	2.0 GB
<input type="checkbox"/>	Android SDK & NDK Tools	891.8 MB	2.8 GB

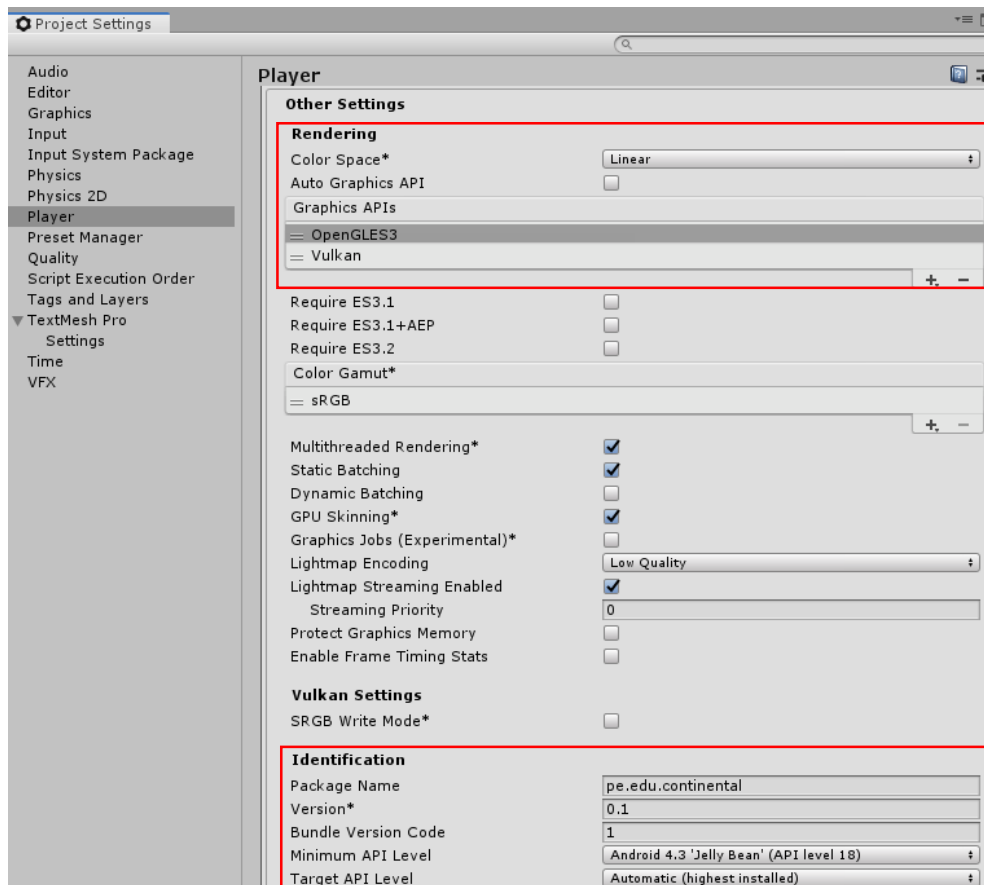
El módulo Android SDK & NDK Tools sirve para compilar nuestro código C# a C++ ganando un cierto porcentaje de performance, seguridad y compatibilidad, y menor uso de memoria (bytecode stripping) gracias al scripting backend IL2CPP: <https://docs.unity3d.com/Manual/IL2CPP.html>

2. Una vez instalado podemos cambiar el "Target" a "Android" en el panel de proyectos de Unity HUB.



- Para que permita crear el archivo APK, necesitamos cambiar el valor "PackageName" dentro Player > Identification, en los Project Settings > Player. Además, si la propiedad "Color Space" es definido como "Linear" no debe elegirse OpenGL ES 2 API como Graphics API, y el API mínimo de Android debe ser el 4.3.

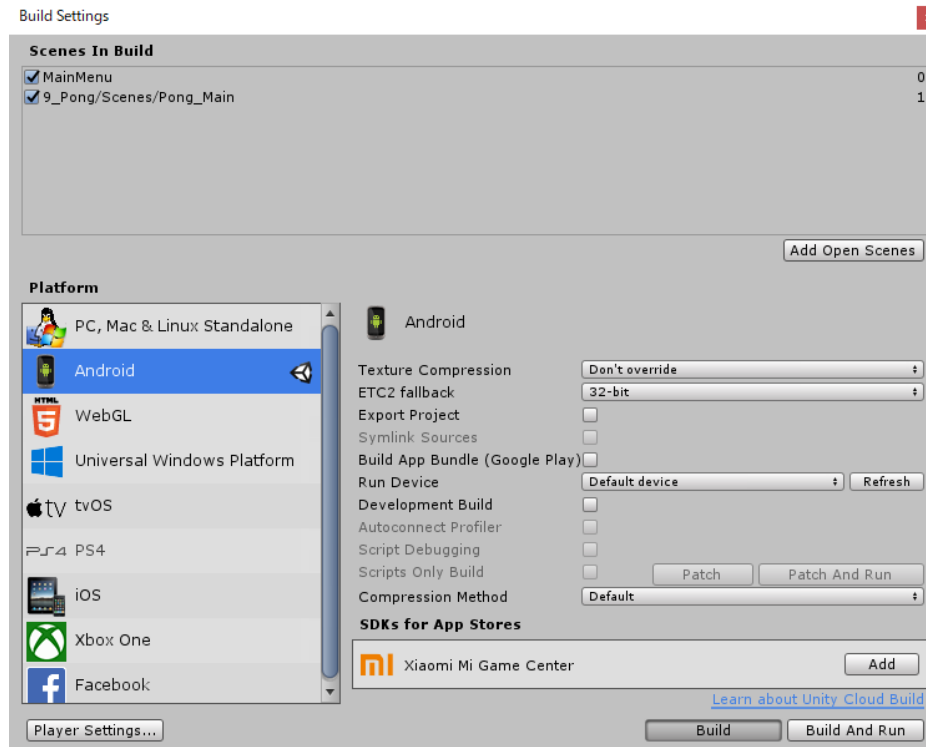
Nota: Adicionalmente tenemos muchas más opciones, se recomienda revisarlas para un mejor entendimiento de cómo generar APK efectivamente: <https://docs.unity3d.com/Manual/class-PlayerSettingsAndroid.html>



- Finalmente, al igual que el anterior ejecutable, abrimos la ventana de build: File > Build Settings. Está vez debemos elegir la Plataforma "Android", presionar la opción "Build" y elegir un lugar de ubicación para el archivo APK.

Nota: Si no aparece el botón "Build" o está bloqueado primero se debe cambiar de plataforma con la

opción "Switch Platform". Este proceso tomará algunos minutos dependiendo de la potencia de la computadora.



5. Al finalizar el proceso, podemos instalar el APK en nuestro dispositivo y probarlo.

Nota: Unity cuenta con una aplicación móvil (Unity Remote) que permite usar los sensores de nuestro Smartphone en el editor, para pruebas más eficientes para Android, IOS y tvOS, se recomienda mucho cuando se necesita el uso de los sensores o se quiera ver rápidamente cómo quedaría el producto final:

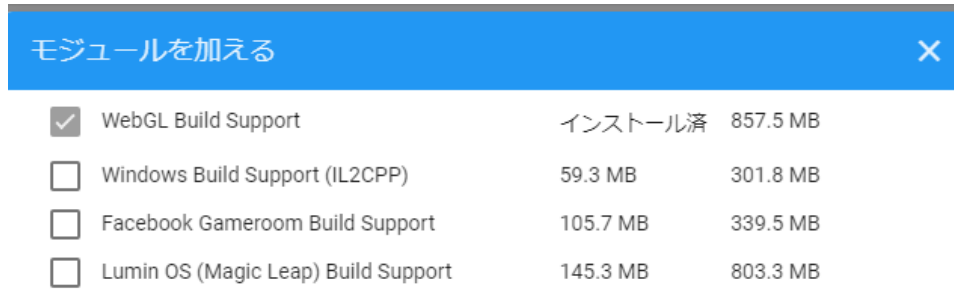
<https://docs.unity3d.com/Manual/UnityRemote5.html>



Exportación a WebGL

A) Pasos para generar el ejecutable para WebGL

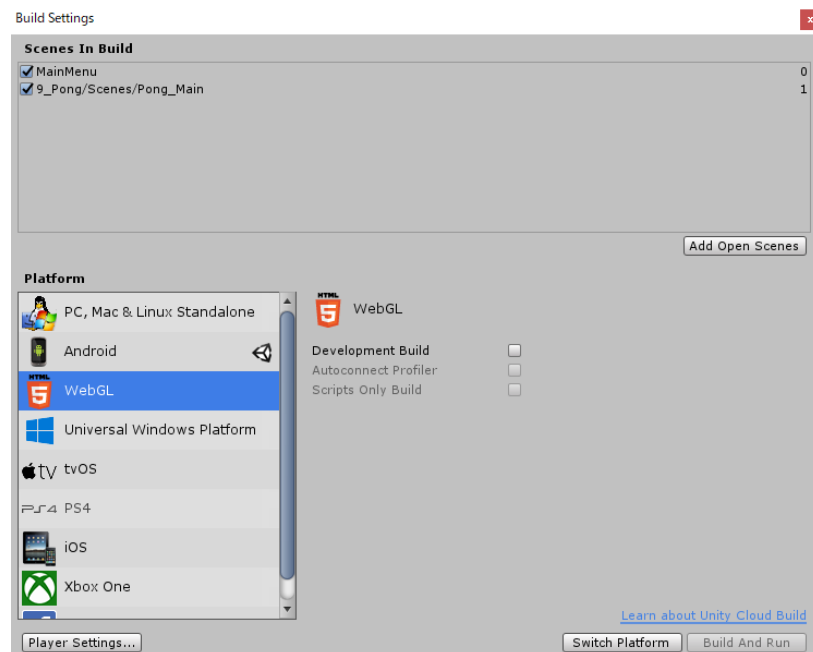
1. Para generar la versión WebGL, al igual que Android, se necesita instalar módulos adicionales, para ello si no lo hicimos antes, tenemos que abrir Unity HUB y agregar el módulo de WebGL



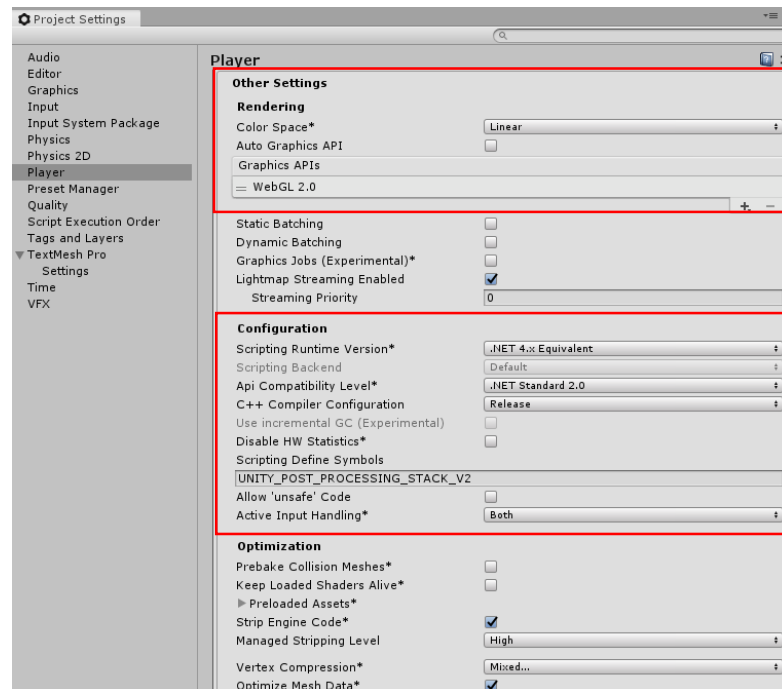
WebGL es una plataforma que permite renderizar gráficos a gran resolución en el explorador. Más información podemos encontrarla en el manual de Unity:

<https://docs.unity3d.com/Manual/webgl.html>

- Al igual que la plataforma Android, se necesita cambiar el “Target” de nuestro proyecto a WebGL usando Unity HUB o la opción “Switch Platform” dentro del editor de Unity.
- WebGL también cuenta con varias opciones en “Player Settings” (Edit > Project Settings > Player Settings) al momento de crear el ejecutable. Se recomienda revisarlas para una mejor comprensión de lo que nos ofrece: <https://docs.unity3d.com/Manual/class-PlayerSettingsWebGL.html>






Al igual que en Android, si se está usando como “Color Space” linear, se necesita especificar los Graphics API a la última versión:



4. Finalmente abrimos la ventana de builds: File > Build Settings y hacemos el build para WebGL.

Nota: WebGL usa el scripting backend IL2CPP para compilar los ensamblados a C++ nativo. Debido a eso tiene mayor tiempo de compilación. En esta página hay recomendaciones para disminuir el tiempo de generación de ejecutables cuando se usa IL2CPP: <https://docs.unity3d.com/Manual/IL2CPP-OptimizingBuildTimes.html>

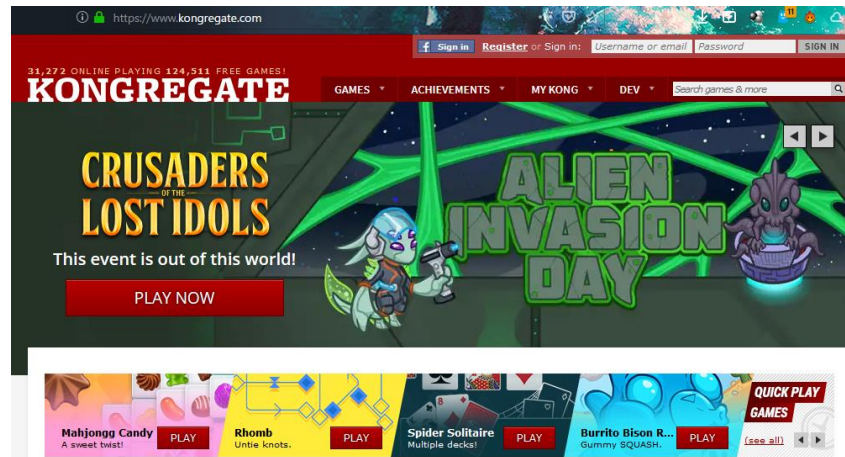
Projects > 2019-04 > Unity_Fundamentals_Examples > Builds > WebGL				
WebGLの検索				
<input type="checkbox"/> 名前	更新日時	種類	サイズ	
 Build	2019/06/30 15:17	ファイル フォルダー		
 TemplateData	2019/06/30 15:17	ファイル フォルダー		
 index.html	2019/06/30 15:17	HTML ファイル	1 KB	

Publicación y comercialización de videojuegos

- A) Presentación de páginas web especializadas en promocionar videojuegos Indie donde poder subir lo creado. El último paso es mostrar la aplicación al mundo para recibir valioso feedback en primer lugar, pero también se podría buscar con algo más de trabajo comercializarlo. Para ello vamos a listar algunas plataformas muy conocidas donde publicar nuestro trabajo:

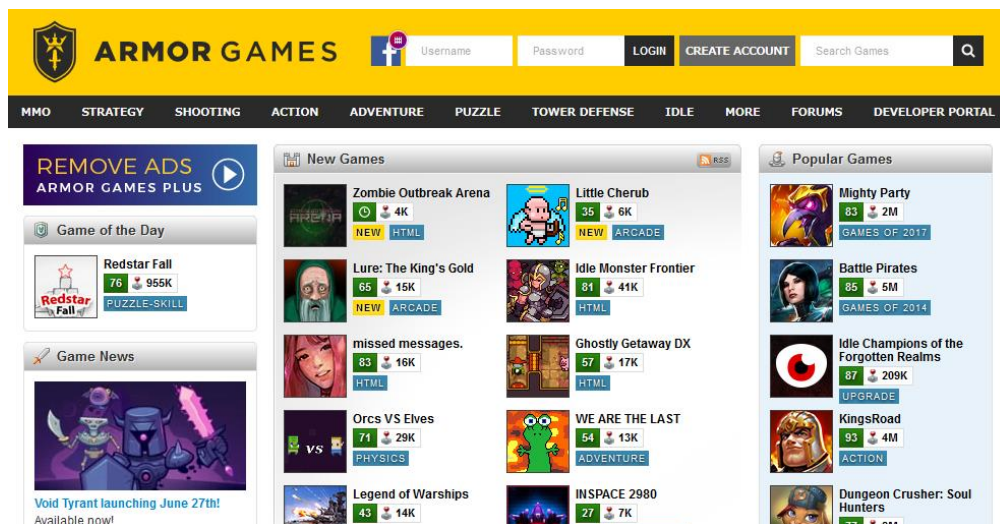
1. Krongregate: <https://www.kongregate.com/>

Es una página muy conocida por jugadores, pero también por desarrolladores, permite subir videojuegos básicamente en formato web (HTML5 / WebGL). Todos los juegos son gratuitos, pero se puede ganar por publicidad.



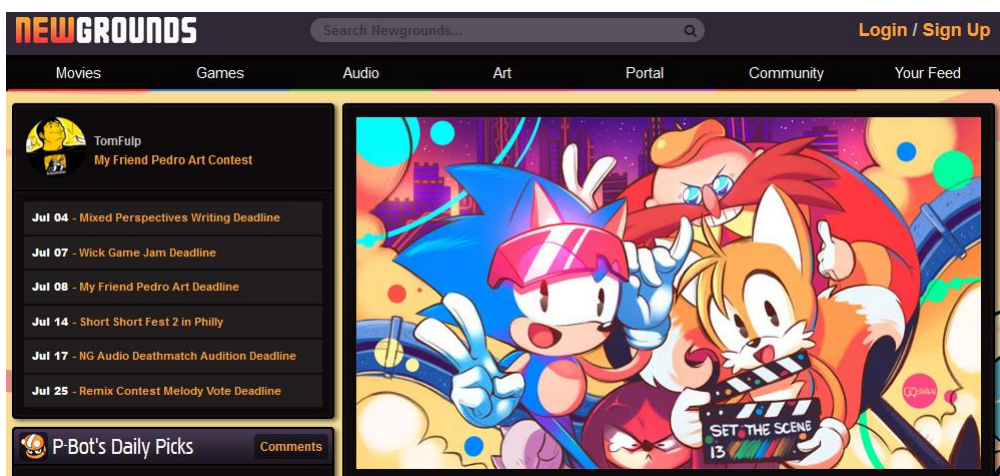
2. Armor games <https://armorgames.com/>

Al igual que Kongregate, esta página también te permite publicar videojuegos gratuitos en formato web y se puede obtener una ganancia por publicidad.



3. Newgrounds <https://www.newgrounds.com/>

Este portal web también te permite publicar videojuegos, pero adicionalmente también películas, música y arte. De la misma forma hay una ganancia por el número de visitas a tu trabajo.



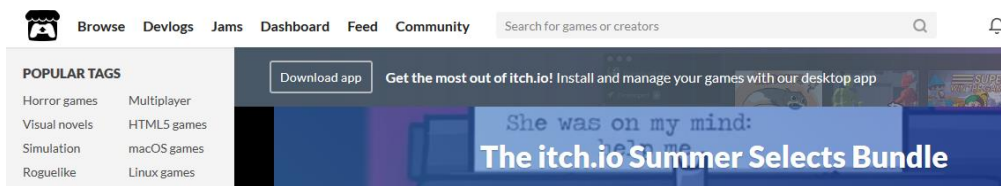
4. Miniclip <https://www.miniclip.com/games/en/>

Miniclip es un portal de juegos en línea que también te permite publicar tu trabajo en formato web, Android y IOS. Igualmente hay un porcentaje por publicidad.



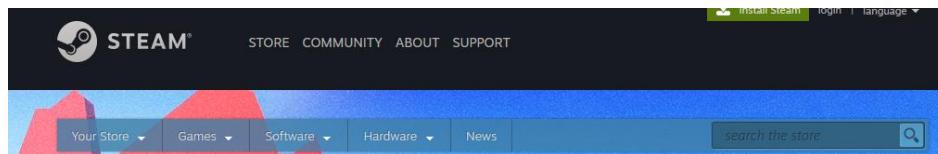
5. Itch.io <https://itch.io/>

A diferencia de los anteriores, el portal web itch.io da la posibilidad de publicar cualquier tipo de aplicación (videojuegos, softwares utilitarios, etc.) en cualquier formato (incluyendo PC), además te permite poner un precio para permitir la descarga de tu trabajo.



6. Steam <https://store.steampowered.com/>

Steam está dedicado en publicar videojuegos para dispositivos finales Standalone (Window MAC y Linux). Para Linux usa su propio sistema operativo SteamOS. Es una buena opción para vender un trabajo con mejores acabados.



7. Finalmente tenemos las tiendas respectivas para Android y IOS. Donde también se puede publicar nuestro trabajo.

Android: <https://play.google.com/>

IOS: <https://www.apple.com/ios/app-store/>

8. Adicionalmente a todas estas opciones se cuenta con campañas financiamiento Startups, las cuales también son una opción.