

Desarrollo de Pensamiento Computacional con Microsoft MakeCode Arcade

Guía 1 de Orientación General y Prácticas

Introducción a MakeCode Arcade

Combinar los elementos del juego, la programación, la tecnología y el impulso por el desarrollo del pensamiento computacional; es la tarea pedagógica de Microsoft MakeCode. Es por ello por lo que entre los recursos que impulsa MakeCode se encuentra Microsoft MakeCode Arcade.

MakeCode Arcade es un entorno para el desarrollo y creación de videojuegos del estilo Arcade mediante el lenguaje de programación por bloques. Estos juegos de tipo Arcade son aquellos juegos que se podían encontrar en los años 70's - 80's en las máquinas recreativas (Pac-Man, Asteroides, Centipies, Mario Bros...).

Los juegos se estructuran por niveles de logros, acumulación de puntos, retos. Los juegos constan de un diseño minimalista, y un personaje que tiene que 'superar' todas las pantallas con el objetivo de conseguir el mayor número de puntos posible.

Entonces, surge una pregunta: Si los videojuegos no cumplen con los principios del aprendizaje basado en juegos, ¿por qué existe un entorno para desarrollar videojuegos? ¿cuál es entonces la importancia de MakeCode Arcade?

La respuesta sigue siendo: impulsar el desarrollo del pensamiento computacional, y establecer ambientes de aprendizaje basados en juegos.

MakeCode Arcade es una herramienta que impulsa el aprendizaje en programación por bloques, cuyo objetivo es la creación de videojuegos independientes los cuales tienen la posibilidad de ser jugados en miniconsolas. En consecuencia, estimula al estudiante a temprana edad en su creatividad y promueve el interés por la programación en un entorno natural y en un ambiente de aprendizaje basado en juegos.

A continuación, veremos los elementos básicos que se deben considerar para el desarrollo de videojuegos y realizaremos nuestro primero videojuego.

Elementos para el desarrollo de videojuegos

En general existe varios elementos a tomar en consideración al momento de comenzar el diseño y desarrollo de un videojuego. Sin embargo, el *primer* paso es definir el título del videojuego. El *segundo* paso por considerar es el propósito del juego.

A continuación, presentamos una serie de elementos generales que se deben tener en cuenta:

Personaje	Poderes	Niveles	Escenarios	Normas
Se conoce como Sprite y se refiere a el personaje principal, enemigos, y gráficos especiales como comida, monedas, flores, etc.	Se establece poderes, super poderes, habilidades especiales y otras funciones para el personaje principal y los enemigos.	Se establecen los niveles del juego. El número de niveles está relacionado con el número de misión por lograr.	Se establecen los criterios de diseño para los diferentes escenarios: Mundos, temas, misiones, universos, etc.	Se establece: reglas generales de acumulación de puntos, las metas, bonificaciones, y las penalizaciones.

En todo diseño se desarrolla la versión de prueba, en la cual se hacen los ajustes correspondientes antes de que el videojuego entre en producción.

Estas etapas de prueba y error son esenciales durante el proceso de desarrollo de videojuegos, y una etapa importante de aprendizaje para el estudiante.

Microsoft Makecode nos permite iniciar el desarrollo de videojuegos desde el nivel de principiante. Ofrece a los educadores y estudiantes una serie de recursos como lecciones de aprendizaje y tutoriales que facilitan la introducción a MakeCode Arcade.

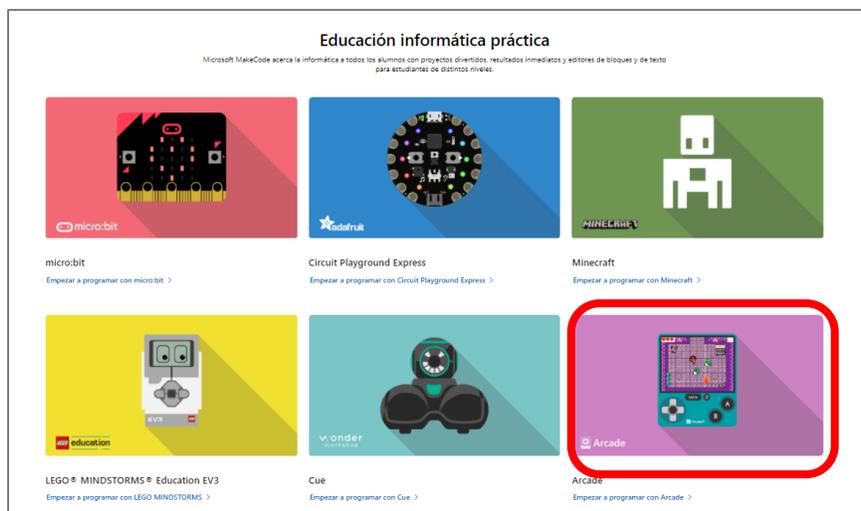
Activando MakeCode Arcade.

Ahora iniciemos con los primeros pasos para el desarrollo de juegos de video y activemos el simulador Arcade.

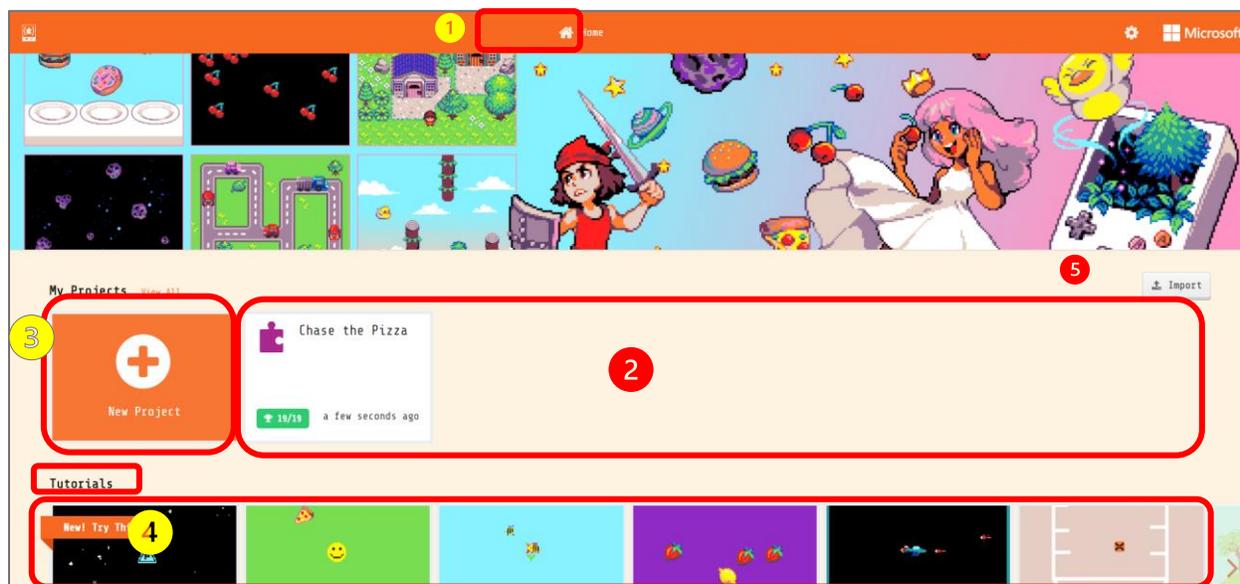
Idiomas disponibles. El simulador de Arcade ofrece la posibilidad de trabajar con 5 idiomas por el momento: inglés, alemán, francés, chino y japones (información actualizada a diciembre 2020).

Por otro lado, también puede utilizar la función de "*traducción*" (o *translate*) del navegador Edge. Sin embargo, los siguientes pasos e ilustraciones las seguiremos en idioma inglés, y se incluirán explicaciones complementarias.

Paso 1. Ingresa a Microsoft MakeCode [aquí](#). Ubique y seleccione el simulador Arcade. Activa el simulador.



Paso 2. Explorando la ventana principal del Simulador. Observa con atención las diferentes secciones que el simulador ofrece.



Entre estas secciones podemos identificar las siguientes

- (1) Botón *Home* o página inicial
- (2) Lista o carrusel de proyectos realizados
- (3) Botón para activar el editor de programación e iniciar un nuevo proyecto o abrir un proyecto.
- (4) Biblioteca de Recursos. Inicia con el acceso a los tutoriales.
- (5) Botón para importar proyectos.
- (6) Flechas de desplazamiento lateral de proyectos.

Desarrollando Videojuegos. Realizaremos un videojuego y durante el proceso de programación destacaremos sus diferentes etapas. Los detalles de diseño del videojuego que vamos a considerar son los siguientes:

Título: Invasión de Asteroides

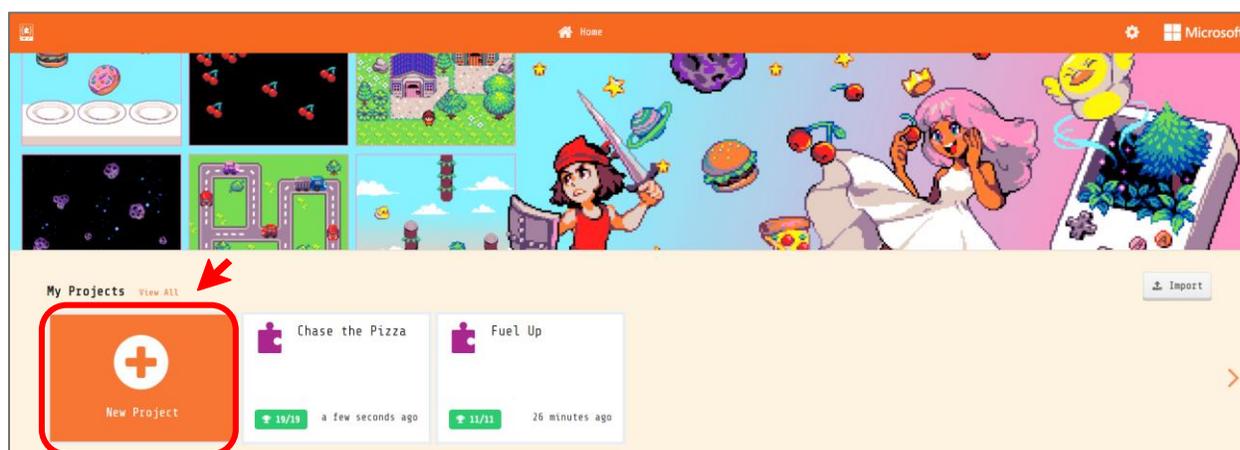
Objetivo: Proteger de los asteroides los planetas

Personaje	Poderes	Niveles	Escenarios	Normas
Héroe: Nave Espacial Enemigos: Asteroides.	Proyectil de nave espacial Bola de recarga de energía	Un solo nivel de juego	El Universo	Evitar ser destruido por asteroides. Máximo de oportunidades: 3

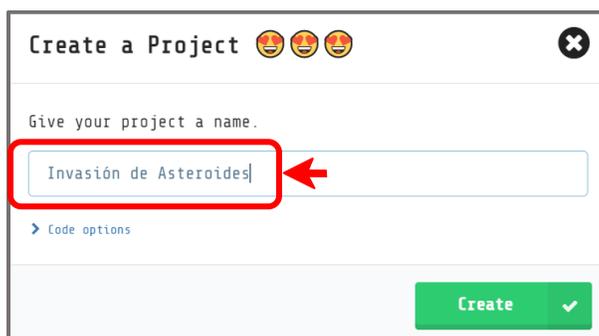
Área académica: Ciencias, Programación

Temas: El Espacio, los Asteroides, La Tecnología y Misiones Espaciales.

Paso 3. Iniciar un nuevo proyecto. Localice el botón "Nuevo proyecto", active el botón para iniciar el nuevo proyecto.



En la ventana emergente, escriba el nombre sugerido para su primer proyecto: "Invasión de Asteroides". Y presione el botón "Create".



Ahora se encuentra en el editor de programa por bloque de Arcade.

Como podrá observar los editores para el desarrollo de programas por bloques son similares a los anteriores y podrá identificar similares elementos en la interfaz gráfica de simulación que ofrece MakeCode Arcade.

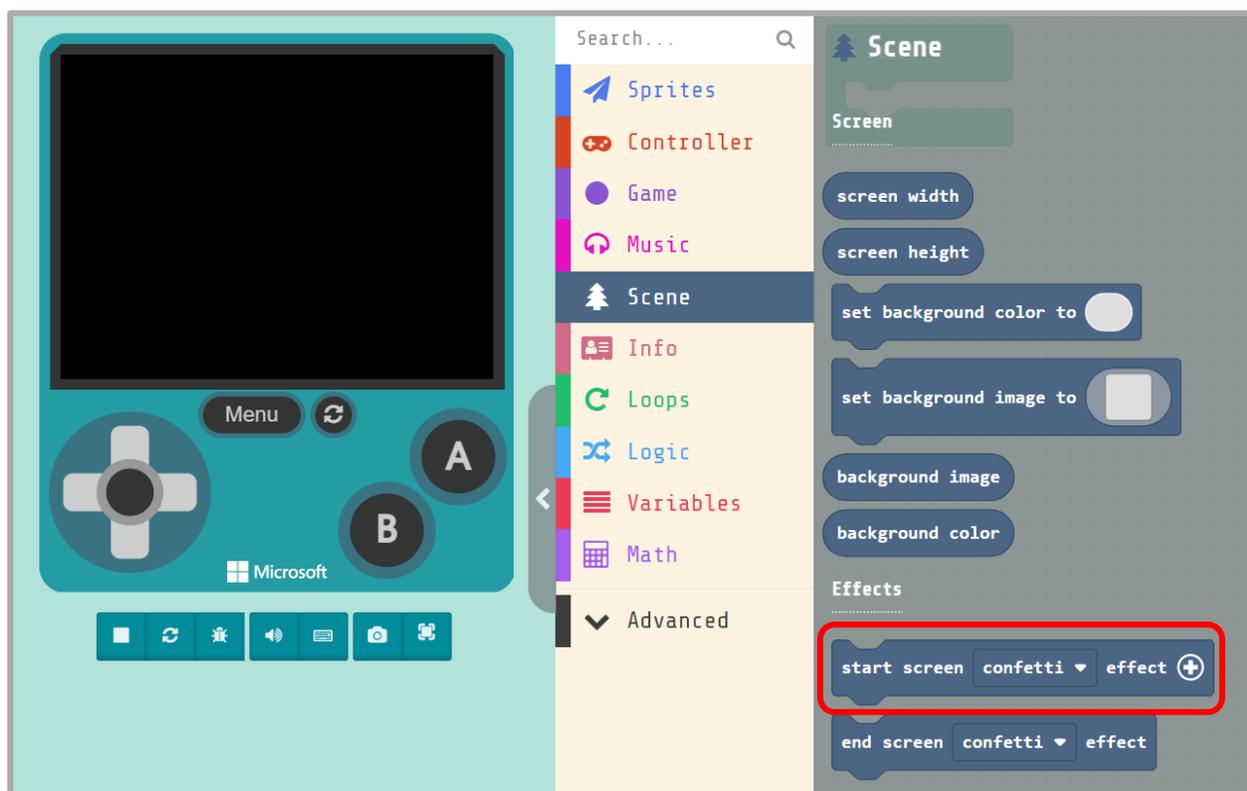
También podrá identificar que el simulador consiste en una consola de juegos de tipo Arcade. La cual contiene la palanca de movimientos, los botones A y B, la opción de MENU, y la función para reiniciar el juego.



NOTA: La estandarización de la interfaz gráfica para la conexión de bloques y eventos de los editores de programas de bloques ha contribuido en facilitar el aprendizaje de múltiples plataformas y proyectos. También, la estandarización permite convertir un programa de bloques en un programa de lenguaje estructurado como JavaScript o Python.

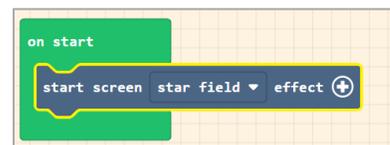
ESCENARIO: El Espacio.

Paso 4. Seleccione del inventario de bloques, la categoría **SCENE (o ESCENA)**. Ubique el bloque "start screen - effect". Seleccione el bloque y arrástrelo a la ventana de edición de programa hasta conectarlo con el bloque "On start". Observe la siguiente pantalla.



El bloque *"start screen - effect"* nos permitirá programar la pantalla inicial del juego y agregar efectos complementarios.

Paso 5. Utilizando el puntero del ratón (mouse), ubíquese en la casilla *"confetti"*, haciendo clic sobre la flecha se despliega el menú de opciones de pantalla. Seleccione la opción ***"star field"***. Observe los cambios en la pantalla del simulador.



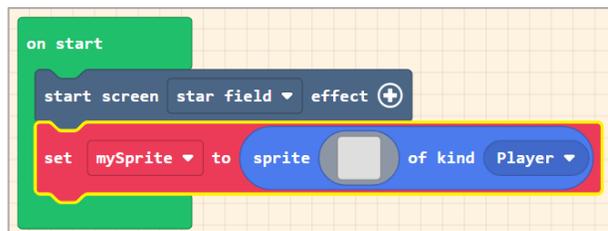
PERSONAJE: Nave espacial

Los personajes del juego, obstáculos, o elementos especiales se programan utilizando la categoría de bloques **SPRITES**.

SPRITE, es una terminología que no tiene traducción, pero hace referencia a una imagen o gráfico de computadora que tiene desplazamiento en la pantalla determinado por una acción específica.

Por defecto el simulador Arcade denomina al primer personaje como ***mySprite***. Sin embargo, nos permite renombrar cada personaje o elemento, y al realizar el cambio de nombre automáticamente genera un bloque del tipo variable con el nombre de nuestro personaje. A continuación, demostraremos esta característica del simulador.

Paso 6. Seleccione nuevamente en el inventario de bloques la categoría **"Sprite" (bloques de color azul)**. Localice el bloque *"set mySprite to Sprite – of kinf player"*.



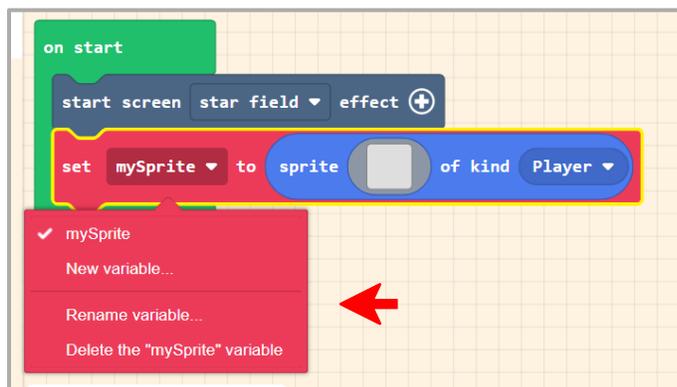
Seleccione el bloque y arrástrelo a la ventana de edición de programa hasta conectarlo con el bloque *"start screen - effect"*

El bloque seleccionado permite configurar dos parámetros del personaje denominado mySprite: su apariencia mediante el editor de imágenes (cuadro gris) y el tipo de participación o role que tendrá en el juego (por defecto su role es el de "Player" – jugador).

Paso 7. Nombraremos a nuestro personaje como ***"nave"***.

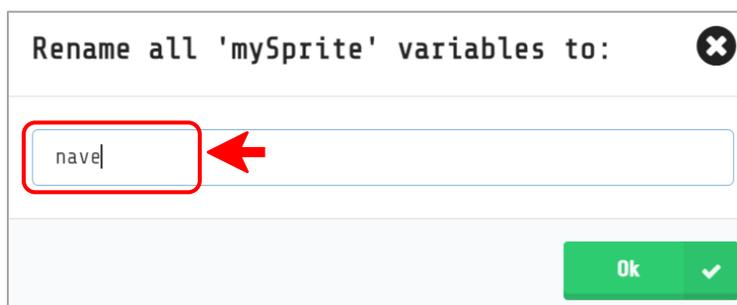
Para ello ubíquese en la casilla *"mySprite"* del bloque *"set mySprite to Sprite – of kinf player"*.

Con un clic despliegue el menú emergente y seleccione la opción ***"Rename variable..."***



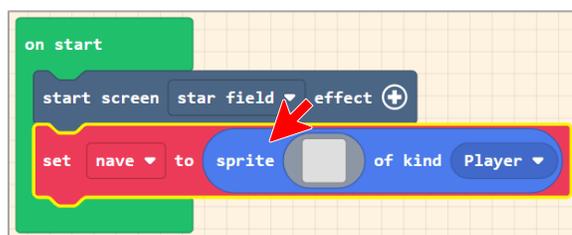
Este procedimiento nos permite trabajar cada personaje como una variable.

Paso 8. En la ventana emergente coloque el nombre de la nueva variable: "nave". Observe el resultado.



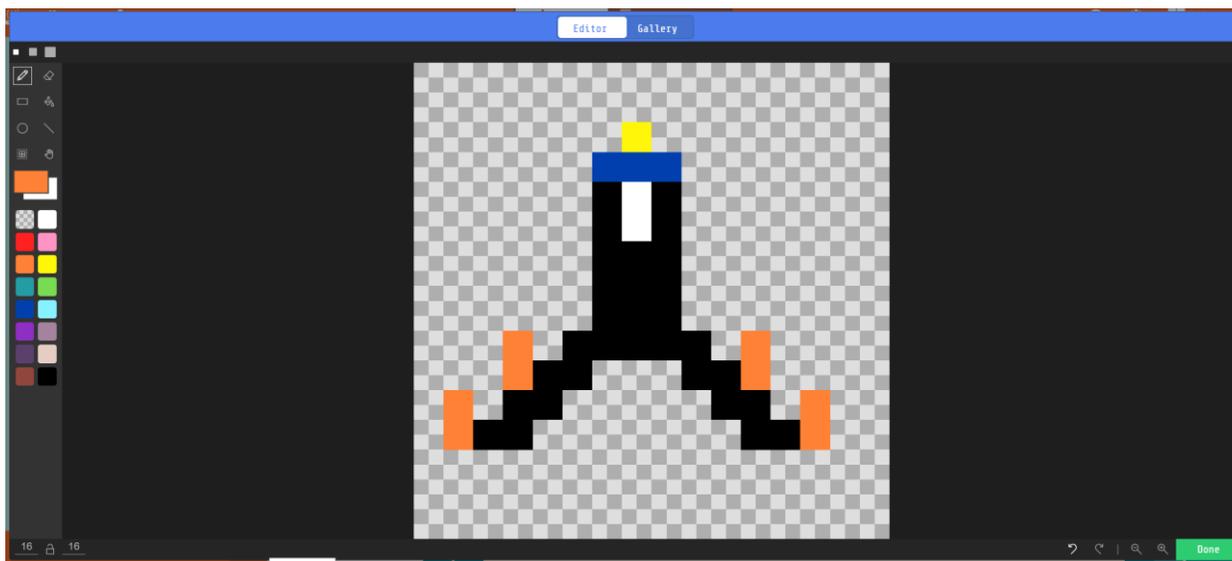
Paso 9. Brindaremos una imagen a nuestro personaje.

Para ello ubíquese en la casilla del **cuadro gris** del bloque "set mySprite to Sprite – of kind player" y haga clic sobre el cuadro gris para activar el editor y la galería de imágenes.

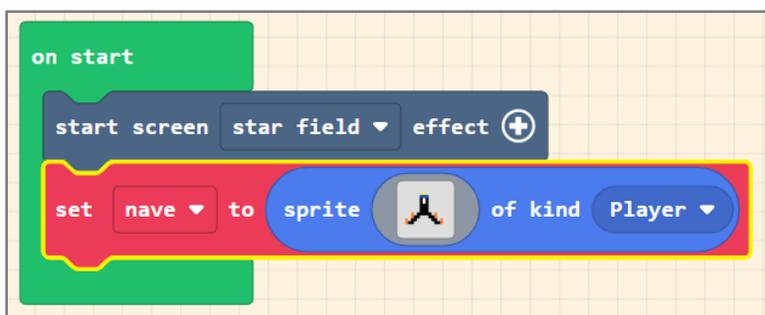


Editor y Galería de Imágenes.

A continuación mostramos la pantalla del editor de imágenes. Hemos sugerido un diseño de nave espacial para el personaje principal. Sin embargo puede seleccionar la galería de imágenes y seleccionar otro tipo de imagen.



Una vez finalizada la edición de la imagen o seleccionada la imagen de la galería, haga clic en el botón "DONE". Verifique en la siguiente figura el resultado.



Tipo de participación del personaje o role del personaje en el juego.

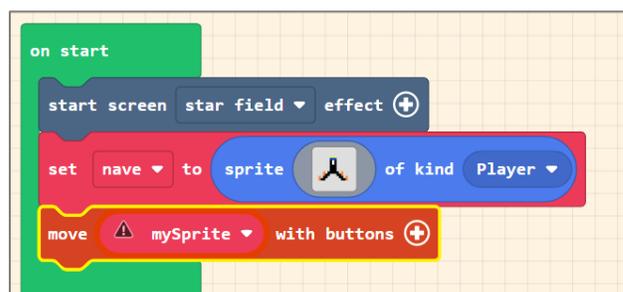
En el caso del personaje "nave" su tipo de participación será la de un jugador. por lo cual el parámetro "Player" del bloque "set mySprite to Sprite – of kind player", no se modifica.

Ahora daremos movimiento al personaje.

CONTROL DE MOVIMIENTO

Paso 10. Seleccione nuevamente en el inventario de bloques la categoría "CONTROLLER" (bloques de color rojo). Localice el bloque "move mySprite with buttons".

Seleccione el bloque y arrástrelo a la ventana de edición de programa hasta conectarlo con el bloque "set nave to Sprite – of kind player".



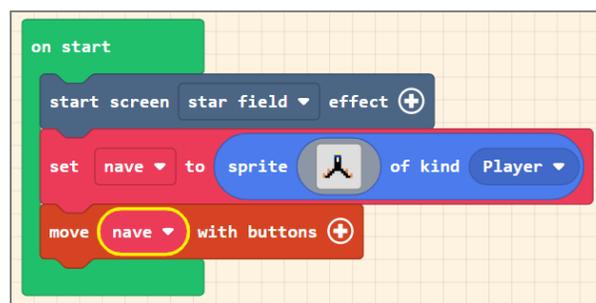
Observe la advertencia de inconsistencia de variable que se genera al conectar el bloque. La razón de esta advertencia es porque debemos colocar la variable correcta.

El edito nos ayuda a que recordemos este paso obligatorio para que la secuencia de programación siga siendo la correcta.

Paso 11. Utilizando el puntero del ratón (mouse), ubíquese en la casilla "mySprite", haciendo clic sobre la flecha se despliega el menú de opciones de pantalla.

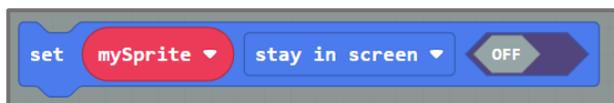
Seleccione la opción "nave". Active el programa y observe sus resultados.

La nave espacial ahora puede ser controlada por medio de la palanca de desplazamiento.



¿La nave espacial se mantiene en la pantalla de la consola de juegos o eventualmente se pierde por los bordes y ya no se observa?

Paso 12. Para evitar que la nave espacial se desplace fuera de la pantalla de juegos de la consola colocaremos límites.

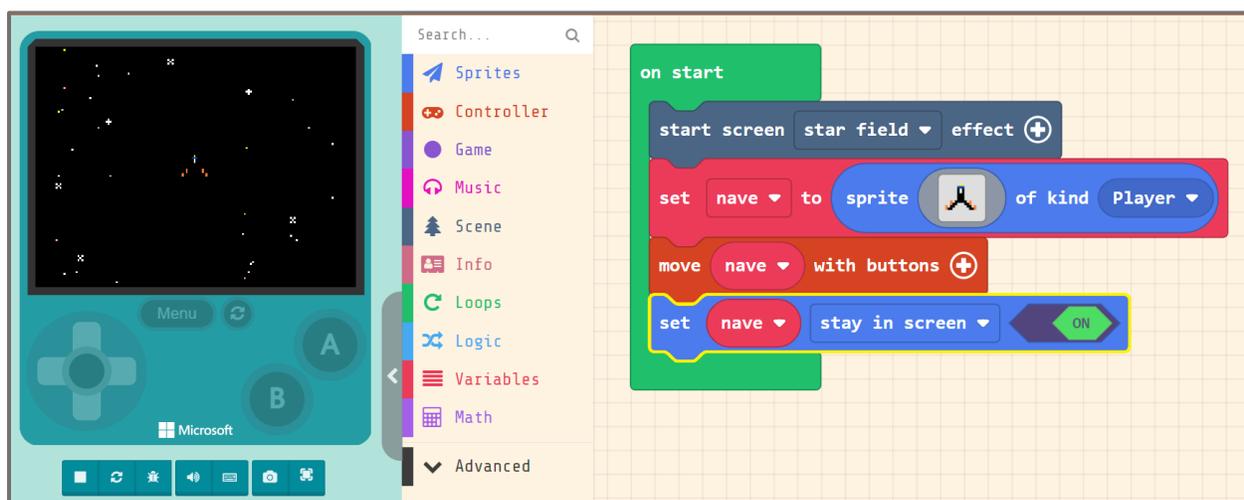


Para ello seleccione nuevamente en el inventario de bloques la categoría **"SPRITE"**. Localice el bloque *"set mySprite stay in screen - off"*.

Seleccione el bloque y arrástrelo a la ventana de edición de programa hasta conectarlo con el bloque *"move nave with buttons"*.

Cambien la posición del interruptor del bloque *set mySprite stay in screen - off* de su posición de partida "off" a la posición **"on"**.

Observe nuevamente la advertencia de inconsistencia de variables que se genera al conectar el bloque. Debe seleccionar la variable **"nave"** nuevamente. Verifique los resultados en la siguiente figura.



La primera etapa del juego esta completa. Ahora se ha desarrollado la pantalla principal del juego, un nivel de dificultad, un personaje y sus movimientos.

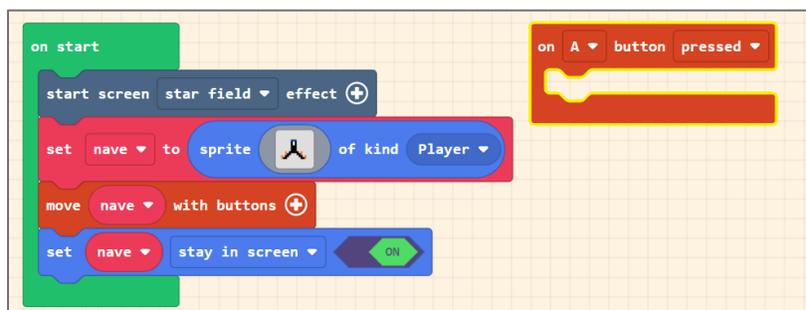
Ahora introduciremos a los enemigos del juego y brindaremos poderes de defensa a la nave espacial.

PROGRAMANDO PODERES.

Equiparemos la nave espacial con proyectiles para que pueda defenderse de los asteroides durante su recorrido.

Paso 13. Los proyectiles serán lanzados utilizando el botón A. Seleccione nuevamente en el inventario de bloques la categoría **“CONTROLLER”**.

Localice el bloque *“on A button pressed”*. Seleccione el bloque y arrástrelo a la ventana de edición de programa.



Paso 14. Seleccione nuevamente en el inventario de bloques la categoría **“SPRITE”**.

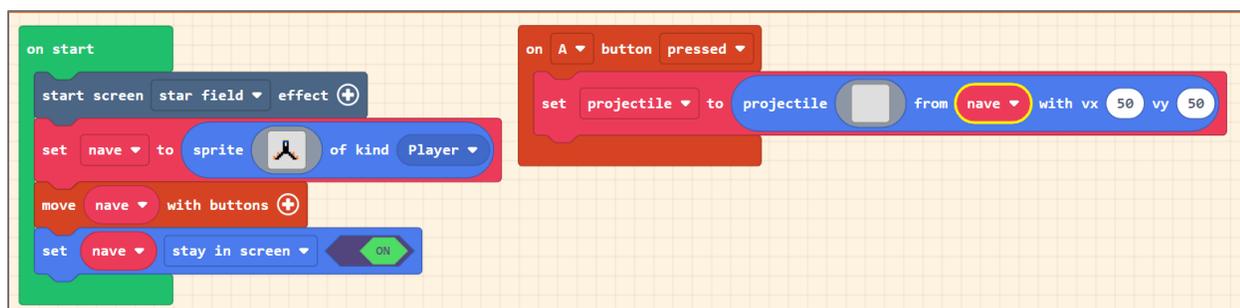
Localice el bloque *“set projectile to / projectile – from mySprite – with vx 50 vy 50”*.



Seleccione el bloque y conéctelo con el bloque *“on A button pressed”*.

Tomen en cuenta nuevamente la advertencia de inconsistencia de variables. Realice el cambio la variable **mySprite** por **“nave”** nuevamente.

Verifique los resultados en la siguiente figura.



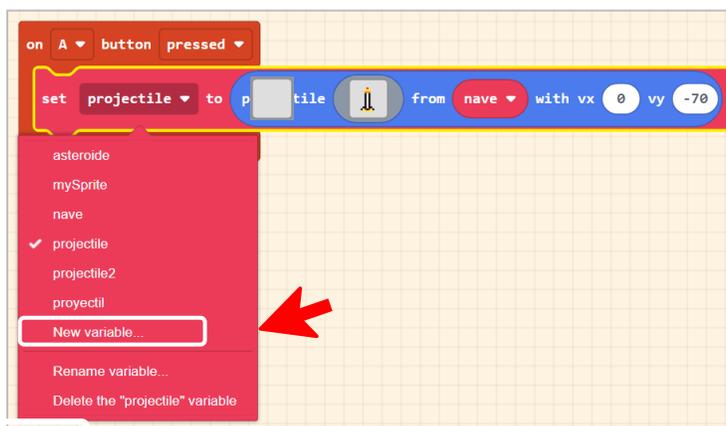
El bloque anterior permite asociar al personaje **nave**, un nuevo personaje que será el proyectil.

Ahora es necesario realizar el nombramiento de una nueva variable con el nombre de **“proyectil”**. Para ello seguiremos el procedimiento aprendido anteriormente.

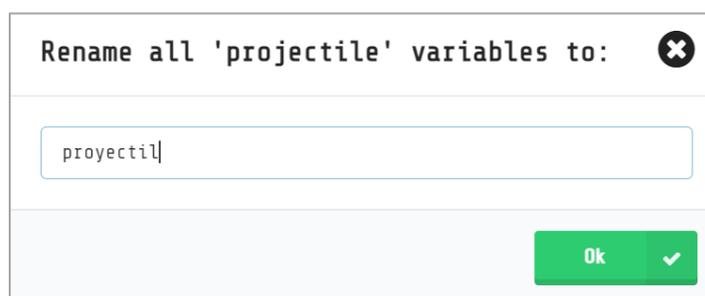
Paso 15. Nombraremos a nuestro personaje utilizando la variable que llamaremos "**proyectil**".

Para ello ubíquese en la casilla "**porjectile**" del bloque "**set porjectile to projectile / from nave with vx 0 vy -70**".

Con un clic despliegue el menú emergente y seleccione la opción "**Rename variable...**"



En la ventana emergente coloque el nombre de la nueva variable: "**proyectil**". Observe el resultado.

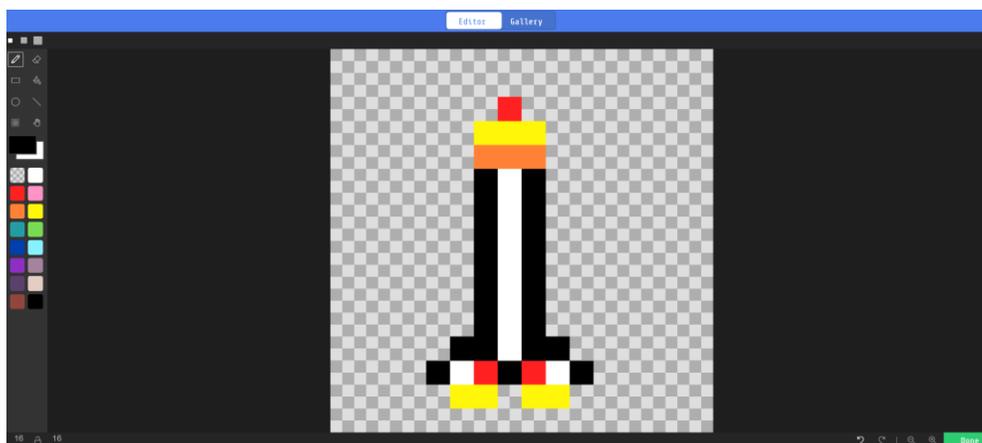


Este procedimiento anterior nos permite trabajar cada personaje como una variable. Esta metodología es importante seguirla ya que es parte de la planificación del diseño del juego.

Ahora, debemos ahora brindar la imagen a este nuevo personaje y definir la su posición de salida.

Paso 16. Para ello ubíquese con el cursos del ratón (mouse) en la casilla del **cuadro gris** del bloque "**set **proyectil** to / projectile – from nave – with vx 50 vy 50**" y haga clic sobre el cuadro gris para activar el editor y la galería de imágenes.

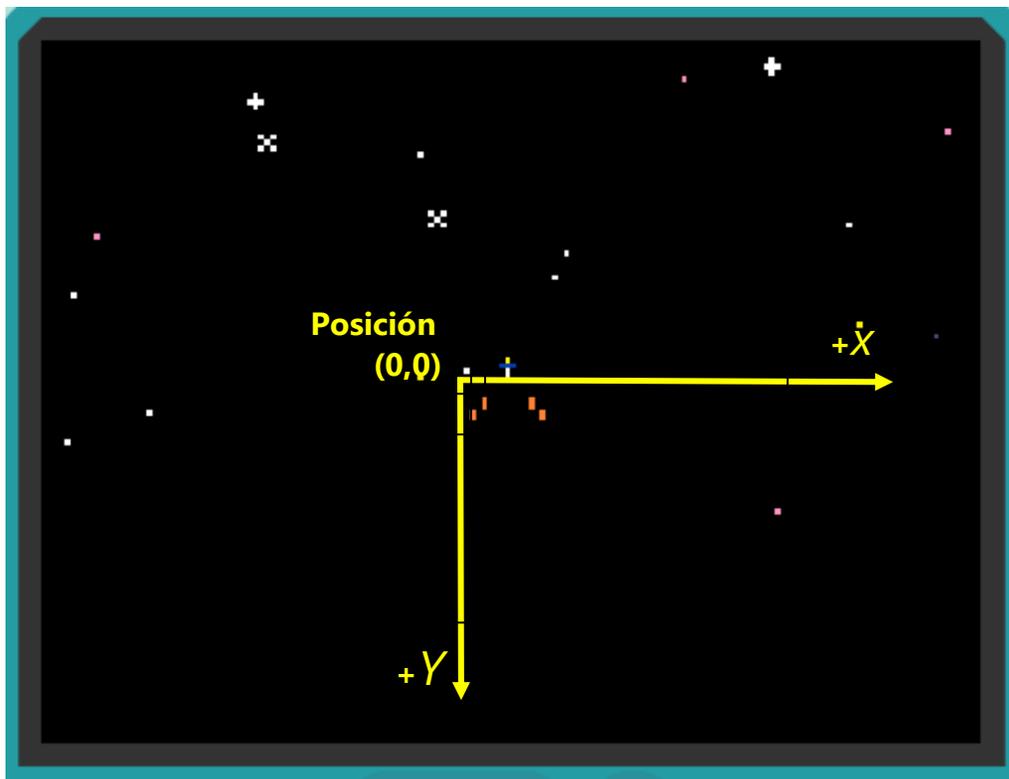
Nuevamente hemos sugerido un diseño para el poryectil. Sin embargo puede activar la galeria de imágenes y seleccionar otro tipo de imagen.



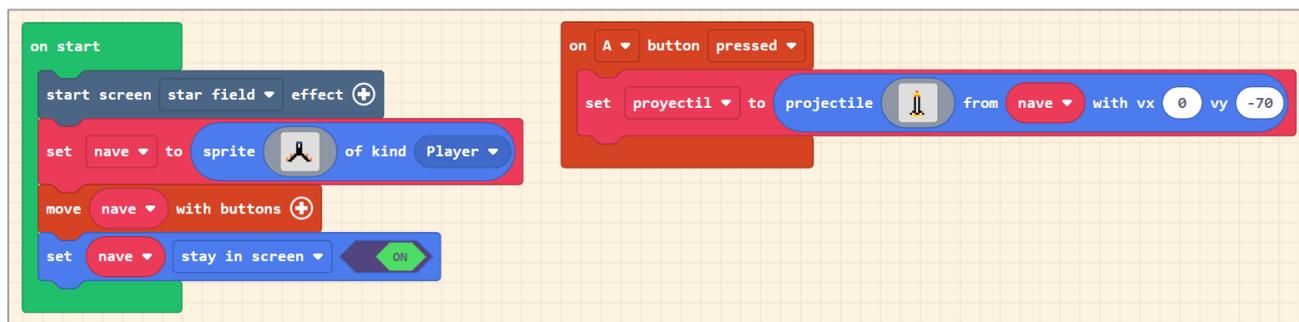
Paso 17. Cambie los valores de las coordenadas de "x" y de "y" indicadas en el bloque "set proyectil to / projectile – from nave – with vx 50 vy 50", por los valores: "**X = 0**" y "**Y = -70**".

NOTA:

COORDENADAS XY ARCADE. En el simulador Arcade, los ejes XY tiene su punto de origen en el personaje principal, es decir, la posición (0,0) siempre se encuentra en la nave espacial. Esta posición se desplaza su la nave espacial se desplaza. La siguiente imagen muestra la disposición de ejes positivos XY (en consecuencia, negativos).



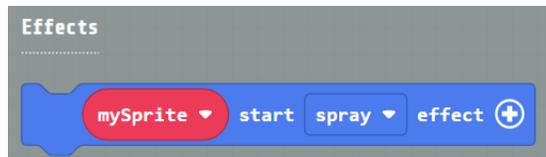
Verifique los resultados en la siguiente figura.



Ahora debemos agregar el efecto de disparo o lanzamiento del proyectil.

Paso 18. Seleccione nuevamente en el inventario bloques la categoría “**SPRITE**”.

Localice el bloque “**mySprite start spray effect**”.



de

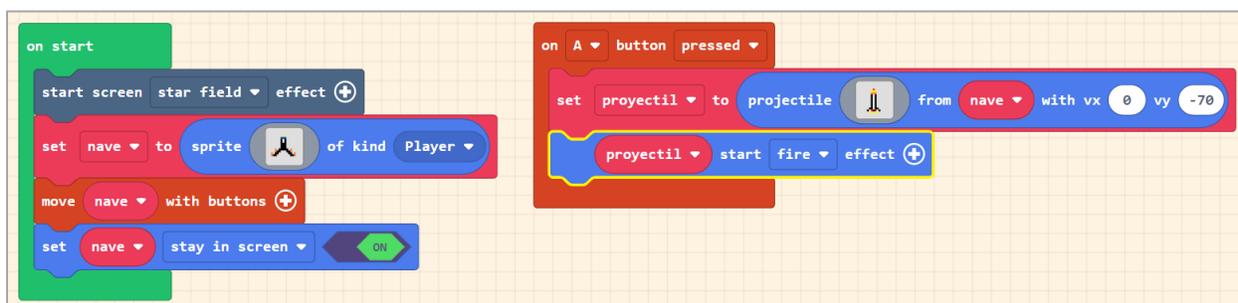
Seleccione el bloque y conéctelo con el bloque “*set proyectil to / projectile – from nave – with vx 0 vy -70*”.

Nuevamente aparece la advertencia de inconsistencia de variables. Pero en esta ocasión el personaje es el proyectil. Realice el cambio la variable por “**proyectil**” para que el efecto a seleccionar se aplique al proyectil.

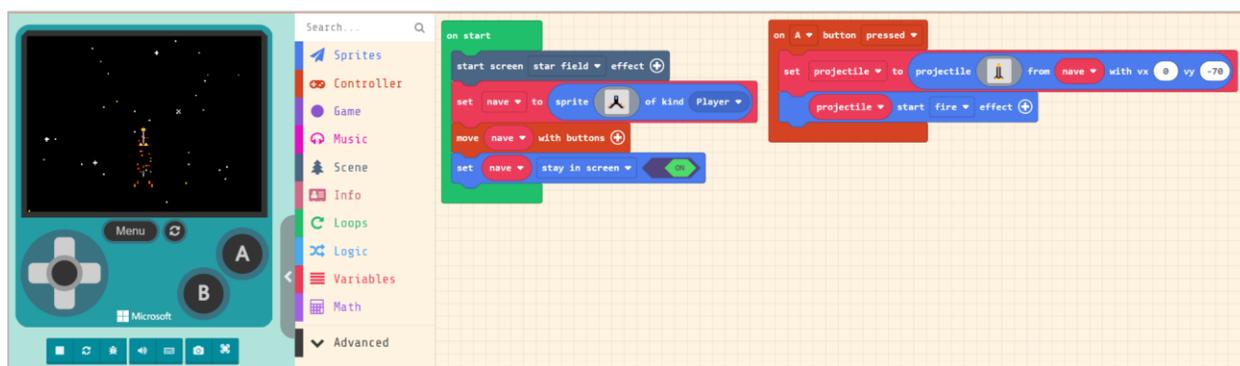
Paso 19. Utilizando el puntero del ratón (mouse) ubíquese en la casilla “*spray*”, haciendo clic sobre la flecha se despliega el menú de opciones de efecto para los personajes (Sprite).

Seleccione la opción de efecto “**fire**”. Active el programa, verifique la disposición de bloques, y observe sus resultados.

Disposición de bloques:



Programa:



Puede observarse en la consola que al presionar el botón A se dispara el proyectil.

REFLEXION.

El juego se encuentra aún en su etapa de desarrollo. La nave espacial (personaje principal), ya puede ser dirigida a lo largo de la pantalla de la consola de juegos mediante el control de dirección. También la nave puede ahora lanzar proyectiles haciendo uso del botón A.

Ambas funciones se pueden observar claramente en la programación de bloques, una sección corresponde al bloque "on start", el cual permite que al iniciar el juego se cargue el escenario y aparezca la nave espacial. Por otro lado, se ha programado la función del botón A, que es brindar a la nave espacial la posibilidad de disparar proyectiles.

Nuevamente podemos observar que hemos programado dos funciones independientes u objetos, lo cual nos permite un escenario interactivo para el usuario final.

ENEMIGOS: Asteroides

Los asteroides son enemigos que deben aparecer en forma repetitiva cuando el juego inicie. Para ello debemos utilizar un bloque que permita actualizar la información del juego cada cierto periodo de tiempo.

Paso 20. Seleccione el inventario de bloques la categoría "**GAME**". Localice el bloque para actualizar el juego "**on game update every 500 ms**".

Seleccione el bloque y arrástrelo a la ventana de edición de programa.



Recuerde que 1 segundo es equivalente a 1,000 milisegundos.

Paso 21. Modifique el valor **500** ms por **1000** ms

El bloque seleccionado permitirá actualizar o ejecutar el código o programa en su interior cada segundo.

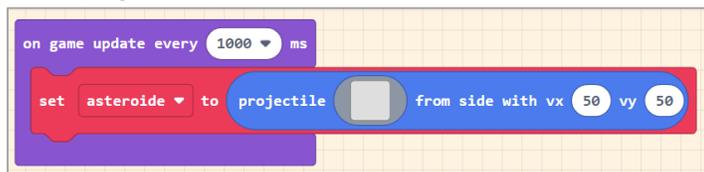
El ciclo de actualización de información cada segundo nos permitirá programar **la aparición de un solo asteroide cada segundo**, logrando con ello crear una lluvia de asteroides.

Paso 22. Seleccione el inventario de bloques la categoría "**SPRITE**". Localice el bloque "**set projectile to projectile – from side with vx 50 vy 50**".

Seleccione el bloque y conéctelo con el bloque "**on game update every 1000 ms**".

Nuevamente aparece la advertencia de inconsistencia de variables. Pero en esta ocasión el personaje es el **asteroide**. Realice el cambio la variable *projectile* por "**asteroide**".

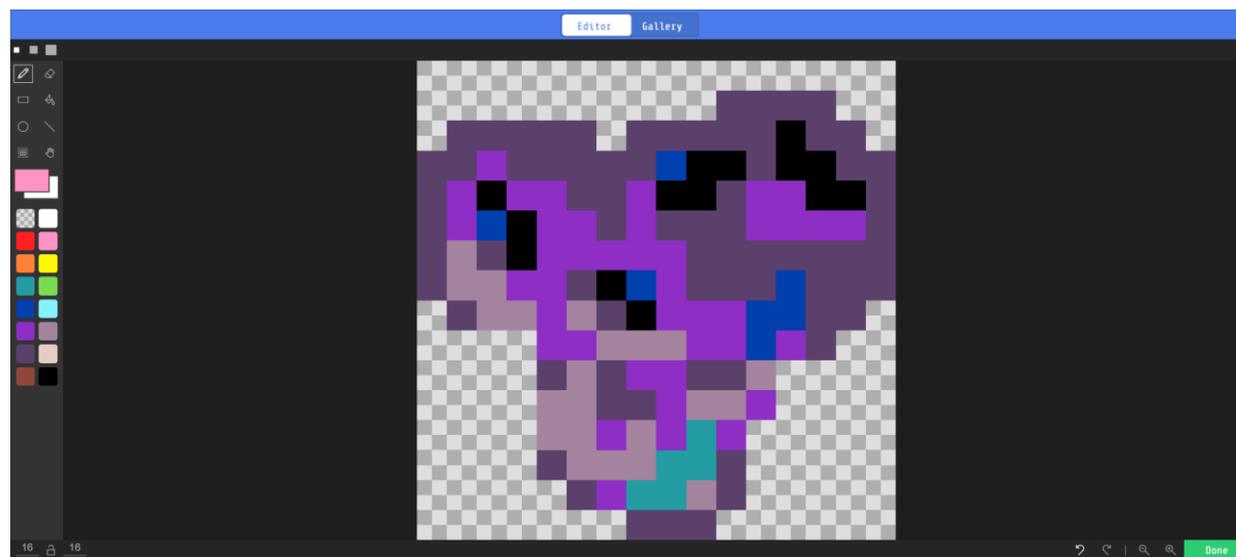
El bloque permite asociar al personaje asteroide la función de proyectil Arcade.



Es es la misma función que aplicamos a nuestro proyectil asociado a la nave. También, el bloque nos permite estable al mismo tiempo el punto de inicio o aparición del personaje (asteroide).

Paso 23. Brindaremos una imagen al personaje asteroide. Para ello ubíquese con el cursor en la casilla del **cuadro gris** del bloque "set **asteroide** to / *projectile* – *from side* vx 50 vy 50" y haga clic sobre el cuadro gris para activar el editor y la galería de imágenes.

Ahora seleccione un asteroide de la galería de imágenes.



Nuevamente hemos sugerido un diseño para el personaje asteroide. Sin embargo puede activar la galería de imágenes y seleccionar otro tipo de imagen.

Paso 24. Ahora fijaremos una ubicación de inicio para el asteroide, en la coordenada $X = 0$; $Y = 50$. Con esta coordenada el asteroide aparece inicialmente desde la esquina superior derecha cada segundo.

La dirección positiva de Y, hace que el desplazamiento de cada aparición sea hacia de arriba debajo de la pantalla.

IMPORTANTE: Se debe destacar que el bloque "set asteroide to projectile – *from side* with vx 0 vy 50"; tiene la instrucción "**from side**".

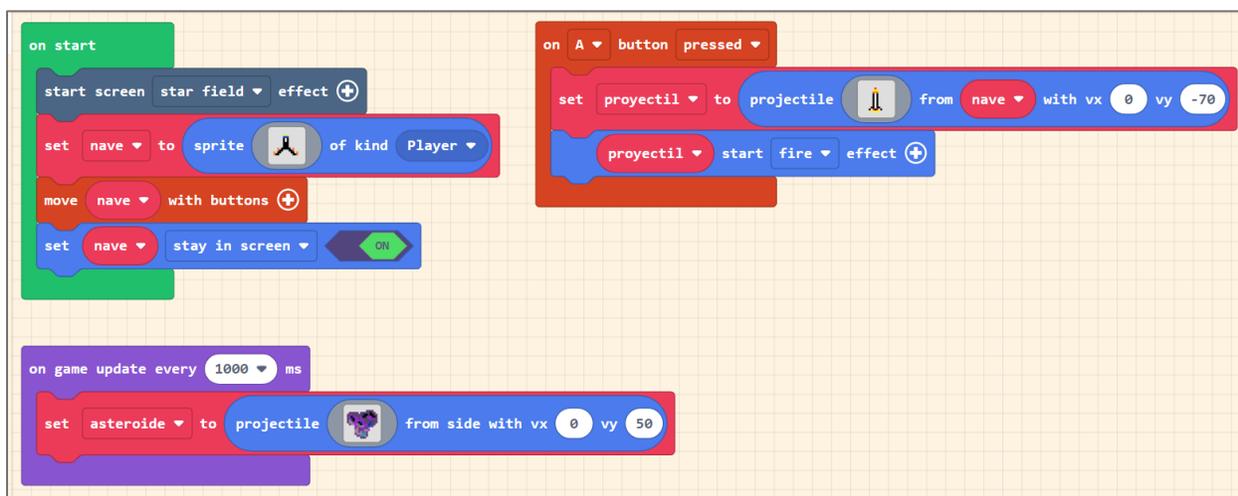
Esta instrucción establece otro tipo de ubicación inicial para las coordenadas XY en la pantalla de la consola Arcade.

La posición de origen ($X = 0$, $Y = 0$) cuando la instrucción es "from side" se encuentra ubicada en la esquina inferior izquierda. Es fija y no está ubicada en el personaje.

Verifique el su resultado en la siguiente pantalla:

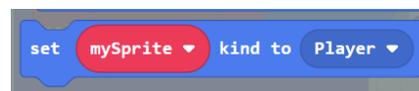


Disposición general de bloques:



Paso 25. Ahora asignaremos a la variable asteroide su tipo de participación en el juego o role. De acuerdo con el plan inicial, el asteroide es un **enemigo**.

Seleccione el inventario de bloques la categoría **"SPRITE"**. Localice el bloque *"set mySprite kind to Player"*.

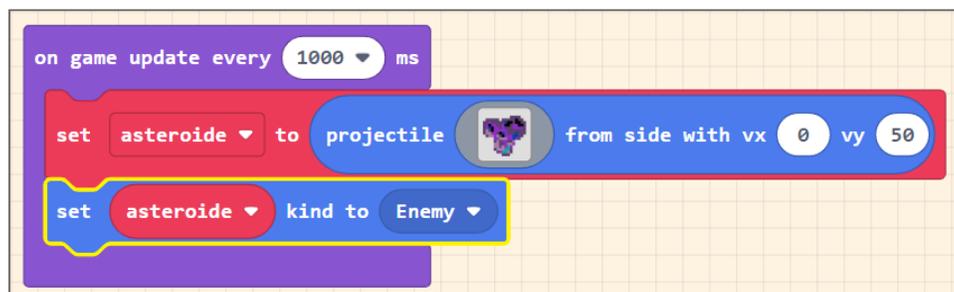


Seleccione el bloque y conéctelo con el bloque *"set asteroide to projectile – from side with vx 0 vy 50"*.

Nuevamente aparece la advertencia de inconsistencia de variables. Pero en esta ocasión el personaje es el **asteroide**. Realice el cambio la variable *mySprite* por **"asteroide"**.

Paso 26. Utilizando el puntero del ratón (mouse), ubíquese en la casilla *"Player"*, haciendo clic sobre la flecha se despliega el menú de opciones de pantalla. Seleccione la opción **"Enemy"**.

Verifique sus resultados en la siguiente figura:



LLUVIA DE ASTEROIDES.

Es más fácil de lo que parece crear el efecto de una **lluvia de asteroides**. Para lograrlo solo será necesario variar en forma aleatoria el valor numérico de la coordenada X. Para ello utilizaremos la función **"random"**.

NOTA: La función *random* permite generar una secuencia de números enteros aleatorios de tal manera que no conocemos la secuencia numera que se genera, y es justo este patrón de valores lo que permitirá actualizar la posición del asteroide cada segundo en forma aleatoria.

Paso 27. Seleccione nuevamente en el inventario de bloques la categoría **"SPRITE"**. Localice el bloque *"set mySprie X to 0"*.

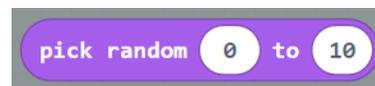


Seleccione el bloque y conéctelo con el bloque *"set asteroide kind to Enemy"*.

Nuevamente aparece la advertencia de inconsistencia de variables. Pero en esta ocasión el personaje es el asteroide. Realice el cambio la variable *mySprite* por **"asteroide"**.

Paso 28. Ahora determinaremos el nuevo valor de la coordenada X utilizando la función *random*. La función *random* es una función *matemática*.

Seleccione en el inventario de bloques la categoría **“MATH”**. Localice el bloque **“pick random 0 to 10”**.



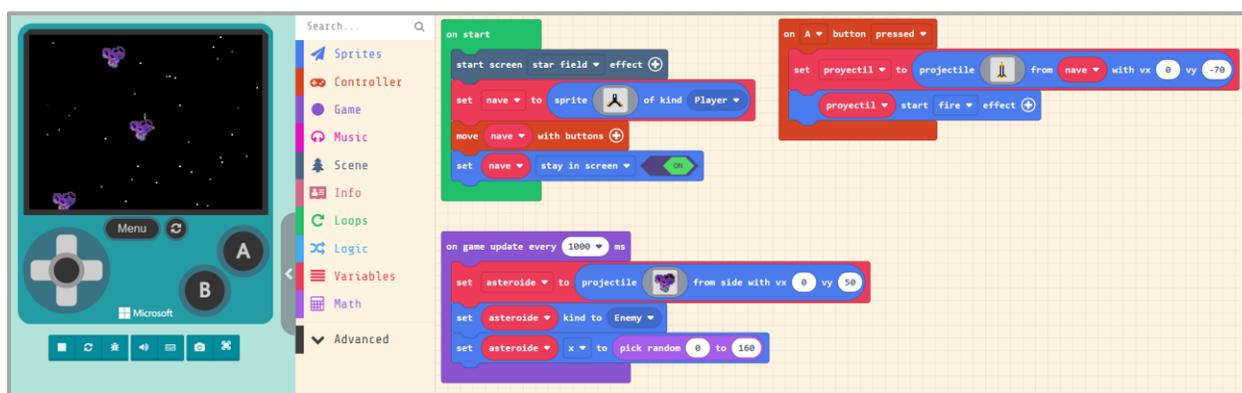
Seleccione el bloque y conéctelo en la ranura del valor 0 del bloque **“set mySprite X to 0”**.

Ahora, cambien el valor numérico superior de la función random de 10 por **160**. Este nuevo valor permitirá establecer un nuevo rango de números que van del 0 a 160, y en consecuencia la posición del asteroide cambiará entre la posición de X=0 y la posición X=160 cada segundo en forma aleatoria. El efecto visual será una lluvia de asteroides.

Active el programa y verifique el resultado en la siguiente figura.



Lluvia de asteroides y disposición final de bloques:



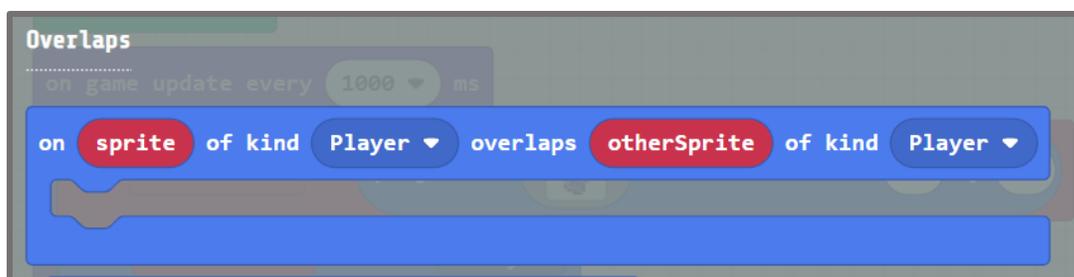
Efecto: DESTRUCCION DE ASTEROIDES

Para lograr programar la función para destruir un asteroide emplearemos la función DETECTAR PERSONAJE o acorde a la terminología del juego Arcade, la función OVERLAP SPRITE.

El bloque OVERLAP (o SUPERPOSICIÓN), permitirá que el proyectil pueda detectar un asteroide sea destruido.

Paso 29. Seleccione en el inventario de bloques la categoría **“SPRITE”**.

Localice el bloque **“on sprite of kind Player overlaps otherSprite of kind Player”**.



Seleccione el bloque y arrástrelo a la ventana de edición de programa.

El bloque seleccionado es un bloque con una función lógica CONDICIONAL. Esta función establece que **Si** un personaje cualquier (Sprite) del tipo Jugador (Player) se SOBREPONEN (overlaps) sobre otro Jugados (otherSprite) del tipo Jugador; entonces se ejecuta el programa en su interior.

Debemos modificar los parámetros del bloque OVERLAP, para que la condición sea:

“cuando un asteroide (o enemigo) se sobre posicione a un proyectil, entonces el asteroide y el proyectil se destruirán”

Para ello debemos cambiar el role de cada jugador en el bloque OVERLAP.

Paso 30. Utilizando el cursor del ratón (mouse), ubíquese en la primera casilla “Player”, del bloque OVERLAP, y haciendo clic sobre la flecha se despliega el menú de opciones de roles de personajes. Seleccione la opción **“Enemy”**

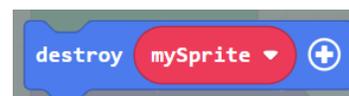
Paso 31. Utilizando el cursor del ratón (mouse), ubíquese en la segunda casilla “Player”, del bloque OVERLAP, y haciendo clic sobre la flecha se despliega el menú de opciones de roles de personajes. Seleccione la opción **“Projectile”**

Verifique su resultado en la siguiente figura:



Paso 32. Ahora coloquemos la acción de destrucción cuando ambos personajes se sobre posicionen.

Seleccione en el inventario de bloques la categoría **“SPRITE”**. Localice el bloque **“destroy mySprite”**. Seleccione el bloque, y conéctelo con el bloque OVERLAP.



Tal como se espera, la advertencia de inconsistencia de variables aparece nuevamente. Realice el cambio la variable mySprite por **“sprite”**.

Utilizando el cursor del ratón (mouse), ubíquese en la **primera variable "sprite"** del bloque OVERLAP, y haciendo clic sobre sobre la variable desplácela hasta la casilla "**mySprite**" del bloque **destroy**.

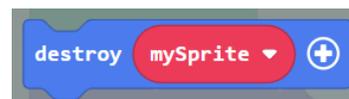
Automáticamente se genera un enlace de conexión de color amarillo. Siga el enlace de conexión, y coloque la variable "**Sprite**" en la casilla del bloque **destroy**.



Verifique su resultado:



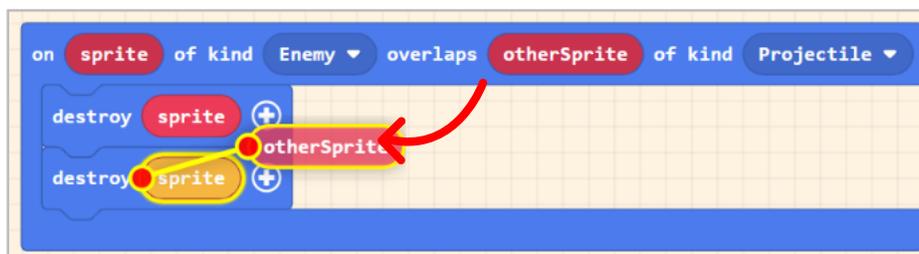
Paso 33. Seleccione nuevamente del inventario de bloques la categoría "**SPRITE**". Localice una vez más el bloque "**destroy mySprite**". Seleccione el bloque, y conéctelo con el bloque OVERLAP.



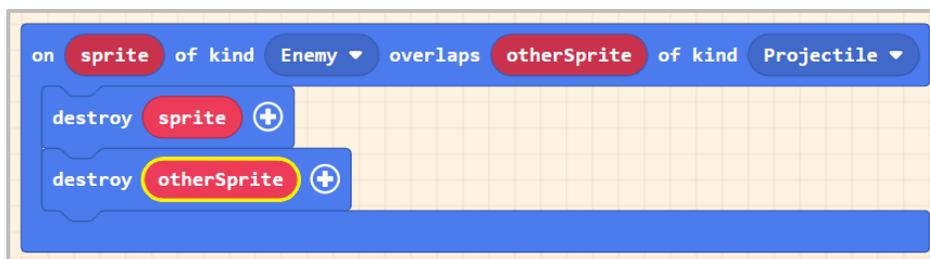
Tal como se espera, la advertencia de inconsistencia de variables aparece nuevamente. Realice el cambio la variable mySprite por "**otherSprite**".

Nuevamente utilizando el cursor del ratón (mouse), ubíquese en la **segunda variable "otherSprite"** del bloque OVERLAP, y haciendo clic sobre sobre la variable desplácela hasta la casilla "**mySprite**" del bloque **destroy**.

Automáticamente se genera un enlace de conexión de color amarillo. Siga el enlace de conexión, y coloque la variable "**otherSprite**" en la casilla del bloque **destroy**.



Verifique su resultado:

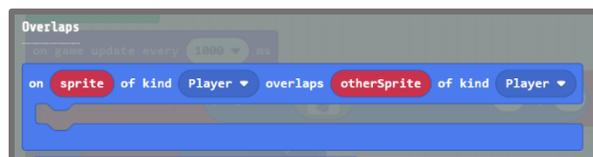


Con el desarrollo de las etapas anteriores, el juego está casi programado en su totalidad. Se han establecidos personajes, poderes, escenario, y configurado funciones y el botón A para poder ejecutar cada una de las etapas requeridas para el funcionamiento apropiado del juego. Queda ahora por establecer las reglas del juego.

ESTABLECIENDO REGLAS.

Regla: La nave espacial será destruida cuando un asteroide colisione con ella. La nave espacial podrá soportar la colisión de 3 asteroides. La nave espacial al colisionar con el tercer asteroide habrá fallado en su misión y el juego finaliza

Paso 34. Seleccione de la categoría “**SPRITE**”, el bloque **OVERLAP**: “**on sprite of kind Player overlaps otherSprite of kind Player**”.



Seleccione el bloque y arrástrelo a la ventana de edición de programa.

Debemos modificar los parámetros del bloque **OVERLAP**, para que la condición sea:

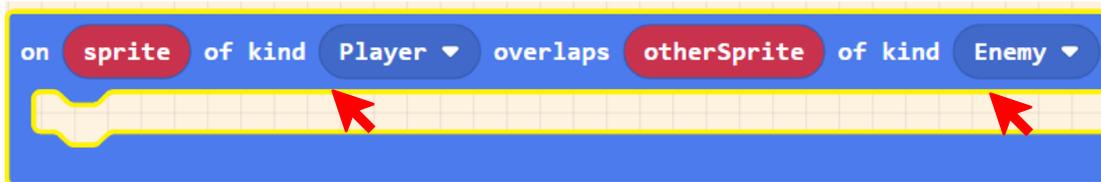
“cuando la nave (jugador) se sobre posicione a un asteroide (enemigo), entonces la nave pierde una vida y el asteroide se destruye”

Para ello debemos verificar nuevamente el role de cada jugador en el bloque **OVERLAP**.

NOTA: observe que solo será necesario cambiar una casilla. La primera casilla del bloque **OVERLAP** ya indica que para el personaje “*sprite*” su role es de jugador (*Player*), por lo que no será necesario modificar el primer parámetro.

Paso 35. Utilizando el cursor del ratón (mouse), ubíquese en la segunda casilla “*Player*”, del bloque **OVERLAP**, y haciendo clic sobre la flecha se despliega el menú de opciones de roles de personajes. Seleccione la opción “**Enemy**”

Verifique su resultado en la siguiente figura:



Ahora programaremos los eventos que hacen que se cumpla la regla. El primero de ellos es que el asteroide al colisionar con la nave espacial se destruirá.

Paso 36. Seleccione nuevamente del inventario de bloques la categoría **"SPRITE"**. Localice una vez más el bloque **"destroy mySprite"**. Seleccione el bloque, y conéctelo con el bloque OVERLAP.

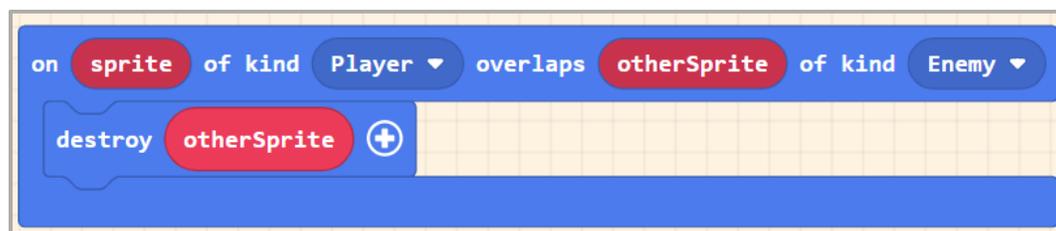
Tal como se espera, la advertencia de inconsistencia de variables aparece nuevamente. Realice el cambio la variable *mySprite* por **"otherSprite"**.

Nuevamente utilizando el cursor del ratón (mouse), ubíquese en la **segunda variable "otherSprite"** del bloque OVERLAP, y haciendo clic sobre sobre la variable desplácela hasta la casilla **"mySprite"** del bloque **destroy**.

Automáticamente se genera un enlace de conexión de color amarillo. Siga el enlace de conexión, y coloque la variable **"otherSprite"** en la casilla del bloque **destroy**.



Verifique el resultado:



El bloque anterior permitirá que el asteroide se destruya al colisionar con la nave espacial.

Ahora programaremos el número de intentos o número de vidas que tendrá la nave espacial para finalizar la misión.

Paso 37. Seleccione el inventario de bloques la categoría **“INFO”**. Localice el bloque **“change life by -1”**. Seleccione el bloque, y conéctelo con el bloque OVERLAP.



El bloque seleccionado automáticamente restará una vida al personaje (nave espacial) cada vez que se cumpla la condición establecida en el bloque OVERLAP. Es decir, cada vez que el personaje (nave espacial) se sobre posicione en un enemigo (asteroide), perderá una vida (o un intento por completar su misión).



IMPORTANTE: Automáticamente el bloque **change life by -1**, brinda al personaje (nave espacial) un máximo de 3 vidas.

Paso 38. Agregaremos un efecto al momento de colisión de la nave con un asteroide. El efecto consistirá en hacer vibrar la nave espacial por la colisión, y este efecto se logra en forma visual haciendo vibrar el enfoque de la cámara del juego.

Seleccione el inventario de bloques la categoría **“SCENE”**. Localice el bloque **“camera shake by 4 pixels for 500 ms”**.

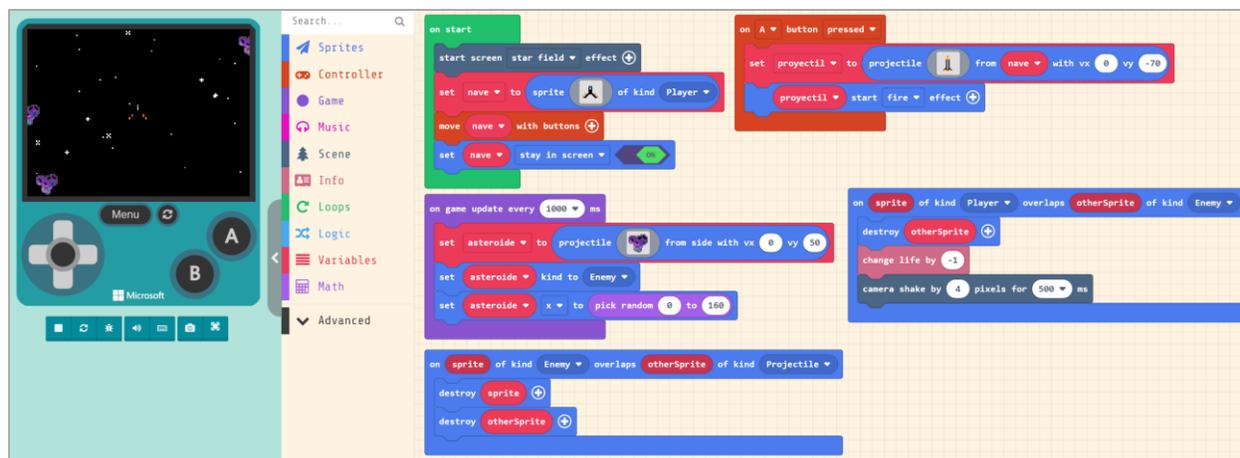


Seleccione el bloque, y conéctelo con el bloque OVERLAP.

El bloque seleccionado hará vibrar el enfoque de la cámara del juego en 4 píxeles durante un espacio de medio segundo (500 ms) únicamente cuando la nave espacial colisione o se sobreponga a un asteroide.



Hemos finalizado las etapas de programación del juego. Verifique la disposición final de los bloques y programación de todas las funciones en la siguiente figura:



Reflexiones

- Microsoft MakeCode no solo impulsa, a simple vista, iniciativas mundiales que apoyan el desarrollo de la ciencia de la computación en salón de clases, sino que de manera muy especial ofrece una colección de recursos para comenzar el aprendizaje y desarrollo del pensamiento computacional desde la primaria.
- Por su parte, las estrategias educativas basadas en juego (Game Based-Learning), como hemos visto, potencia el aprendizaje y autoaprendizaje de los estudiantes en todos los niveles educativos, pero especialmente su mayor impacto ocurre en la etapa inicial del aprendizaje (primera infancia y primaria).

Lea los siguientes artículos:

- [Crear un videojuego con Microsoft MakeCode Arcade: primeros pasos \(educaciontrespuntocero.com\)](https://educaciontrespuntocero.com/)

Responda a las siguientes interrogantes en forma personal. Comente en su red social sus reflexiones.

- ¿Cuál es la importancia de crear un video juego en el proceso de enseñanza y desarrollo de habilidades para la vida?
- ¿Se cumplen los principios del aprendizaje basado en juegos en el desarrollo y diseño de videojuegos?
- ¿Qué se logra entonces cuando un estudiante es el creador de un videojuego y no un consumidor? ¿Hay aprendizaje o es simplemente entretenimiento?

Referencias

- Microsoft MakeCode Arcade: <https://arcade.makecode.com>
- [Documentation \(makecode.com\)](https://makecode.com/)

Hemos aprendido cómo los simuladores y editores que ofrece MakeCode son recursos que pueden integrarse al proceso educativo de los estudiantes en las áreas de STEM, aprendizaje por proyectos, desarrollo de pensamiento computacional y programación.

Ahora conoceremos como integrar los recursos de MakeCode en el aula Microsoft Teams.