**BATTLE FRONT**

**López Pablo 38601565**

[*pabloflv@yahoo.com*](mailto:pabloflv@yahoo.com)

**Lucero José 42978101**

[*jose.maximiliano.lucero@gmail.com*](mailto:jose.maximiliano.lucero@gmail.com)

**Santirachi Damián 39810265**

[*dsantirachi@gmail.com*](mailto:dsantirachi@gmail.com)

**Verges Nehuen 39407041**

[*nehuenverges@hotmail.com*](mailto:nehuenverges@hotmail.com)

**Yáñez Jonathan 45188238**

[*Jonathanalexis98@hotmail.com*](mailto:Jonathanalexis98@hotmail.com)

****

“Estudiantes de Segundo año cursando la carrera de licenciatura en sistemas en la Universidad Nacional de Lanús”.

**Resumen:**

El presente trabajo tiene como objetivo detallar como fue el desarrollo de un juego programado desde cero utilizando el lenguaje de programación Python y una librería orientada a la elaboración de juegos en 2D llamada Pygame. Para la confección del mismo se tomó como idea base a unos de los primeros juegos que fueron creados, el “Pong” con el fin de adaptarlo a los tiempos actuales. Para ello el juego se realizó poniendo en práctica el uso de herramientas como Photoshop, orientado para la parte gráfica del desarrollo y GITHUB un controlador de versiones que se usó con el fin de poder tener un mejor control del proyecto. Otro de los objetivos que tuvo este trabajo fue poner en práctica el uso de las metodologías agiles para la óptima organización del grupo de trabajo. La metodología empleada en este proyecto fue Kanban.

***Palabras claves:*** Pong, Pygame, Python, Metodologías agiles, Kanban, Github, Photoshop, Unla, Sistemas, Trabajo practico

**Abstract:**

The present work aims to detail how was the development of a game scheduled from scratch using the Python programming language and a library oriented to the development of 2D games called Pygame. For the preparation of the same was taken as a basic idea to one of the first games that were created, the "Pong" in order to adapt to current times. For this the game was made putting into practice the use of tools like Photoshop, oriented for the graphic part of the development and GITHUB a version controller that was used in order to have a better control of the project. Another of the objectives of this work was to put into practice the use of agile methodologies for the optimal organization of the working group. The methodology used in this project was Kanban.

***Keywords:*** Pong, Pygame, Python, Agile methodologies, Kanban, Github, Photoshop, Unla, Sistems, Practical work

**Introducción:**

En la actualidad a la hora de realizar un proyecto es muy común usar distintas herramientas y metodologías para la gestión del mismo con el fin de darle un mayor seguimiento y garantizar una óptima elaboración del mismo, es por eso que se puso en marcha este trabajo, el cual tiene como objetivo poner en práctica el uso de estas herramientas para futuros proyectos laborales. Nos propusimos elaborar un juego desde cero pero tomando como base la idea de uno de los primeros juegos que fue creado, el “Pong”. Creamos un grupo de trabajo y lo organizamos a través del uso de una metodología ágil denominada Kanban, actualizamos el juego creando personajes y dándoles una historia acorde a la sociedad estadounidense de la década del 90’.

El código fue creado desde cero utilizando el lenguaje de programación Python con su respectiva librería Pygame que nos permitió agregarle nuevas funcionalidades al juego dándole competitividad y calidad gráfica a los usuarios sin que pierda la esencia clásica de un juego en dos dimensiones. Para poder realizar el proyecto se fomentó el uso de Github, un controlador de versiones en el cual se fueron guardando los avances permitiéndonos así facilitar la gestión de los cambios que se iban dando en el trabajo. Además para la elaboración del juego se usaron herramientas clásicas en el diseño de proyectos como diagramas y esquemas.

**Documentación Técnica:**

Flujo de ejecución del programa:

Al arrancar el juego lo primero que hace es declarar y setear las variables que se usarán a lo largo del programa, así como iniciar la instancia de ventana para mostrar el menú.

Luego entra en un loop para verificar las entradas del usuario y determinar la opción elegida, el cual verifica según la ocurrencia de un evento keydown el presionado de una tecla y luego verifica cual fue esa tecla para determinar que hacer a continuación.

Acorde a la selección del usuario el juego puede enviarlo a una partida, una tabla de puntajes, un tutorial del juego o salir del programa.

Sección Partida:

La sección partida (start) opera con un flujo similar al del menú: luego de setear las variables necesarias para la operación de la sección, entra en un bucle que inicializa los objetos y variables necesarios para una partida y actúa como “precursor” del bucle interior, el cual corresponde a cada partida de manera individual.

Cuando el bucle interior interrumpa su ejecución, es decir, la partida termine, ya sea saliendo con la tecla escape, por tiempo o porque algún jugador ganó, el bucle precursor retomará el control y acorde al tipo de salida del bucle de partida reinicializará los objetos y variables o volverá al menú.

Sección de Puntajes:

La sección de puntajes (score) es la encargada de mostrar los mejores puntajes de las partidas que se hayan jugado. Opera cargando 10 puntajes almacenados en un archivo, mostrándolos en pantalla y aguardando a que el usuario presione la tecla escape para volver al menú.

Sección Tutorial:

La sección tutorial (how to game) se encarga de mostrarle al usuario los controles del juego, el flujo de esta sección es simple, una vez que se renderiza en pantalla el texto de los controles, el programa queda a la espera de que el usuario presione la tecla para volver al menú.

**Problemas encontrados:**

Angulo de rebote para la pelota:

Uno de los primeros problemas con los cuales nos encontramos durante el desarrollo del juego fue determinar el ángulo y velocidad de rebote de la pelota cada vez que esta chocaba contra una pared o un jugador. La resolución a este problema lo encontramos en la librería pygame misma, la cual trae integrada una función para encargarse de las colisiones.

Una vez resuelto ese problema quedo la cuestión de cómo manejar el rebote, es decir, dependiendo de dónde pegase hacia donde iba a salir. Esto se resolvió partiendo el “rectángulo” de los sprites de los jugadores en 4, si la pelota pega en la sección superior del rectángulo rebota hacia arriba, sino hacia abajo, y si rebota en la sección delantera rebota hacia adelante, de lo contrario hacia atrás.

Las colisiones con las paredes se manejan de manera similar, una vez que se detecta una colisión se revisa si la pelota golpeó el techo o el piso del rectángulo que conforma al área de juego y acorde a eso rebota hacia arriba o abajo, los lados verticales siempre conservan el movimiento en el eje Y, aunque lo invierten en el eje X.

Presionado de múltiples teclas al mismo tiempo:

Otro problema encontrado durante el desarrollo fue manejar el presionado de varias teclas al mismo tiempo, lo cual es vital para que pueda haber más de un jugador manejando su personaje al mismo tiempo y para que el movimiento de estos en diagonal sea posible.

Esto se resolvió implementando una lista de teclas presionadas, la cual se actualiza a medida que van llegando eventos de presionado o levantado de las teclas que corresponden al movimiento de cada jugador. Esta lista luego se revisa para determinar, por jugador, las teclas presionadas y determinar la dirección en la cual se moverá cada uno.

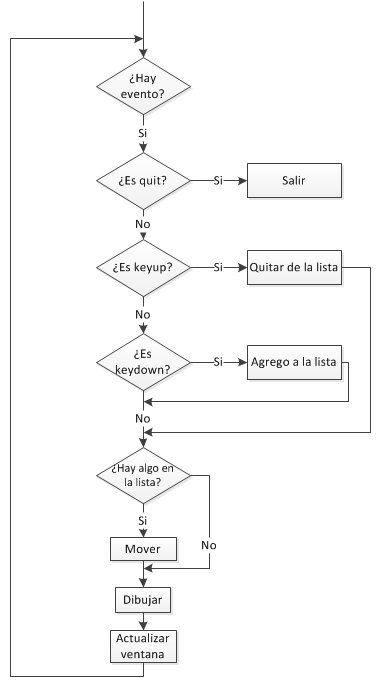
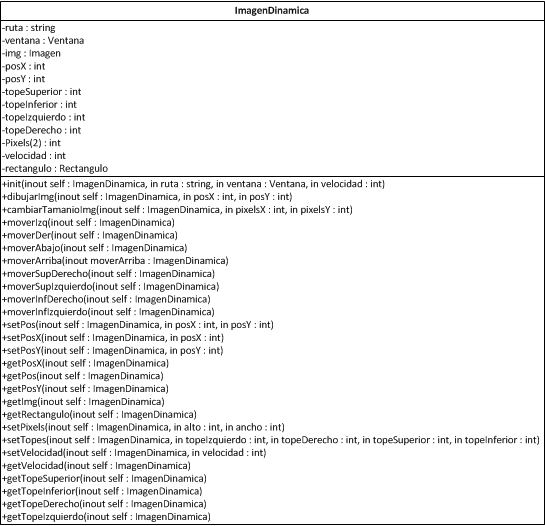
****

Diagrama de clases:

Por su estructura, el programa no requirió más que una clase, la cual está representada a continuación:



**Historia del juego:**

El juego creado se basa en el “Pong”, un juego diseñado en la década del 70 y que representaba un partido de tenis entre dos jugadores. Siguiendo ese tipo de jugabilidad en 2 dimensiones se creó un juego al cual se le dio su propia historia basada en las problemáticas que vivía la sociedad estadounidense en la década del 90’.

La Historia del juego está basada en la zona suroeste estadounidense, lugar dominado por las incesantes guerras de pandillas que buscan manejar los barrios de la ciudad y el tráfico de drogas de la zona. Ambientado en 1992, el juego cuenta la historia de Kevin c, quien decide volver a su ciudad de origen, tras 5 años de haberse establecido en otro estado. La trama cuenta como Kevin en su vuelta a su lugar natal decide tomar el mando de la pandilla apodada Bloods y empieza una rivalidad contra Los Crips en la lucha por el control del tráfico de drogas Los Ángeles.

Kevin c realiza enfrentamientos contra otros pandilleros para evitar la derrota de su pandilla, ganar el respeto de sus rivales y seguir controlando su ciudad natal.

**Manual de uso:**

Es un juego multijugador con un máximo de dos jugadores que compiten al mismo tiempo entre ellos para ver quien se quedará con la victoria.

En la parte superior de la pantalla se encuentra un contador que se inicia en 90 segundos y va disminuyendo en uno hasta llegar a cero, los jugadores deben derrotar a su adversario antes que ese contador finalice y para lograrlo deben destruir todas las vidas que se encuentran situadas a las espaldas de los personajes. Si el contador llega a cero y ningún jugador logra vencer, la partida finalizará y ganará el jugador que tenga más vidas al momento de que el tiempo finalice. En caso de que ambos jugadores tengan la misma cantidad de vidas, el juego terminará en empate. Si durante la partida hay un periodo de tiempo en el cual la pelota no toca ninguna de las paredes verticales, la velocidad se irá incrementando infinitamente agregándole una mayor dificultad al juego.

Cada jugador que logre explotar todas las vidas del otro participante antes que se agote el tiempo ganará una victoria en el ranking general, visible desde el menú del juego. Aquel que tanga la mayor cantidad de victorias será el dominador del juego hasta que alguien logre superarlo.

Dentro del juego:

Escape: Vuelve a la pantalla principal del juego.

Movimiento del jugador 1:

W: desplazamiento hacia arriba del personaje.

S: desplazamiento hacia abajo del personaje.

A: desplazamiento hacia la izquierda del personaje.

D: desplazamiento hacia la derecha del personaje.

Movimiento del jugador 2:

* : desplazamiento hacia arriba del personaje.
* : desplazamiento hacia abajo del personaje.
* : desplazamiento hacia la izquierda del personaje.
* : desplazamiento hacia la derecha del personaje.

**Recursos Humanos:**

**Minutas de las reuniones:**

Jueves 12/8:

Nos juntamos el día jueves para averiguar sobre múltiples juegos simples de programar, votamos y discutimos pero no hubo acuerdo. Se seleccionaron tres posibles juegos los cuales son: Snake, el Pong, y el Gorilla.bas.

Viernes 18/8:

Nos juntamos el día viernes de 9:10 a 9:50 a hablar sobre las dificultades de hacer tal juego o tal otro y si es posible hacerlo en el tiempo que nos queda de cursada y con las habilidades de programación actuales que posee cada integrante del grupo. Nos pusimos de acuerdo en hacer un juego elegido por votación siempre y cuando lo podamos hacer sin que se complique demasiado su elaboración, caso contrario bajaremos nuestra prioridad a un juego de menor nivel. Hemos hablado sobre las distintas formas de programar el juego elegido y llegamos a una conclusión.

Viernes 25/8:

Este día nos pusimos de acuerdo en cómo utilizaremos el controlador de versiones Git, quien trata el proyecto y quienes van a ser los miembros. Además se desarrolló el Kanban y se dividieron las tareas con las que va a empezar cada integrante del grupo.

Viernes 15/9:

Nadie hizo nada sobre el juego, todavía no empezamos a utilizar Git, y aunque ya tenemos nuestro repositorio, hoy empezamos a hacer un diagrama de clases entre el grupo para tener en cuenta que objetos y atributos tendremos en nuestro juego, cuantas imágenes importar y cuál será el ancho y alto de la pantalla.

Jueves 21/9:

Nos fijamos cual de nuestros integrantes sabe más de diseño gráfico y le preguntamos si estaba dispuesto a hacer el fondo del juego teniendo en cuanta varios aspectos claves del mismo. El juego está en fase de desarrollo. Se presentó un problema ya que según que dimensión de la pantalla hagamos, la imagen de fondo debe ser distinta, entonces tuvimos que fijar una resolución para hacer el fondo, pero dentro de la zona del fondo está la zona del juego con otras resoluciones, por lo que al no tener aún estas resoluciones estaríamos programando topes y márgenes del fondo que todavía no sabemos las medidas.

Viernes 6/10:

Nos juntamos el viernes para ver los avances de cada integrante con respecto al proyecto. El juego ya está muy avanzado. Al faltar poco para terminar decidimos agregarle algo nuevo, la música que se escucharía de fondo durante el juego. Se le encargo a un integrante del grupo que seleccione una lista de temas para luego someterla a votación y decidir cuál será el audio del juego.

Viernes 20/10:

Una vez ya elegida la música del juego se comenzó a discutir como programar las ultimas funcionalidades del juego, el menú y como se determinaría las puntuaciones de los jugadores.

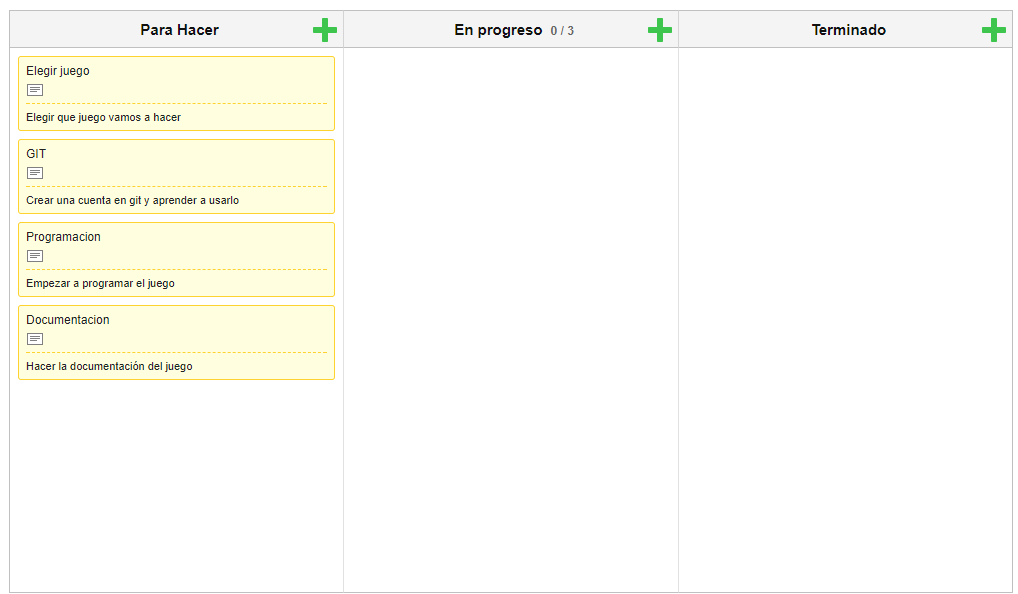
Viernes 27/10:

Se probó por última vez el juego y se llegó a la conclusión de que ya no hay más por hacer con respecto a la programación. Se comenzó a realizar la documentación.

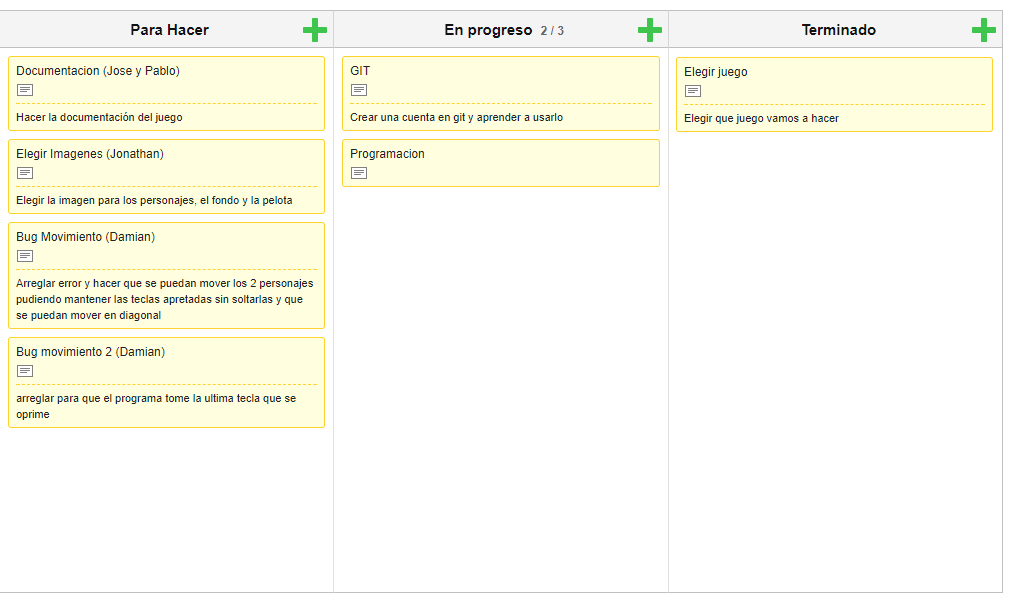
**Kanban:**

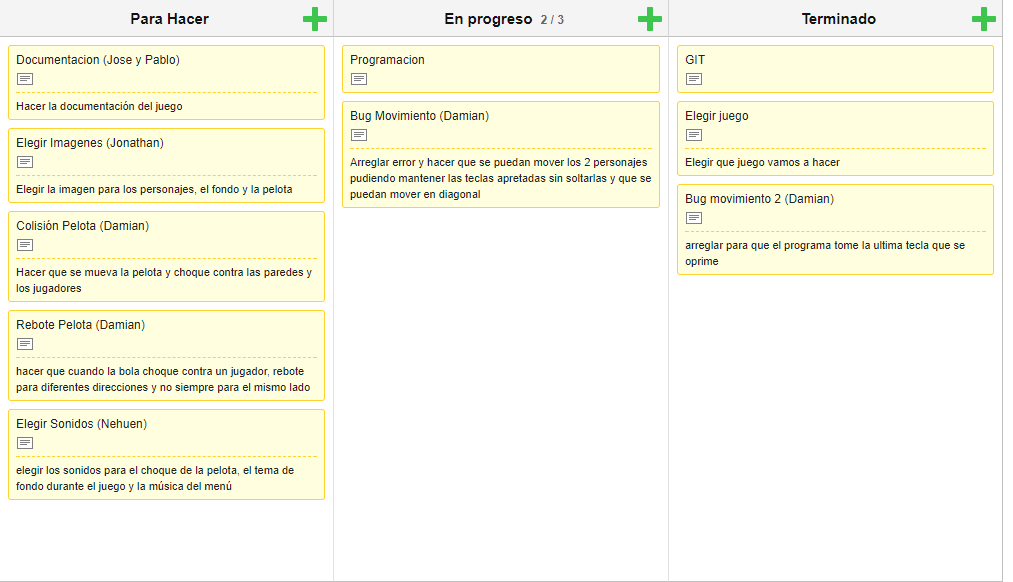
Para la organización del grupo durante la elaboración de este trabajo se eligió utilizar la metodología ágil kanban que consiste en crear un cuadro el cual se dividirá en las columnas que necesite el proyecto (siendo las más comunes To-do, haciendo y terminado). En la primera columna se colocan las tareas que hay que hacer y el integrante del grupo que la realizará y mientras se vaya avanzando en las tareas estas cambiaran de una columna a otra hasta que se finalicen.

Para empezar se crearon tres columnas y se asignaron las tareas principales para comenzar el proyecto:

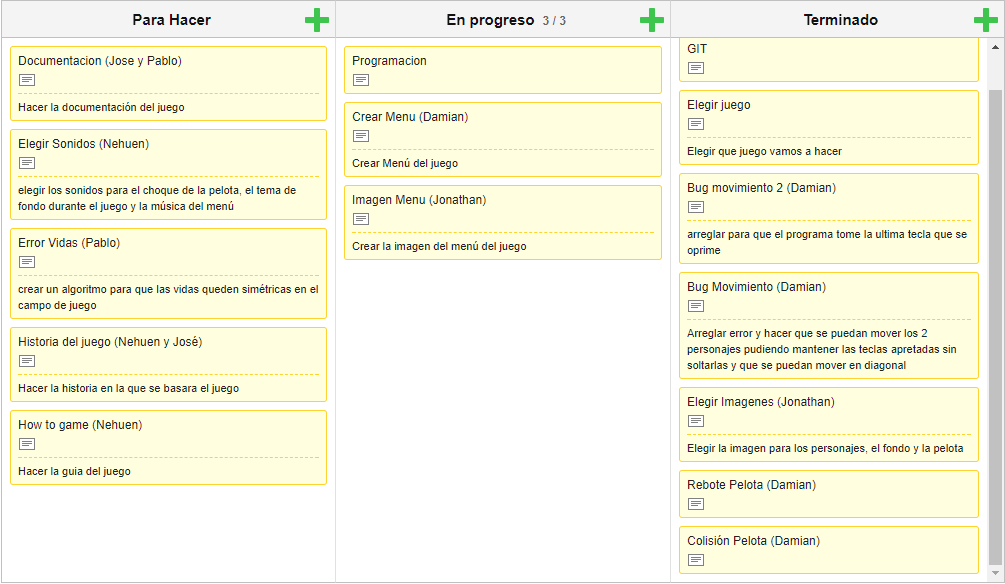


A medida que se iban realizando algunas tareas comenzaban a surgir nuevos problemas a resolver:

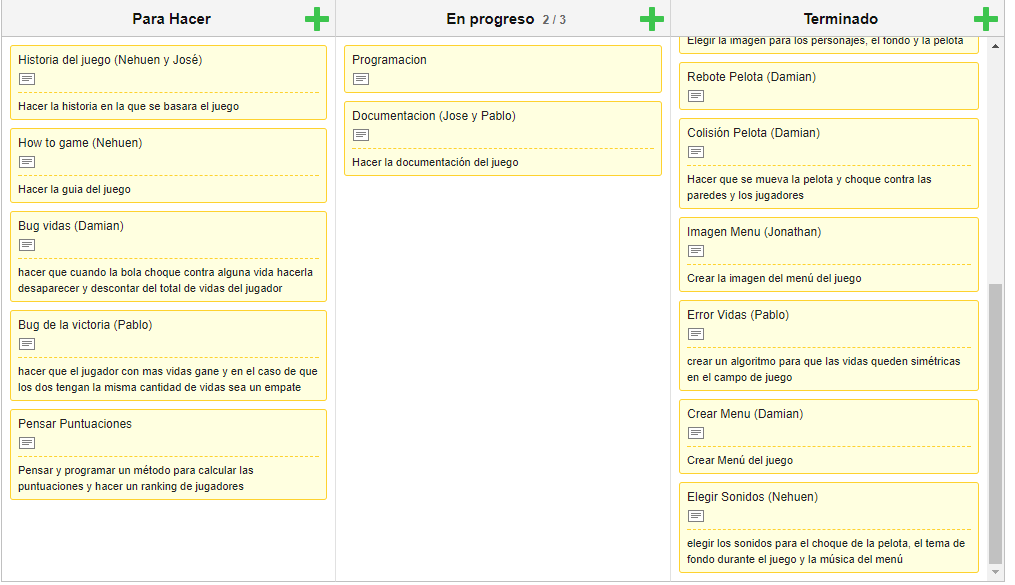


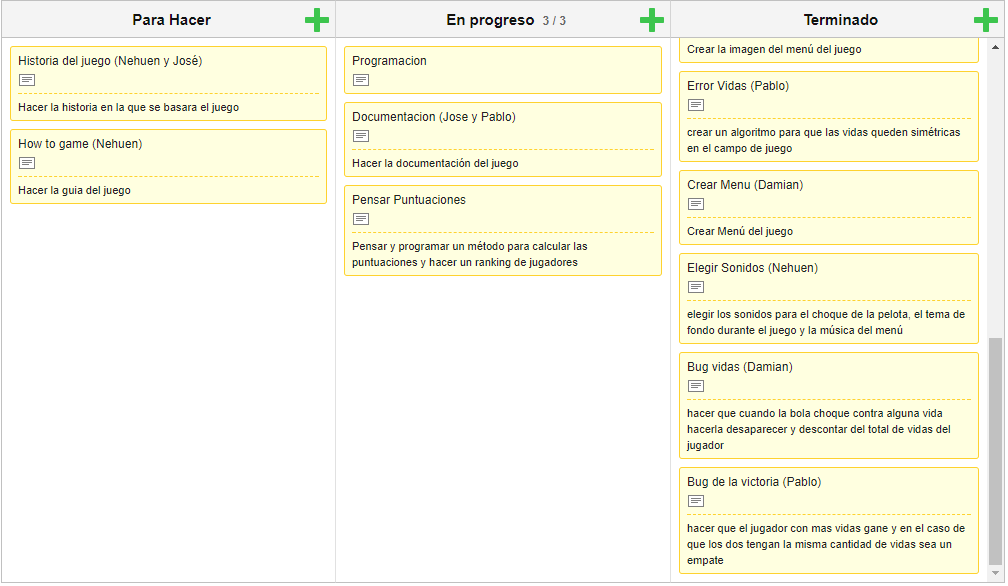


****

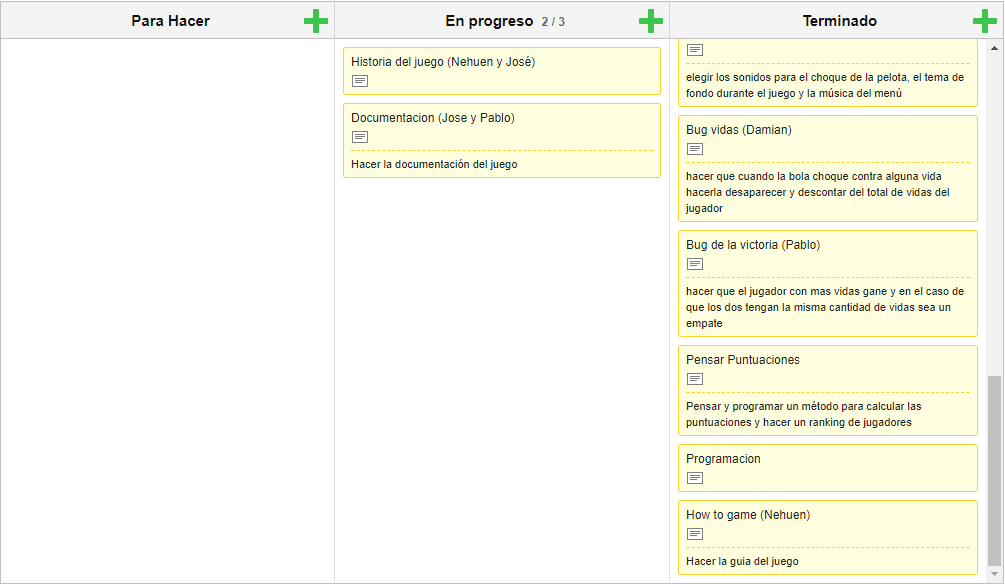
****

Las tareas se fueron resolviendo hasta terminar con el proyecto:





Una vez que ya no quedaban mas tareas “Para Hacer”, se terminaron las ultimas tareas y se dio por finalizado el proyecto:

****

**Futuras líneas de investigación:**

Para futuros proyectos que incluyan al juego programado en el presente trabajo se recomienda realizar lo siguiente:

Agregar las siguientes funcionalidades al juego:

* Hacer un opción de configuraciones para poder elegir:

+ Pantalla completa o no.

+ Resolución de la pantalla.

+ Teclas de movimiento del jugador.

* Poder elegir distintos tipos de personajes y fondos para el juego.
* Agregar poderes para aumentar la dificultad y mejorar la experiencia del juego.
* Agregar una IA al juego para poder jugar 1v1 y a partir de ahí agregar diferentes niveles y dificultades.

Reparar los siguientes problemas que presenta el juego:

* Solucionar que el juego a veces se traba (anda más lento) ejecutándolo en la misma pc o en computadoras diferentes por problemas con los fps.
* Arreglar el bug que provoca que si te quedas quieto en el mismo lugar la pelota siempre se moverá para los mismos lados.
* Arreglar el bug que si te quedas enfrente de las vidas el enemigo no podrá ganar.
* En multijugador hacer que cada vez que gane alguien la pelota valla para el lado contrario para que no siempre empiece la pelota con el mismo jugador ya que le daría ventaja.

**Conclusión:**

El grupo de trabajo pudo aprender a usar las nuevas herramientas de trabajo que facilitaron la realización del mismo. El uso de kanban fue muy productivo ayudándonos a ordenar el trabajo de manera adecuada lo cual ayudó a poder finalizarlo a tiempo. El controlador de versiones fue de mucha ayuda a la hora de trabajar con el código ya que nos permitió un mejor manejo del mismo.

En cuanto a la programación del juego, se pudo aprender un nuevo lenguaje de programación y se logró cumplir los objetivos que nos habíamos propuesto al iniciar con el proyecto. El juego no presenta grandes fallos pero en futuros trabajos se le podrían agregar más funcionalidades.

**Agradecimientos:**

Gracias a la empresa de videojuegos Rockstar Games por la inspiración brindada la hora de realizar la historia del juego.

Agradecemos a Martin Olmos, estudiante avanzado de licenciatura en sistemas en la Universidad Nacional de Lanús, por el apoyo moral y lo conocimientos sobre Python que nos brindó para la elaboración del juego.

Gracias a Google por facilitarnos la búsqueda de la información necesaria para la elaboración del proyecto.

Agradecemos a Toto, Ylvis, Imagine Dragons, System of Down, Slipknot y a la empresa Sony por proporcionarnos la música que se utilizó en el jugo.

Gracias a KanbanFlow por facilitarnos una herramienta para realizar kanban de una forma gratuita y más sencilla.

Gracias a Github por brindarnos un controlador de versiones gratuito que nos ayudó en la elaboración del proyecto.

**Referencias:**

Manejo de archivos en Python:

[*http://librosweb.es/libro/python/capitulo\_9/metodos\_del\_objeto\_file.html*](http://librosweb.es/libro/python/capitulo_9/metodos_del_objeto_file.html)

[*https://stackoverflow.com/questions/10140281/how-to-find-out-whether-a-file-is-at-its-eof*](https://stackoverflow.com/questions/10140281/how-to-find-out-whether-a-file-is-at-its-eof)

Manejo de listas y tuplas en Python:

[*http://recursospython.com/guias-y-manuales/listas-y-tuplas/*](http://recursospython.com/guias-y-manuales/listas-y-tuplas/)

Pygame curso:

[*https://www.youtube.com/playlist?list=PLpOqH6AE0tNherBf6bzGiDM1uIy\_E0WJH*](https://www.youtube.com/playlist?list=PLpOqH6AE0tNherBf6bzGiDM1uIy_E0WJH)

[*https://www.youtube.com/playlist?list=PLIQSt1Vybrne-bqSgbJ9KBD9kRSN58v-g*](https://www.youtube.com/playlist?list=PLIQSt1Vybrne-bqSgbJ9KBD9kRSN58v-g)

Python curso:

[*https://www.youtube.com/playlist?list=PLIQSt1VybrnfWMMFAQubhGBlz\_Z1nT7mB*](https://www.youtube.com/playlist?list=PLIQSt1VybrnfWMMFAQubhGBlz_Z1nT7mB)

Python y Pygame:

[*http://programarcadegames.com/index.php?chapter=introduction\_to\_sprites&lang=es*](http://programarcadegames.com/index.php?chapter=introduction_to_sprites&lang=es)

Descargar Python:

[*https://www.python.org/downloads/*](https://www.python.org/downloads/)

Descargar Pygame:

[*http://www.pygame.org/download.shtml*](http://www.pygame.org/download.shtml)

Herramienta de kanban:

[*https://kanbanflow.com/*](https://kanbanflow.com/board/842c9cebfa1af4222cddc1cf735fada3)

Controlador de versiones Git:

[*https://github.com/*](https://github.com/)