

Universidad Tecnológica Nacional
Facultad Regional Córdoba

Ingeniería en Sistemas de Información

Prof. Ricardo Medel



Análisis de comunidad de Software Libre

HaxeFlixel



Alumno:	Miguel Moyano
Año:	2017
Legajo:	52575
E-Mail:	mmoyano16@gmail.com

Córdoba, 20 de Junio de 2017

Índice

Proyecto de Software libre.....	3
Nombre del proyecto.....	3
Objetivo de HaxeFlixel.....	3
Descripción del proyecto.....	3
Características de HaxeFlixel.....	4
Enlaces relacionados.....	4
La comunidad en HaxeFlixel.....	5
Introducción.....	5
Principales desarrolladores.....	5
Portales de comunicación.....	5
Como se trabaja en la comunidad.....	6
Herramientas utilizadas.....	7
Otras formas de colaborar.....	7
Mi experiencia en la comunidad.....	9
Como ingresar a la comunidad.....	9
Propuesta de colaboración.....	11
Conclusión.....	13

Proyecto de Software libre

Nombre del proyecto

HaxeFlixel

Objetivo de HaxeFlixel

HaxeFlixel es un Framework para el lenguaje Haxe que tiene por objetivo brindar las librerías necesarias para llevar a cabo el desarrollo de videojuegos 2D que puedan ser compilados para múltiples plataformas.

Descripción del proyecto

HaxeFlixel es un proyecto fundado por Alexander Hohlov¹, el cual esta basado en un framework llamado Flixel para el desarrollo de juegos en ActionScript3 (AS3) (juegos flash).

El proyecto surge de la necesidad de superar las limitaciones que el propio ActionScript3 imponía: exportaba juegos en el formato de Adobe Flash bajo el runtime AIR. Con las frecuentes actualizaciones de los navegadores web y la tendencia a la diversidad de sistemas operativos (OSX, Windows, Linux, Android, iOS) los juegos que se basaban en flash poco a poco fueron perdiendo territorio debido a que estos entornos dejaron de soportar dicha tecnología. Además, en la industria se comenzaban a imponer herramientas que permitían el desarrollo para múltiples plataformas como lo son Unity, Cocos2D, GameMaker y otros motores que al día de la fecha siguen en vigencia.

Otro de los problemas con lo que se topa el desarrollo de videojuegos independientes, y otro de los motivos que impulsan el desarrollo de este proyecto, es la disponibilidad de una herramienta gratuita para los creadores de videojuegos, que no imponga limitaciones en cuanto a funcionalidad. Muchos de los motores y librerías en la actualidad son pagos, o bien son gratuitos con funcionalidades limitadas para ciertas mecánicas o compilación (el caso mas conocido es Unity, el cual tiene una versión gratuita muy limitada, y su negocio se basa en la venta de paquetes de desarrollo que amplían el rango de funcionalidad disponible).

Así es como Hohlov comienza la creación de HaxeFlixel, un framework basado en el viejo Flixel de AS3, pero a diferencia del anterior, desarrollado completamente en Haxe², un lenguaje OpenSource moderno el cual se caracteriza por disponer de una compilación multiplataforma (cross-compiler) que permite distribuir los programas realizados a distintas plataformas y que corran de forma nativa.

1 Conocido como Beeblerox en la comunidad de Github (<https://github.com/Beeblerox>)

2 Haxe: <http://haxe.org/>

De esta forma, HaxeFlixel se convierte en una gran alternativa libre y gratuita para desarrolladores independientes de videojuegos, y al ser OpenSource aprovecha el potencial de una gran comunidad que apoya el crecimiento y la actualización de esta gran herramienta.

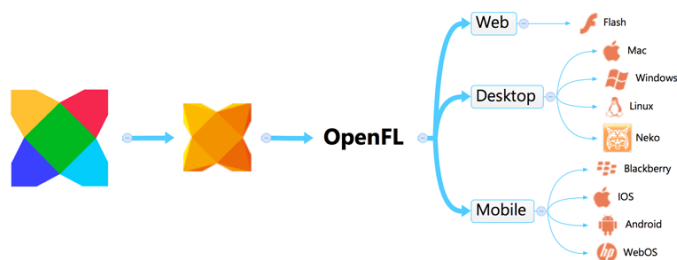


Ilustración 1: Esquema de los distintos tipos de compilación que ofrece Haxe

Características de HaxeFlixel

HaxeFlixel, gracias a la comunidad que lo respalda, constantemente se encuentra evolucionando y mejorando las soluciones que ofrece. Si bien la cantidad de características que lo conforman es gigante, algunas de las mas importantes son:

- Posibilidad de mostrar miles de objetos en movimiento simultáneamente en pantalla.
- Control de colisiones por hitbox y colisiones perfectas
- Manejo de conjuntos de objetos similares de forma sencilla
- Generación de partículas
- Creación de niveles basados en TileMaps³
- Manejo de textos e imágenes
- Funciones matemáticas avanzadas
- Motor de física incorporado
- Manejo de colores
- Manejo de sonidos y su interacción con el entorno y otros objetos
- Pathfinding
- Reciclaje de objetos

Algunas mejoras contra el Flixel de AS3:

- Desarrollado sobre un robusto y potente lenguaje como lo es Haxe, que permite la complicación multiplataforma. (esto incluye compilación para dispositivos móviles)
- Gran rendimiento utilizando aceleración gráfica de los GPU
- Herramientas de depuración de código
- Mejoras en el manejo de recursos gráficos y sonoros.

Enlaces relacionados

- Sitio oficial de HaxeFlixel: <http://haxeflixel.com/>
- Repositorio Github de HaxeFlixel: <https://github.com/HaxeFlixel/>
- Sitio oficial de Haxe: <http://haxe.org/>

3 Una de las herramientas mas utilizadas es <http://www.mapeditor.org/>

La comunidad en HaxeFlixel

Introducción

HaxeFlixel es un gran proyecto colaborativo que tiene un gran numero de colaboradores que mejoran constantemente las características que lo conforman y proveen de un gran numero de tutoriales para hacer mas fácil la labor del desarrollador independiente.

Principales desarrolladores

Beeblerox (Alexander Hohlov), como creador del Framework es claramente el principal desarrollador y el que comanda el desarrollo de esta tecnología. Sin embargo, con el tiempo, HaxeFlixel se ha vuelto producto de la colaboración de grandes desarrolladores que continúan y mejoran constantemente su trabajo:

- **Beeblerox:** Creador y principal desarrollador del framework
- **Chris:** Mantenimiento de los sitios web, foros, logos y desarrollador de herramientas y correcciones de gran parte de los Bugs de HaxeFlixel
- **GameDevSam:** Creador de grandes aportes a la optimización de la librería.
- **Gama11:** Administración de la documentación principal.
- **Larsiusprime:** Principal desarrollador de la librería UI de HaxeFlixel. Desarrollador de Defender's Quest, uno de los principales Demos del framework
- **ProG4mr:** Uno de los principales desarrolladores del motor de física de la librería.
- **Sergey Miryanov:** Veterano en la comunidad. Participa en la toma de decisiones principales

Más allá de los desarrolladores principales, HaxeFlixel es soportada por muchísimos desarrolladores más: al día de hoy, el repositorio principal cuenta con 1002 desarrolladores que interactúan con el código del framework, y cuenta con 50 colaboradores aproximadamente para el resto de los repositorios asociados.

Portales de comunicación

HaxeFlixel cuenta con un gran numero de portales desde los cuales los distintos colaboradores y desarrolladores independientes pueden tener contacto con el equipo principal:

- **Foro oficial:** <http://forum.haxeeflixel.com/>
- **IRC:** #haxeeflixel
- **Twitter:** @HaxeFlixel
- **Github:** <http://github.com/haxeeflixel>
- **IndieDB** (publicacion de juegos independientes):
<http://www.indiedb.com/engines/haxeeflixel>
- **Reddit:** <https://www.reddit.com/r/haxeeflixel>

- **Slack:** <https://haxeflixel.slack.com/>
- **Steam:** <http://steamcommunity.com/groups/haxeflixel>
- **Discord:** <https://discord.gg/0uEuWH3spjck73Lo>

Como se trabaja en la comunidad

El núcleo de HaxeFlixel se encuentra en su sitio de Github. El mismo cuenta en la actualidad con 12 repositorios que dividen los distintos aspectos que forman parte del Framework, entre los mas importantes:

- **Flixel:** Core de HaxeFlixel. Contiene las funcionalidades principales y es el repositorio con mas actividad por parte de la comunidad
- **Demos:** Colección de un gran numero de demos desarrollados con HaxeFlixel, que hacen de guia para los desarrolladores que utilizan esta tecnología para sus juegos
- **UI:** librería GUI para HaxeFlixel
- **Addons:** Conjunto de clases adicionales, desarrolladas en gran parte por la comunidad, que incrementan la funcionalidad ofrecida por HaxeFlixel. Entre los principales addons, FlxCamera para el manejo de cámaras y Nape para el manejo del motor de física.
- **Tools:** aplicaciones de linea de comandos para la creación de templates y demos de forma automática.
- **Docs:** la documentación que respalda a HaxeFlixel.

Cada uno de estos repositorios están puestos de forma pública de forma que el código pueda ser accedido por la comunidad para colaborar y aprender.

La forma de trabajar es sencilla: Se colocan las características a desarrollar, los bugs encontrados y demás tareas a realizar en el correspondiente tablero de Issues. Quienes colaboran, están constantemente en contacto con esta lista de y seleccionan aquellas en las que sienten que pueden ayudar. Estas tareas constantemente se cargan de comentarios, tanto del encargado de la tarea como de aquellos usuarios que desean colaborar en las discusiones sobre el desarrollo de la misma, dando consejos, criticas constructivas y supervisión de lo que el colaborador realiza. La comunidad de HaxeFlixel, a diferencia de otras, manejan de forma sencilla las etiquetas para estas Issues, disponiendo de la cantidad justa y necesarias para poder clasificar las actividades a llevar a cabo.

Para el desarrollo, poseen básicamente 3 ramas de trabajo: una versión estable (master), una de desarrollo (dev) y una de documentación (doc), las cuales son clonadas por los colaboradores para comenzar con el desarrollo.

Una vez que un colaborador lleva a cabo la tarea pendiente, este hace una solicitud al servidor para combinar su código con el de la versión estable (Pull Request). Al igual que con las Issues, la PR (Pull Request) recibe un gran numero de comentarios por parte de la comunidad y sobre todo de los desarrolladores principales, que están en constante contacto con los colaboradores para supervisar que lo que se desarrolla sea de la mejor calidad posible.

Como la comunidad esta formada por personas que no solo dedican su tiempo para HaxeFlixel sino que también tienen sus actividades y trabajos, es que se proponen normas de

trabajo para facilitarle el trabajo a quienes supervisan y quienes realizan las pruebas para que el código de los colaboradores pueda ser fusionado con las versiones estables:

- Describir claramente los problemas/issues relacionados al código trabajado
- Respetar las normas de codificación propuestas por el equipo (<http://haxeflixel.com/documentation/code-style/>)
- Realizar buenas y exhaustivas pruebas sobre el código desarrollado
- Proveer un link con un demo que implemente la funcionalidad nueva, y que compile y funcione correctamente

Una vez que la nueva funcionalidad pasa las pruebas realizadas por colaboradores, que se aprueba la checklist impuesta por los desarrolladores principales, y luego de la aprobación general de la comunidad, el código desarrollado se combina con la versión oficial de HaxeFlixel y la misma queda a disposición de todos aquellos que deseen estudiar y utilizar el código.

Herramientas utilizadas

Al ser un framework sobre un lenguaje de programación (Haxe), no existe una restricción sobre que herramientas utilizar para el desarrollo. Aun así, el equipo de HaxeFlixel propone 3 IDEs con los que se puede trabajar cómodamente y que disponen de extensiones especializadas para el framework y que fueron desarrollados por la misma comunidad:

- SublimeText (<https://www.sublimetext.com/>)
- Visual Studio Code (<https://code.visualstudio.com/>)
- FlashDevelop (<http://www.flashdevelop.org/>)

Ademas, existe un sin fin de aplicaciones y portales que son utilizados por los desarrolladores para crear juegos y compartirlos con la comunidad. Algunos ejemplos:

- Paint.Net: potente editor gráfico de uso gratuito que ofrece un gran numero de herramientas como manejo de capas, transparencias, efectos y demás. (<https://www.getpaint.net/>)
- OpenGameArt: portal en donde usuarios pueden compartir distintos recursos audiovisuales para incorporar a los desarrollos de videojuegos. Normalmente trabajan con licencias Creative Commons. (<https://opengameart.org/>)
- Audacity: Software de edición de audio OpenSource con todo lo que necesita un músico para dar vida a los juegos. (<http://www.audacityteam.org/>)

Otras formas de colaborar

Si bien se describió la forma en que los desarrolladores que colaboran con HaxeFlixel crean el código de funcionalidades y ejemplos, existen muchas otras formas en las que se puede ayudar, las cuales son descriptas claramente por el equipo principal en la documentación del Framework:

- Publicitar HaxeFlixel: en los medios que los colaboradores consideren mejor. Mientras mas se expanda, mayor colaboradores vendrán a HaxeFlixel y mejores funcionalidades ofrecerá el framework. Algunas de las propuestas por el equipo son:
 - Seguir en las distintas redes sociales

- Puntuar el repositorio en Github
 - Escribir entradas en blog y difundir el trabajo realizado por la comunidad.
- Realizar tutoriales: Tanto en formato de blog, manual o video tutoriales. Son de gran ayuda para quienes inician en este mundo
- Donaciones: Se aceptan donaciones en su pagina de Patreon.

Mi experiencia en la comunidad

Particularmente, me encontré con HaxeFlixel en un curso de desarrollo de videojuegos 2D, llevado a cabo de forma gratuita por el equipo de Fundasoft, quienes se dedican a la investigación y desarrollo de software con el objetivo de aportar a la comunidad con soluciones tecnológicas sin ningún costo.

Luego de realizado el curso, comencé a desarrollar juegos como un pasatiempo, y a medida que avanzaba notaba que hay muchas cosas que se pueden aportar y mejorar en la comunidad de HaxeFlixel. Entre algunas de ellas:

- Existe poca documentación en español, y seria bueno brindar para la comunidad de habla hispana tutoriales y ejemplos que permitan aprender de esta gran herramienta para crear videojuegos
- Si bien existen muchos demos en el repositorio oficial de Github, creo que existen muchos tipos de juegos que aun no han sido contemplados y que seria bueno desarrollar y compartir para que sean ejemplo para quienes quieran crear juegos similares
- De la mano de lo anterior, se pueden desarrollar mini librerías sobre HaxeFlixel para determinados géneros de juegos.

Así es que, y en base a un juego que siempre quise realizar, me propuse colaborar con un demo/addon para el desarrollo de juegos de estilo BeatEmUp 2D con HaxeFlixel



Ilustración 2: Golden Axe, reconocido juego del género Beat Em' Up

Como ingresar a la comunidad

En primer medida, lo importante es ingresar al sitio oficial de HaxeFlixel y seguir las indicaciones que el equipo detalla para saber como colaborar. Si bien la comunidad es algo nueva, tiene un tamaño considerable y es por eso que, con el tiempo, han sabido organizar todo el contenido y las pautas de trabajo, por lo que uno fácilmente puede obtener la información de como colaborar y como comunicarse.

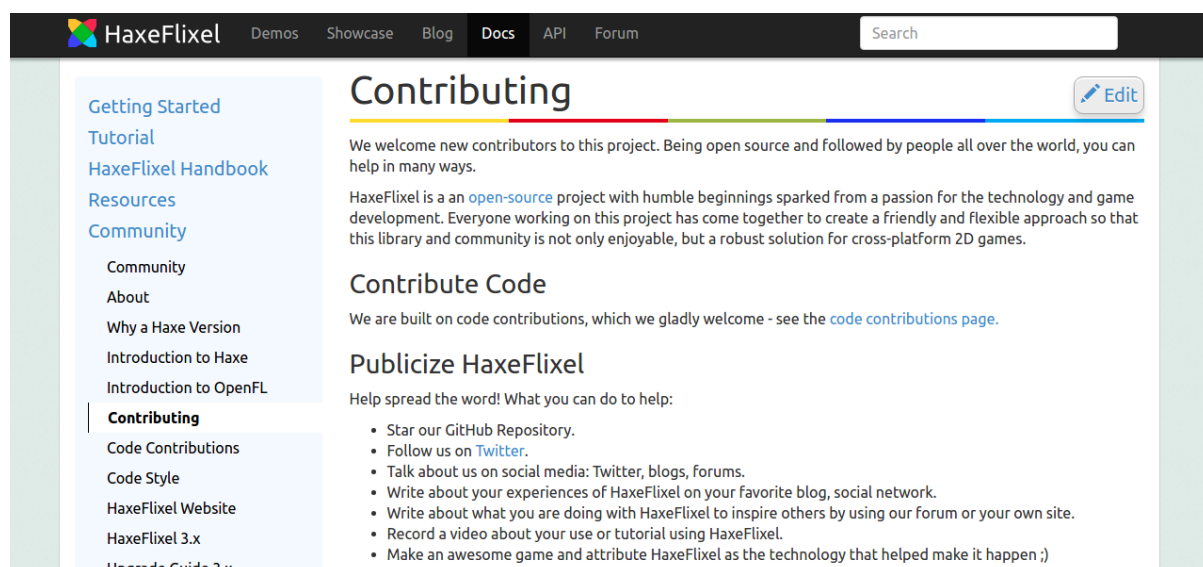


Ilustración 3: Información sobre como contribuir con la comunidad

En principio, para colaborar, se solicita en la pagina que se envíe un Tweet a @HaxeFlixel escribiendo solo la dirección de correo. Esto hace que el equipo de HaxeFlixel te envíe una solicitud para unirse al Slack oficial, en donde se encuentran canales para hablar sobre distintos aspectos para colaboraron.

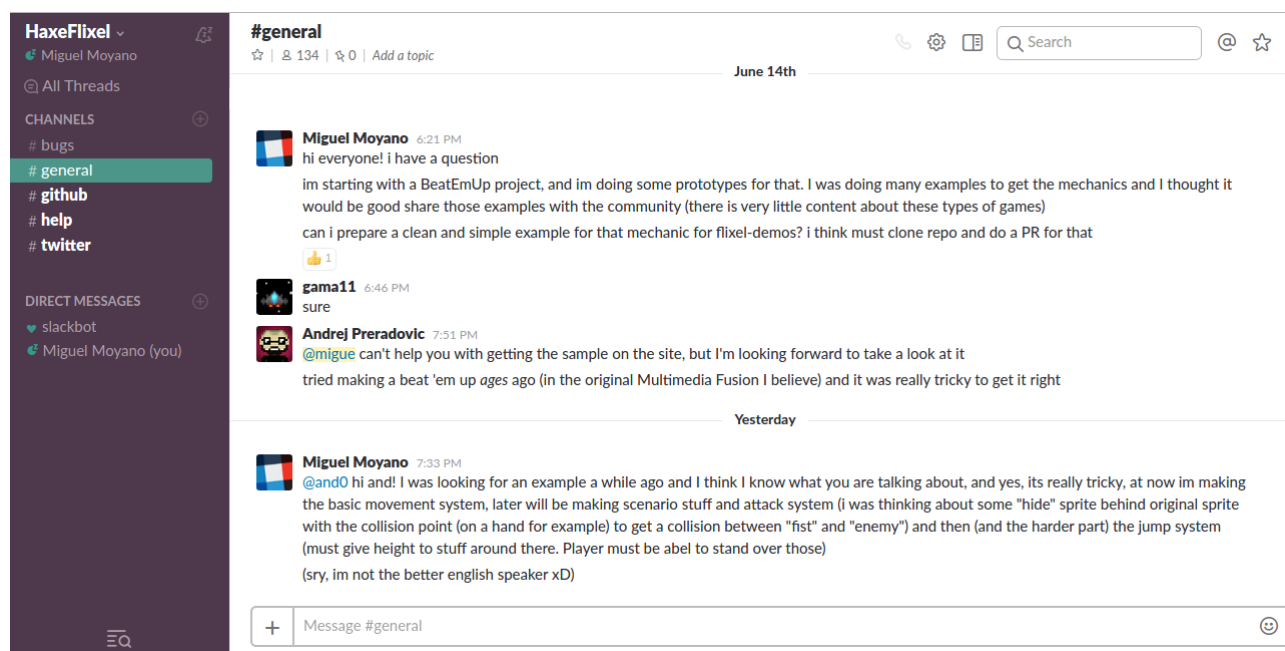


Ilustración 4: Chat principal de Slack, donde se puede interactuar con miembros de la comunidad de forma directa

Una vez dentro es importante presentarse y comentar tu situación actual, de donde sos, cuales son tus expectativas. En mi caso, la comunidad me recibió con los brazos abiertos, e incluso me di con que uno de los youtubers que sigo que se dedican al desarrollo “en vivo” de juegos, estaba como colaborador, a lo que me identifica (siempre veo sus videos) y me saluda gratamente.

Es importante comunicarse en ingles, tanto en el chat como en el foro, porque la gran mayoría de los desarrolladores son de habla inglesa. Sin embargo, entienden que existen

colaboradores de todas partes del mundo, y si hay problemas de redacción no hacen problemas, sino que tratan de entender y ayudar. Es una comunidad muy calida que te da lugar a que aportes tu grano de arena.

Propuesta de colaboración

En cuanto a mi colaboración, propuse la realización de un ejemplo de juego BeatEmUp en el canal general del Slack de HaxeFlixel, a lo que me contestaron con una aprobación. Básicamente, se debe realizar un clon del repositorio, trabajar sobre tu ejemplo y luego hacer un Pull Request como se comentaba anteriormente, siguiendo con todas las normas de la comunidad.

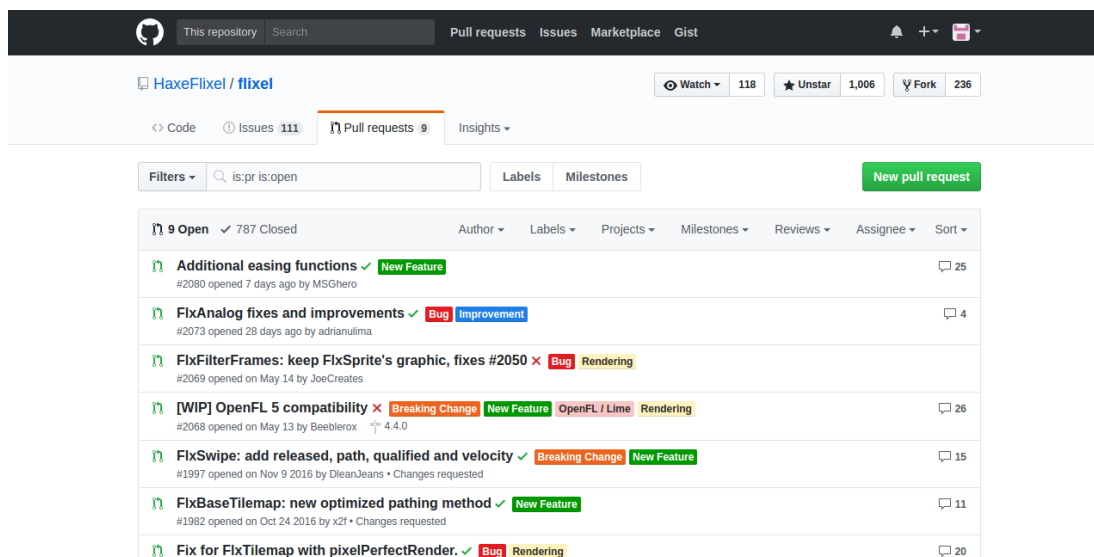


Ilustración 5: Listado de Pull Request del Core de HaxeFlixel, pendientes de aprobación

Actualmente me encuentro desarrollando las mecánicas principales para este tipo de juegos, que es compleja al manejar un espacio denominado como 2.5D, ya que tiene que manejar profundidad, gravedad y otros aspectos que son comunes en entornos 3D.

La idea es realizar en primer medida un ejemplo sencillo en las que se plasmen las mecánicas básicas y a su vez proveer clases que sirvan de base (como un pequeño framework) para el desarrollo de este tipo de juegos (aportes en flixel-demos y en flixel-addons)

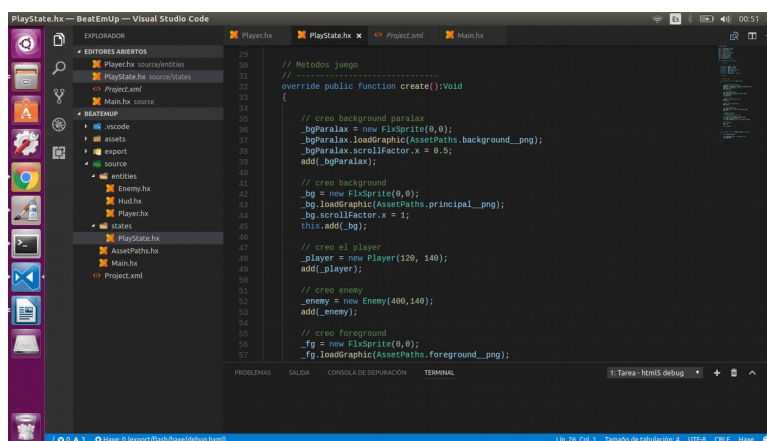


Ilustración 6: Programación HaxeFlixel en Visual Studio Code

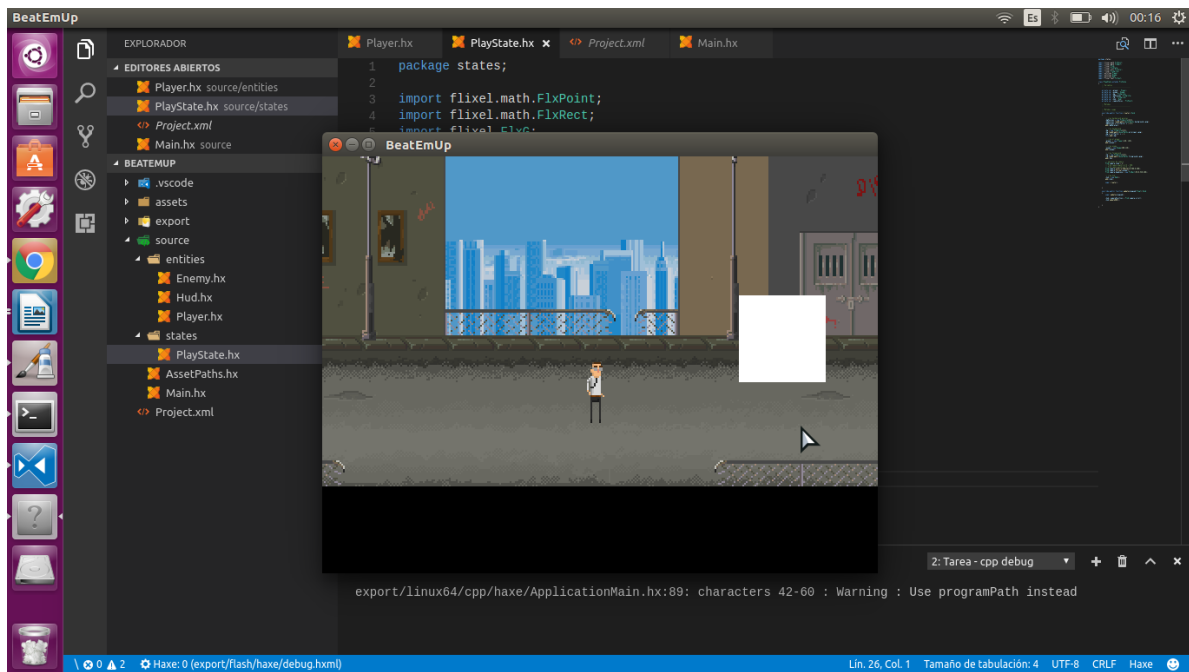


Ilustración 7: Ejecución de la prueba compilando en C++

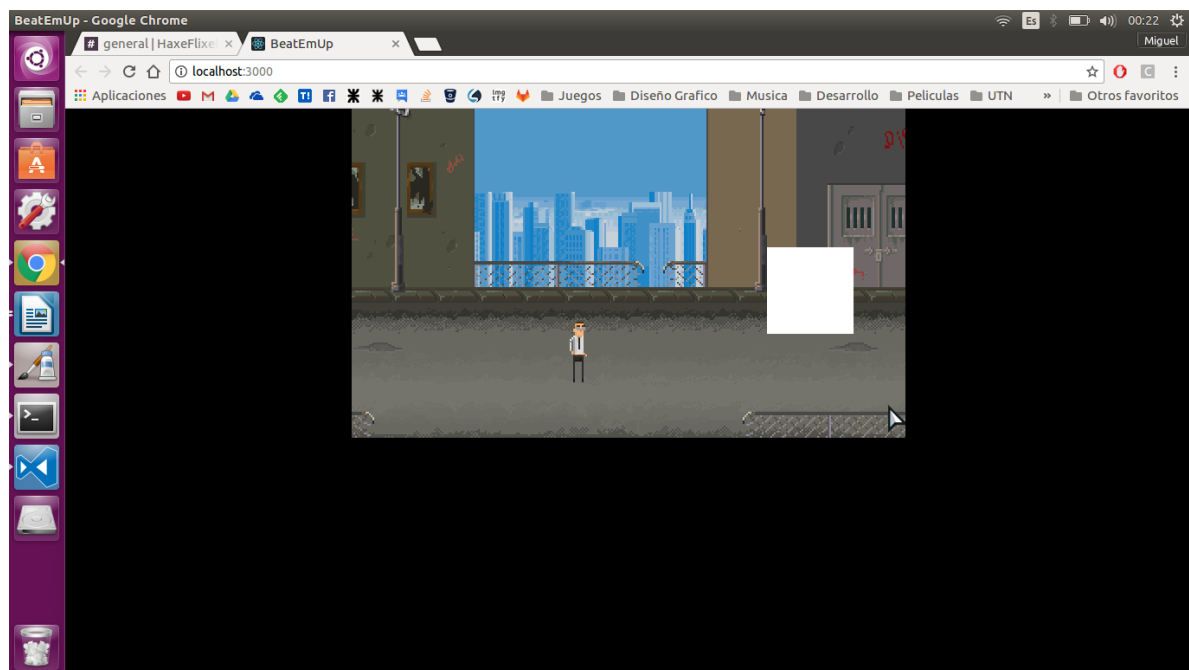


Ilustración 8: Ejecución de la prueba compilando en Javascript/Html

Lo bueno es que la comunicación es muy sencilla y directa: manejarse por Slack, comentar lo que sea necesario, con una respuesta rápida y agradable por parte de la comunidad. No se deben dar tantas vueltas para poder plasmar inquietudes o sugerencias. La documentación esta bastante clara, tanto la relacionada con el código como la relacionada con la comunidad, por lo que la mayor parte de las dudas pueden ser resueltas de forma rápida en la misma pagina de HaxeFlixel.

Por ultimo, estoy planificando realizar una serie de video-tutoriales para Youtube que muestren ejemplos para distintos tipos de mecánicas de juegos, simples, pero que plasmen lo esencial para cada modalidad, y en español, para que los usuarios de habla hispana no se vean

marginados de este gran framework. La idea es publicar estos videos y poder llevar a cabo un taller gratuito en la facultad de la mano de la gente de CoderDojoCordoba.



Conclusión

El mundo del desarrollo de videojuegos es enorme, así como las herramientas que este nos ofrece para crear nuestros proyectos. Si bien los productos que son pagos nos ofrecen facilidades para trabajar, existen muchas alternativas gratis y de fuentes abiertas que nos permiten llevar a cabo un desarrollo serio y potente de un videojuego (tanto herramientas para la programación como para el diseño de imágenes y edición de sonido), y muchas veces estas alternativas son beneficiosas para los desarrolladores mismos, quienes al participar en conjunto con una comunidad, incrementan sus conocimientos y capacidades.

HaxeFlixel es un entorno de desarrollo novedoso y moderno, que brinda una alternativa muy buena y eficiente para los creadores de videojuegos independientes, y sin duda alguna va ganando lugar en la industria, así como en numerosas jornadas y competencias en donde chicos y grandes, independientes y pequeños emprendimientos optan por esta potente herramienta para sus creaciones.