Desarrollo de Pensamiento Computacional con Microsoft MakeCode Arcade

Guía 2 de Orientación General y Prácticas

MakeCode Arcade – Personajes y recordando PacMan

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: No Pacaman

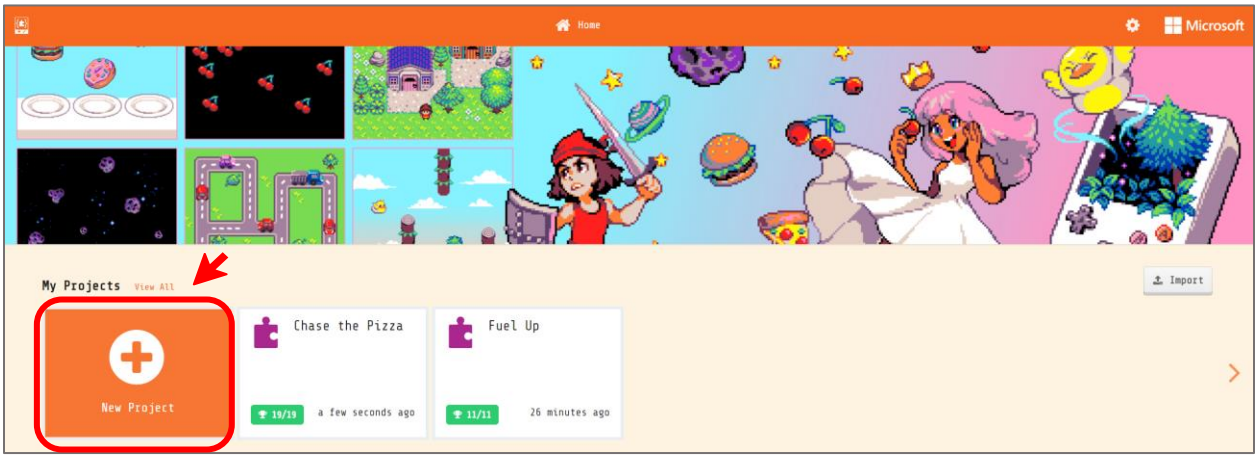
Objetivo: Proteger a Pacman de los enemigos fantasmas

Personaje	Poderes	Niveles	Escenarios	Normas
Héroe: Pacman Enemigos: Fantasma.	NO	Un solo nivel de juego	El desierto	Evitar ser destruido por los fantasmas Máximo de oportunidades: 3

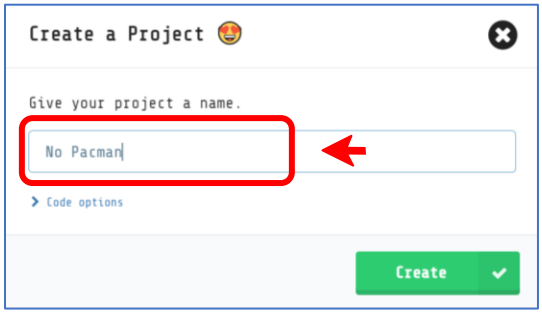
Área académica: Programación

Temas: Principios básicos de programación.

Paso 1. Iniciar un nuevo proyecto. Localice el botón "*Nuevo proyecto*", active el botón para iniciar el nuevo proyecto.

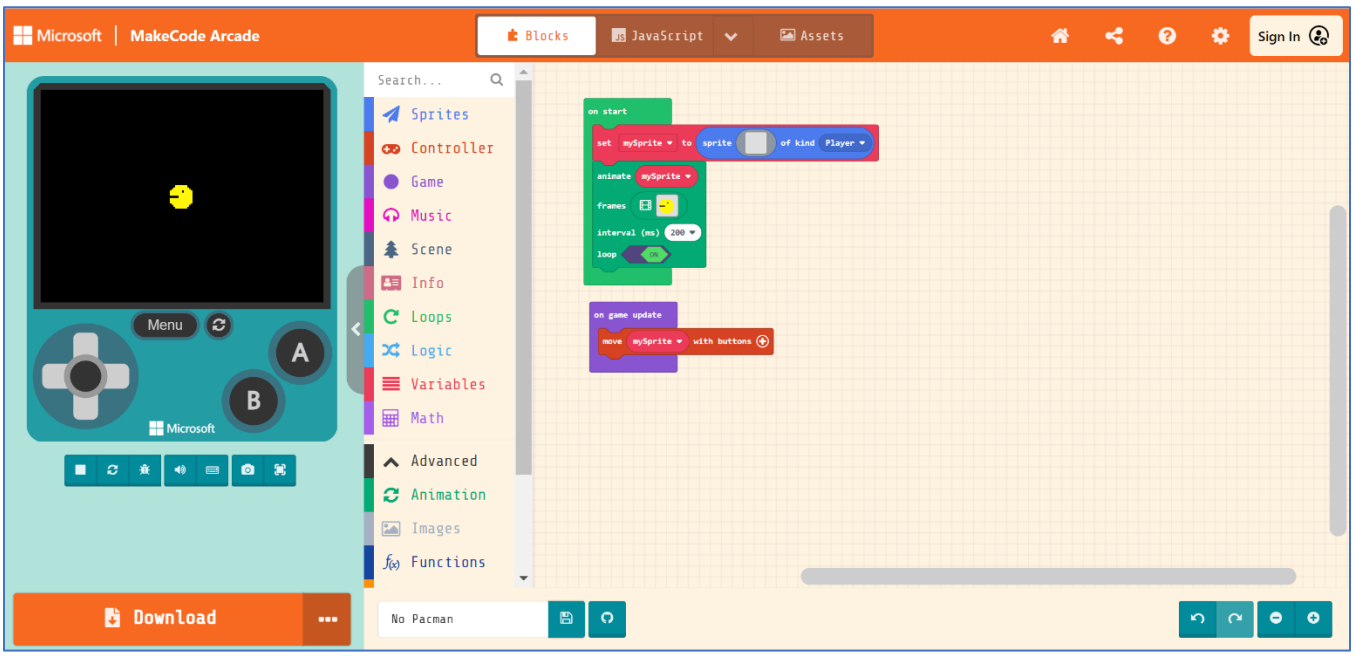


En la ventana emergente, escriba el nombre sugerido para su primer proyecto: "*No Pacman*". Y presione el botón "*Create*".



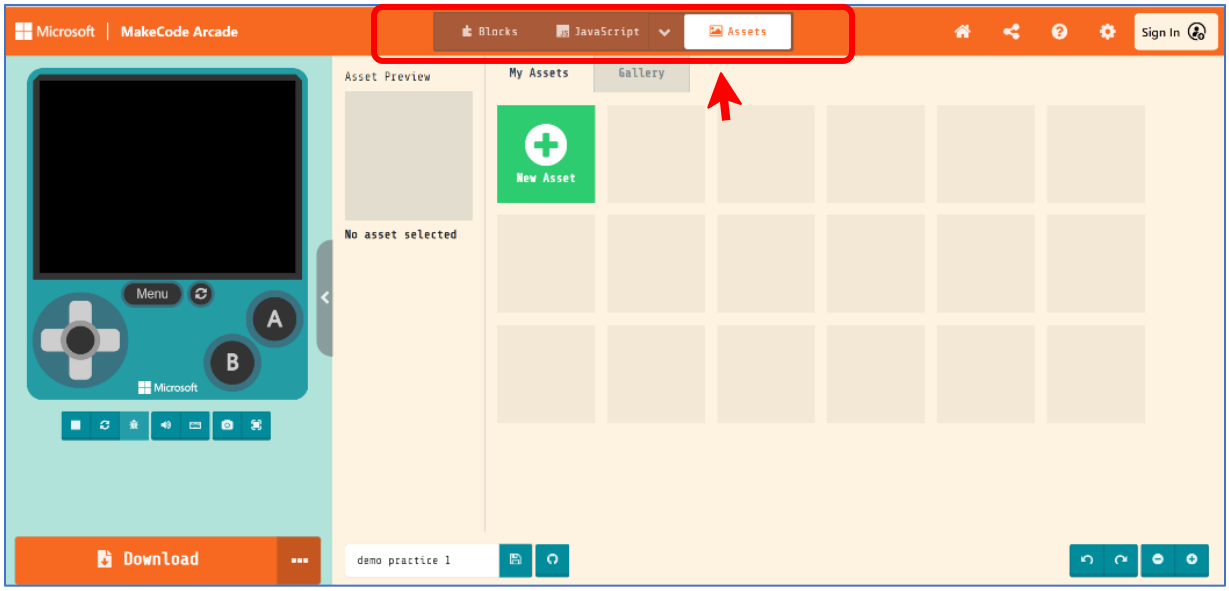
PERSONAJE

Comenzaremos por recrear al personaje ***Pacman***, el cual estará compuesta por una secuencia de dos imágenes, las cuales permitirán recrear sus movimientos básicos. El código final se muestra a continuación como una referencia de esta primera etapa.

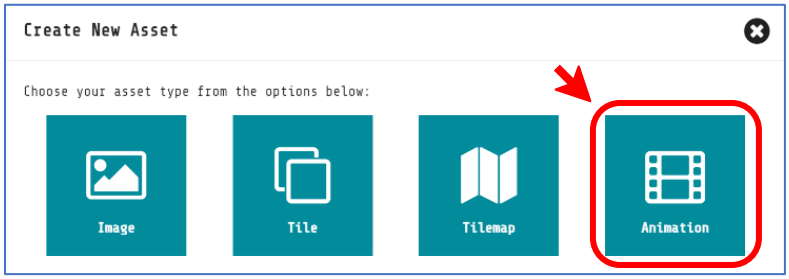


Paso 2. El menú Assets. Utilizaremos la opción de menú "assets" (o en español, activos o recursos). Para ello utilizaremos el seleccionador de desplazamiento que se encuentra en la parte superior de la pantalla. Desplaza hacia la derecha el seleccionar hasta la opción "Assets".

Verifique según la siguiente figura.



Paso 3. Selecciona el botón "*New Asset*". De las opciones que se muestran en la ventana emergente "Create New Asset", selecciona la opción "*Animation*".



Paso 4. Utilizando el editor de imágenes construya las siguientes imágenes utilizando el editor de imágenes de 16 x 16 pixeles.

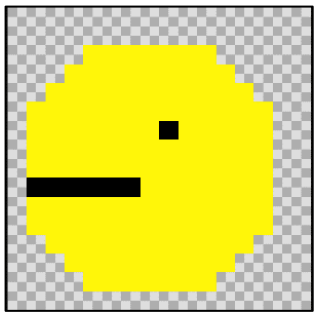


Imagen 1

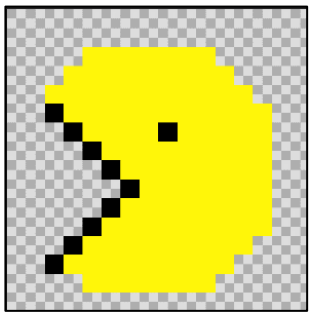


Imagen 2

¿Cómo se construyen las figuras?

Imagen 1

Las imágenes se construyen utilizando la *función círculo y seleccionando el color amarillo*.

Luego debe colocarse aproximadamente en el centro del área de edición, y con el apoyo del mouse puede expandir el punto amarillo hasta lograr la circunferencia.

Una vez lograda la circunferencia, utilice la función rellenar (siempre con el color amarillo).

Nuevamente se coloca en el centro del círculo, y haciendo clic izquierdo podrá colorar el interior del círculo y rellenar de color amarillo los espacios internos.

Ahora, cambiamos a la función lápiz, seleccionaremos el color negro y con el lápiz podemos realizar los detalles del ojo y la boca de la figura 1 de Pacman.

Verifique su resultado en la siguiente figura.

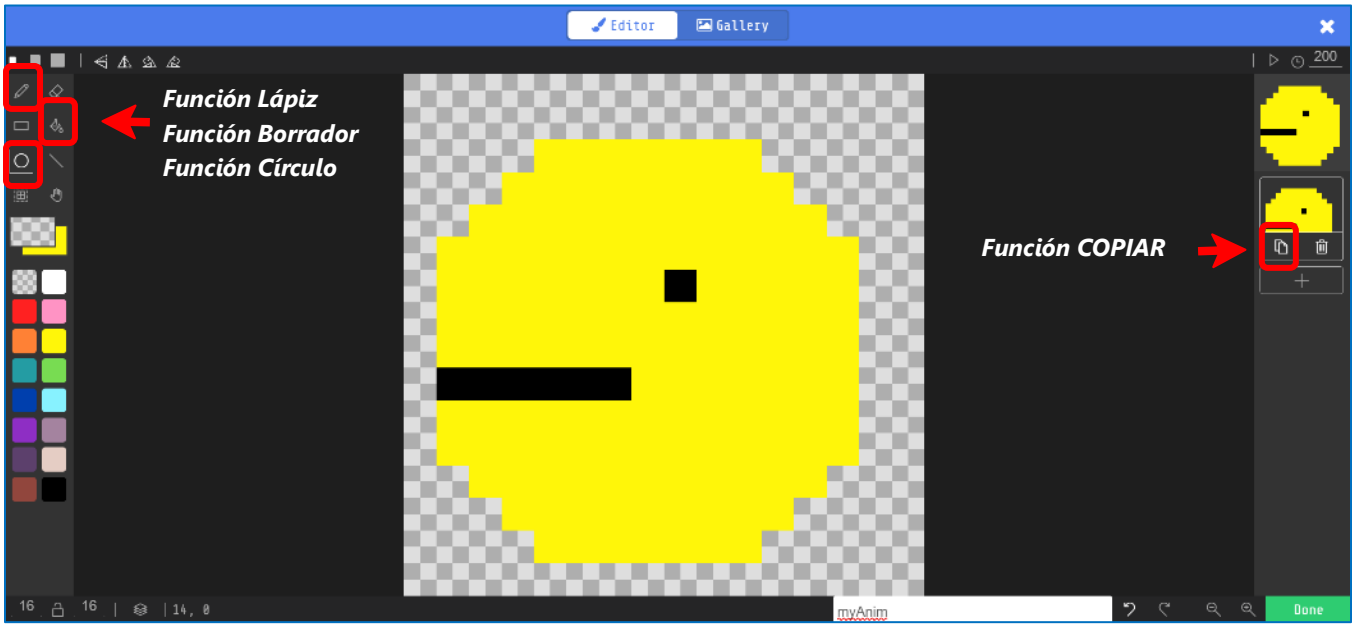
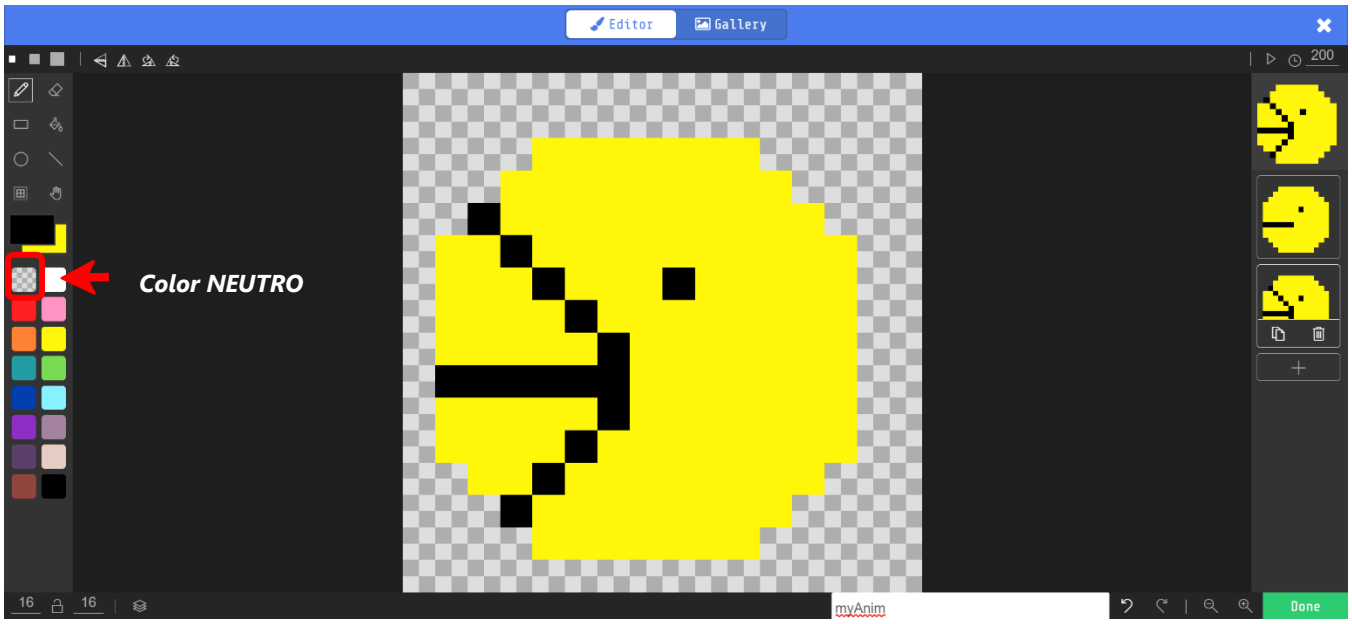


Imagen 2

Para la figura 2, es importante no perder la simetría que realizamos en la imagen 1. Por lo que realizaremos una copia de la imagen 1. Esto lo realizamos con el botón copiar que se indica en la pantalla anterior.

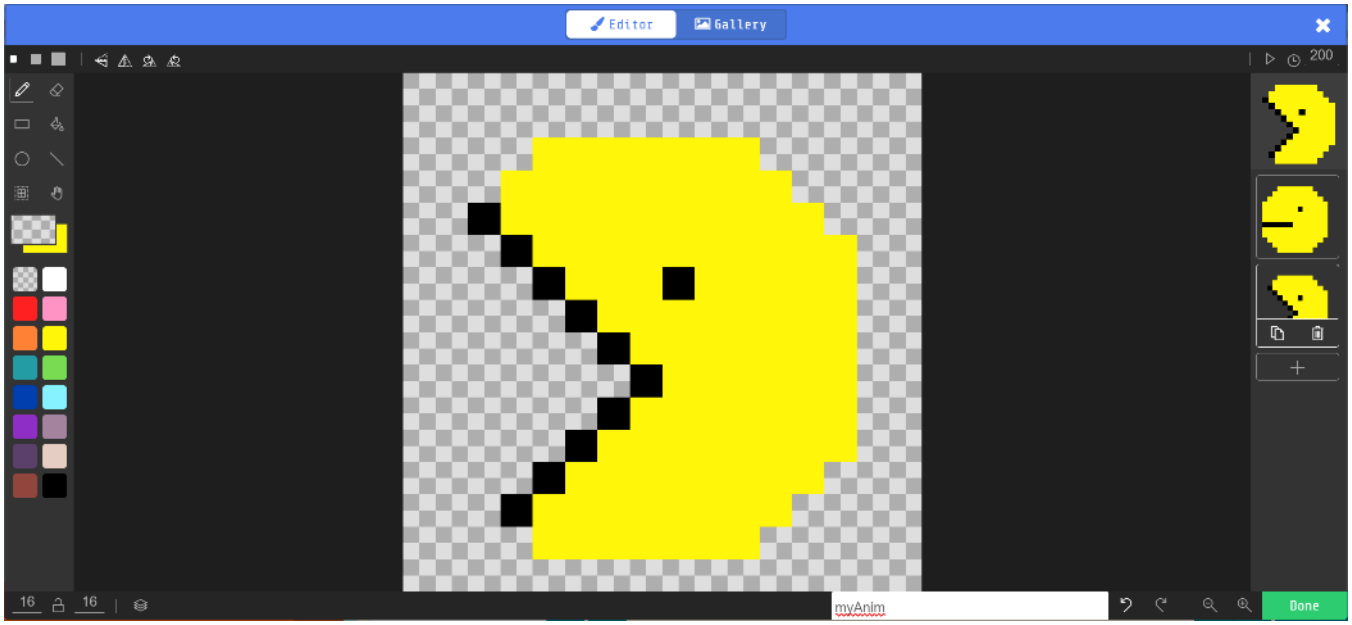
Ahora utilizaremos la función lápiz, y dibujaremos la nueva forma de la boca.

La nueva forma de la boca será una boca abierta. Revise la siguiente figura para comprobar su avance.



Ahora debemos utilizar debeos utilizar el color neutro, tal como se indica en la figura, para poder borrar el interior de la boca en la imagen 2.

La imagen 2 completa queda de la siguiente manera.



Hemos completado el diseño del personaje. Ahora verifiquemos la animación.

Verificando el funcionamiento de la animación.

La animación estará compuesta por la superposición simultanea de ambas imágenes, logrando el efecto de la apertura de la boca de Pacman, estableciendo el tiempo de cambio entre las imágenes.

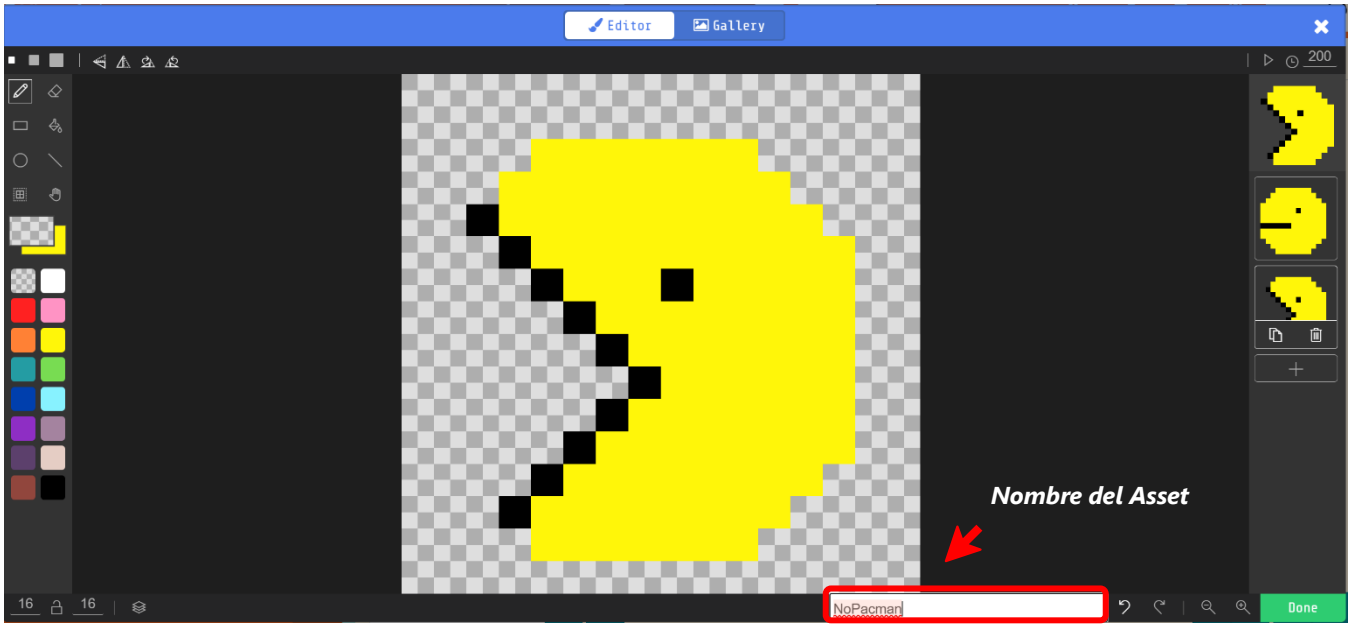
MakeCode Arcade nos permite verificar el funcionamiento de una animación mediante el uso de dos opciones: tiempo de animación y la verificación de su comportamiento.

Estas opciones las encontramos en la esquina superior derecha de la ventana de diseño. Presione ahora el botón "*play*", y compruebe el funcionamiento para un tiempo establecido de *200 milisegundos*.

Práctica: modifique los diferentes valores de tiempo y observe los cambios.

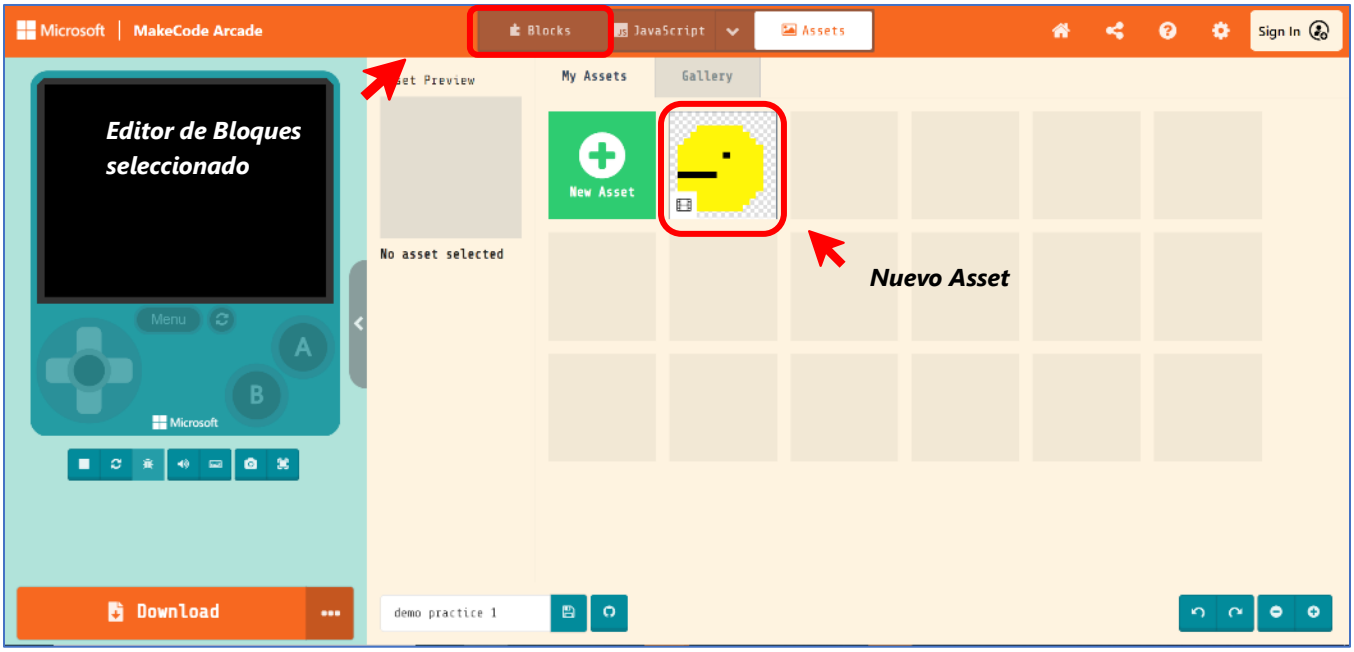
Paso 5. Nombre a la animación. Ahora estamos listos, y debemos brindarle un nombre a la animación creada para representar al personaje de Pacman.

Ubique la opción para colocar el nombre de la aplicación al final de la ventana de diseño, y coloque el nombre "NoPacman", tal y como se indica en la figura a continuación:



Presione el botón "Done" y automáticamente guardamos la animación como parte de nuestra colección de animaciones o figuras.

Observe ahora su colección de recursos (Assets), y verifique la nueva animación.



Ahora estamos listos para comenzar a programar. Para ello desplace nuevamente el seleccionador de la parte superior de la pantalla a la opción "Blocks", tal como se muestra en la figura, y regresaremos al editor de programación por bloques.

El personaje principal.

Para que nuestro personaje principal aparezca y se encuentre listo para su funcionamiento, será necesario iniciar el proceso de programación, y para ello utilizaremos las variables que MakeCode Arcade establece por definición.

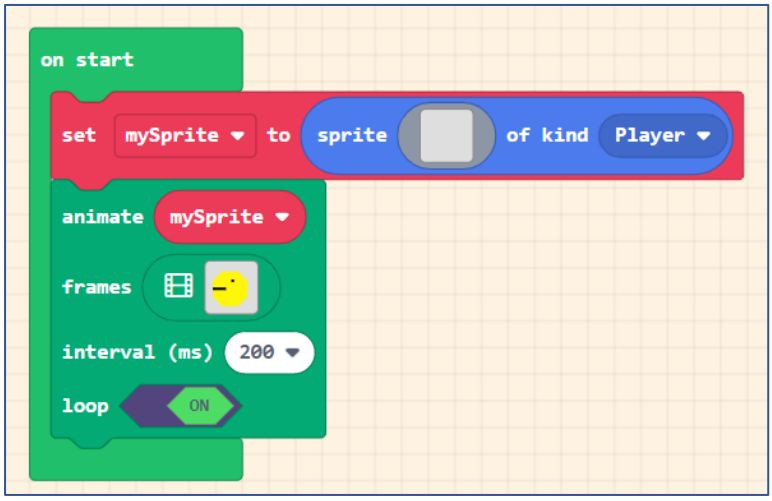
Los personajes son variables de programación, y el personaje principal, Pacman, se reconocerá con ***la variable MySprite.*** El termino Sprite es un término que en el ámbito de los videojuegos, los *sprites* son un conjunto de imágenes que representa un personaje u objeto (o una parte de ellos) de manera gráfica, y que se utiliza para poder crear cualquier efecto de movimiento o para cambiar su estado o posición en la escena.

Paso 6. El paso que daremos a continuación es establecer el programa que determine que la variable MySprite es el jugador (player), al momento de iniciar el juego.

Posteriormente, activaremos la animación que hemos preparado para representar al jugador. Este paso se desarrolla mediante tres tipos de bloques:

- ***On Start***, el cual permite que las acciones de programación escritas en su interior se activen en el momento de inicio del juego.
- ***Set MySprite***, de la familia de *bloques Sripte*; que define al personaje y
- el bloque ***Animate*** de la familia de bloques *Animation,* para activar la animación.

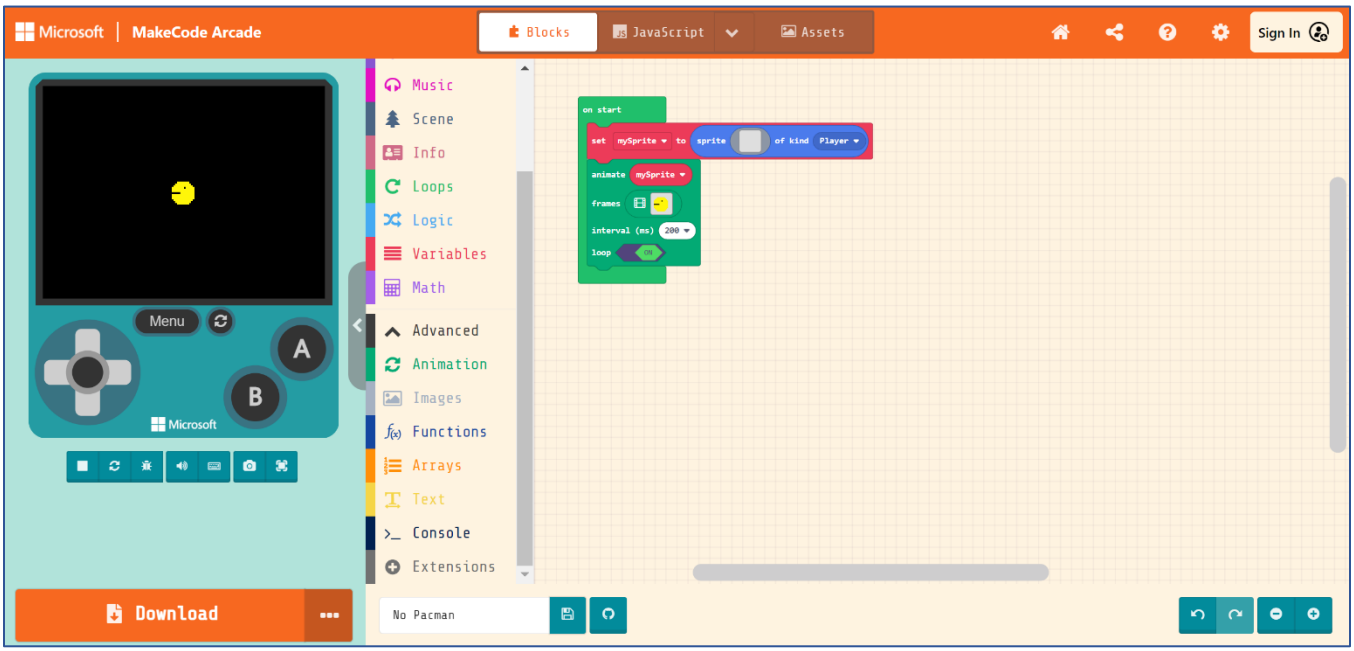
Realice el siguiente proceso de programación.



El bloque Set mySprite define la variable como una variable del tipo "Sprite", y de la clase jugador ("player").

El bloque Animate, establece que la variable mySprite estará compuesta por el conjunto de imágenes (frames) que diseñamos anteriormente, y con un intervalo de cambio de 200 milisegundos en un ciclo infinito de repeticiones activo (*loop on*).

Verifique sus resultados.



Nuestro personaje principal realiza su aparición por primera vez, y podemos comprobar que se coloca en el centro de la pantalla, y que el comportamiento de la animación es el esperado.

Paso 7. Colocando movimientos. Ahora programaremos los movimientos básicos para el personaje principal, y agregaremos dos bloques:

- ***On Game Update***, el cual permite repetir una acción de forma perpetua e indefinida, cuando se realiza algún cambio en el juego o variables del juego. Esta función nos permitirá mantener activo el control de movimientos durante todo el tiempo que decidamos jugar.

- ***Move Sprite with controlers (⊕),*** es el bloque nos permite desplazar al personaje con los controladores básicos.

Realice el siguiente proceso de programación complementario, tal como se indica en la figura.

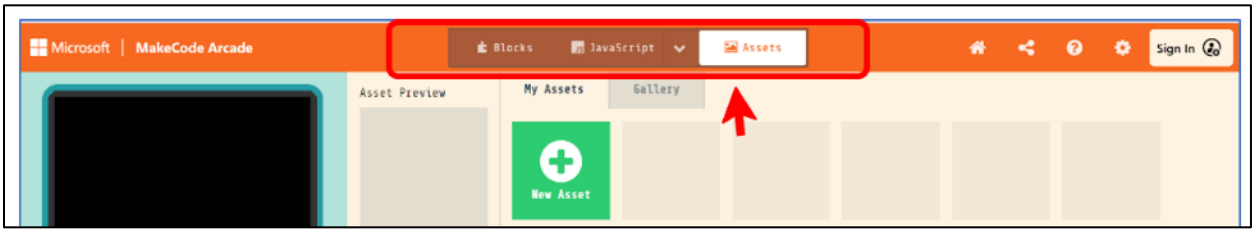
Verifique ahora su resultado y compruebe que puede mover al personaje principal por la pantalla del juego.

Observe que el personaje se sale de los límites de la pantalla, por lo que ahora será importante establecer los límites del juego, es decir, el escenario o mundo.

Esta acción también se puede realizar con los bloques de programación que limitan salir del espacio establecido por la pantalla del simulador, pero utilizaremos esta oportunidad para crear nuestro mundo.

Creando el escenario o mundo.

El escenario o mundo es también un tipo de recurso (o asset), por lo que ahora debemos cambiar nuevamente a la función "Assets", y hacemos clic nuevamente en la opción *New Asset*.



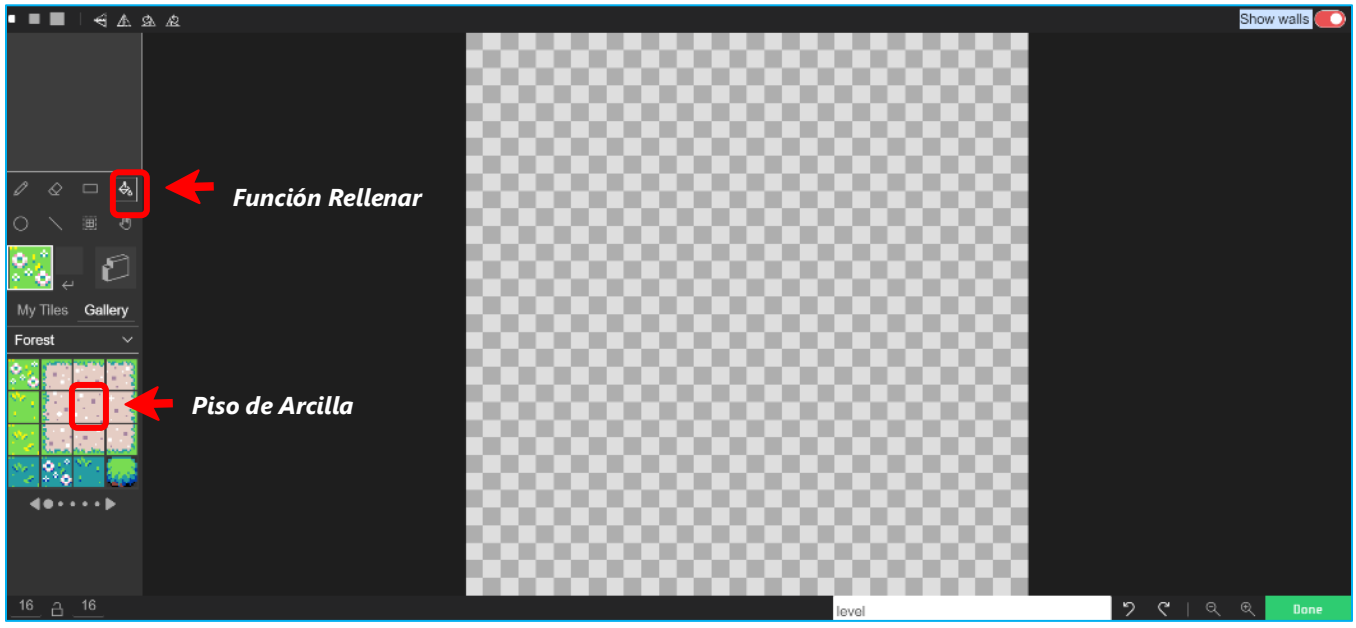
De las opciones que se nos presentan a continuación, tenemos dos alternativas para trabajar el escenario o mundo.

Tile (o mosaico): es una imagen que se muestra como una parte rectangular de la escena del juego. La escena del juego se crea con mosaicos colocados en una fila y columna en un mapa de mosaicos. El mapa de mosaicos contiene todos los mosaicos que se muestran en la escena. La escena del juego tiene una colección de mosaicos.

Tilemap (mapa de mosaicos): permite desarrollar niveles y definir espacios de juego para que los personajes de tus juegos se desplacen. Son escenarios predeterminados.

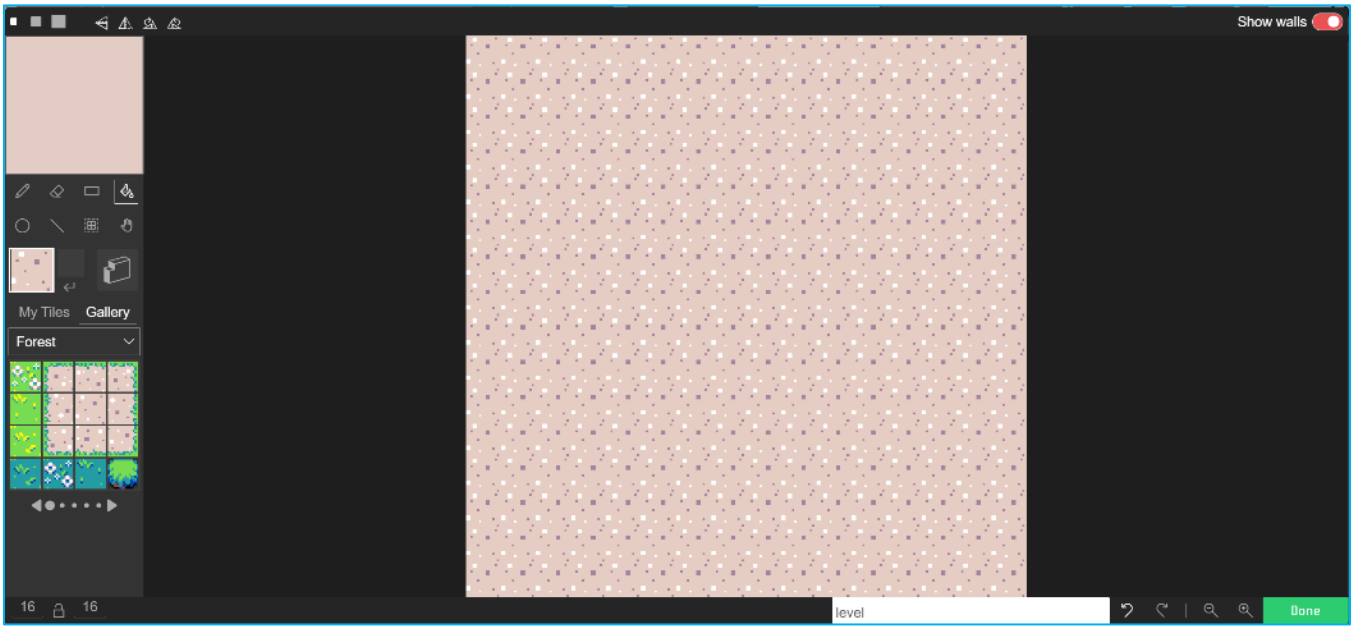
Paso 8. Utilizaremos la opción Tilemap.

Con la selección del mapa de mosaicos construiremos el mundo, partiendo por la selección del piso, y el muro del mundo. Verifique ahora la siguiente pantalla.

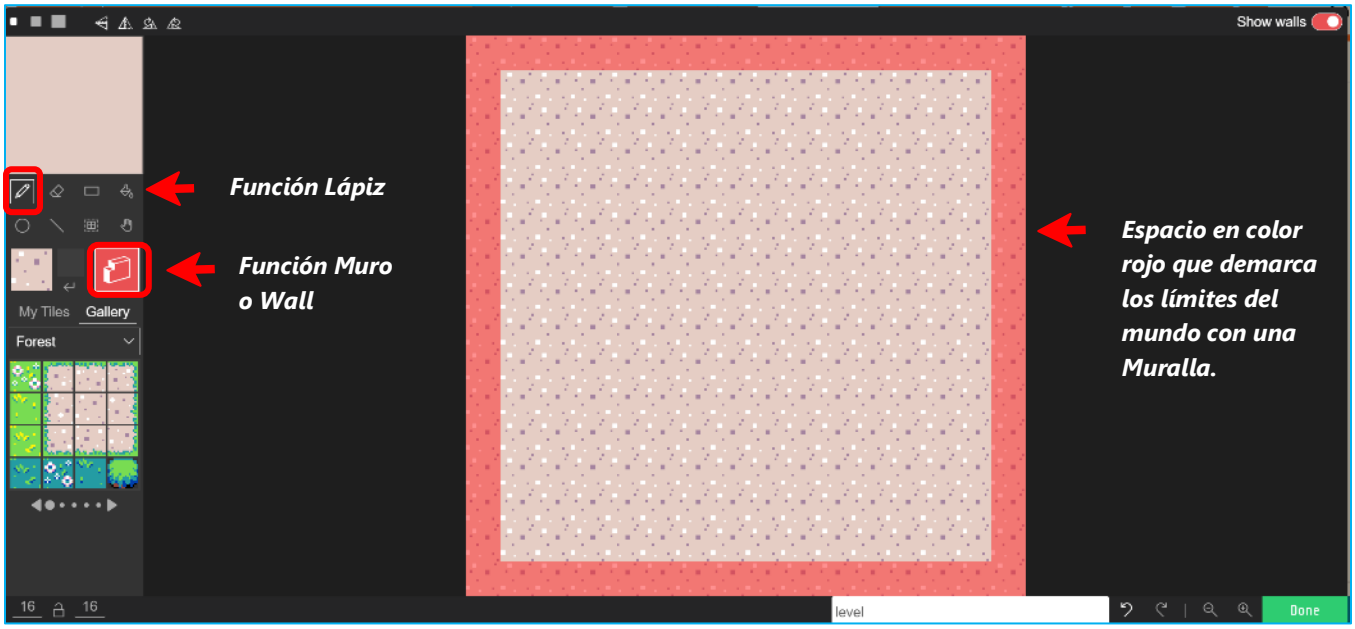


Con el editor de mundos, utilizaremos los bloques sugeridos de la colección Forest (o Bosque). Utilizaremos el bloque de arcilla indicado en la figura, y rellenaremos todo el espacio disponible del mundo para crear el piso del mundo en la venta de 60 x 60 pixeles.

Haga clic sobre la ventana de edición de mundos y verifique sus resultados acordes con la siguiente figura.

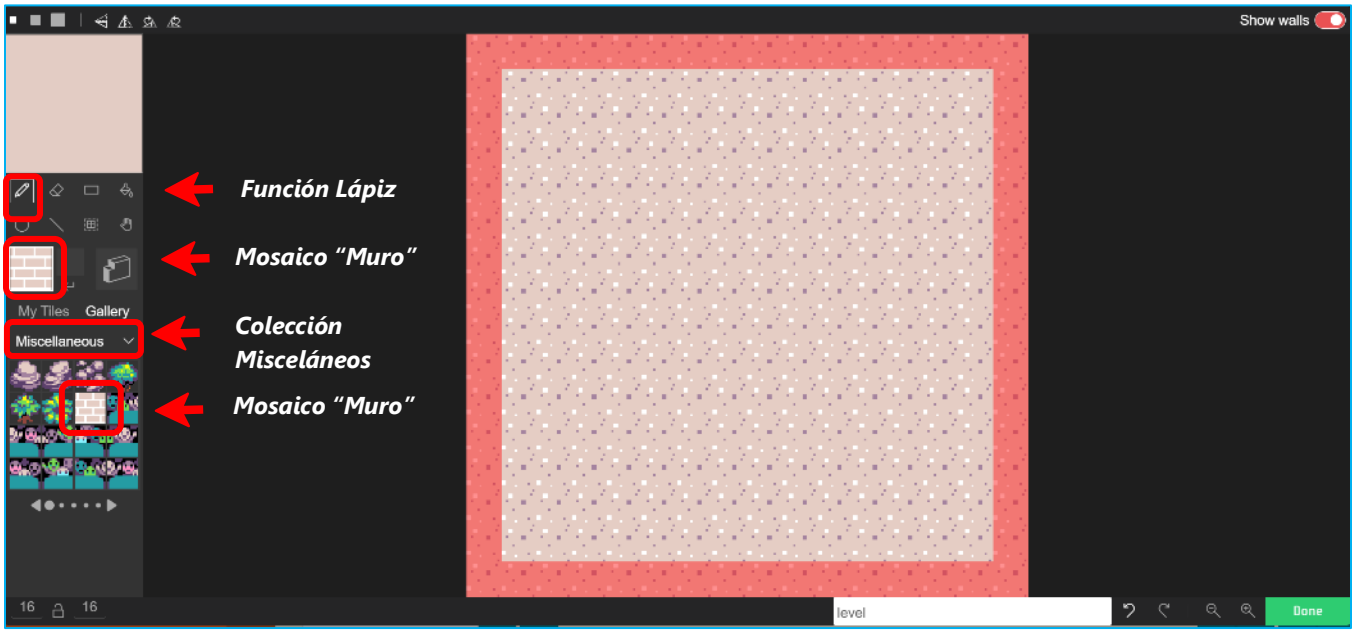


Ahora crearemos los límites del mundo con la función Muro (o Wall), y la función lápiz, tal como se muestra en la figura a continuación, y con la opción de colocar una muralla activa, lograremos establecer un perímetro que impida al personaje el desplazamiento sobre estos espacios.

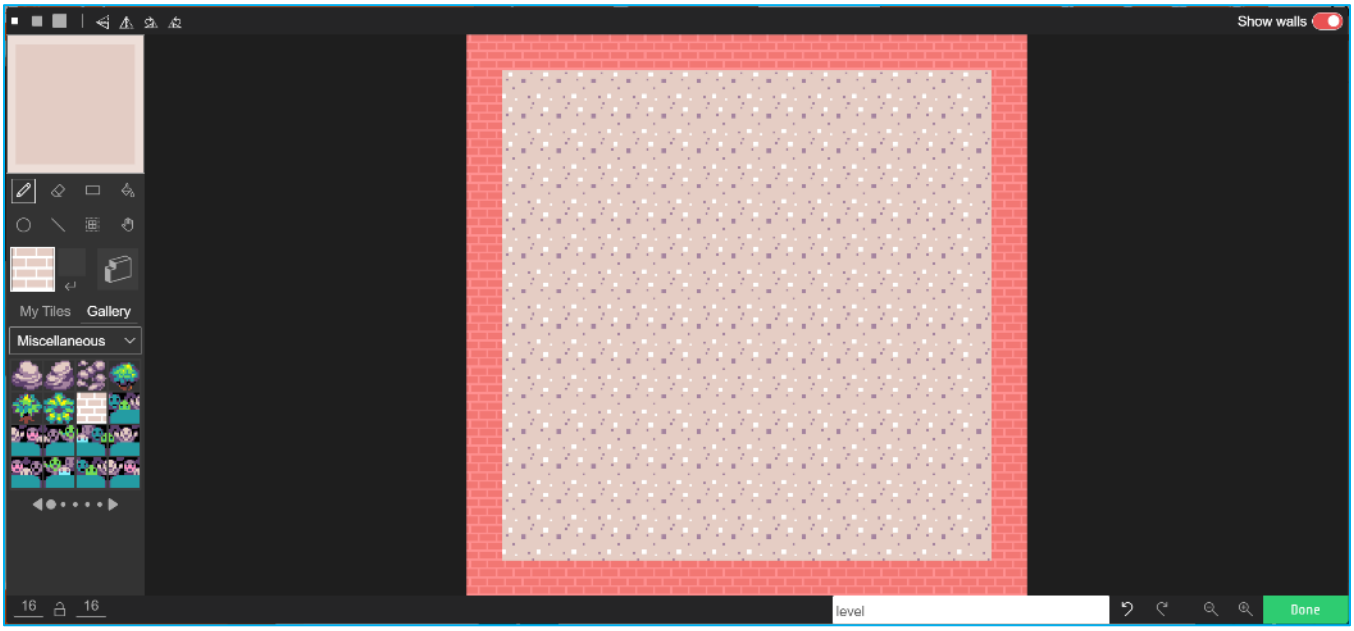


Puede observar en la figura anterior que, utilizando el mouse, logramos delimitar los límites del mundo. El color rojo establece los lugares del mapa en los cuales el personaje no podrá desplazarse.

Es importante ahora decorar los bordes de los límites del mundo. Para ello seleccionaremos la familia de *tilemap misceláneos*, y seleccionaremos el tipo de mosaico (o tile) muro (o wall), al mismo tiempo seleccionaremos nuevamente la función lápiz, tal como se indica en la siguiente figura:



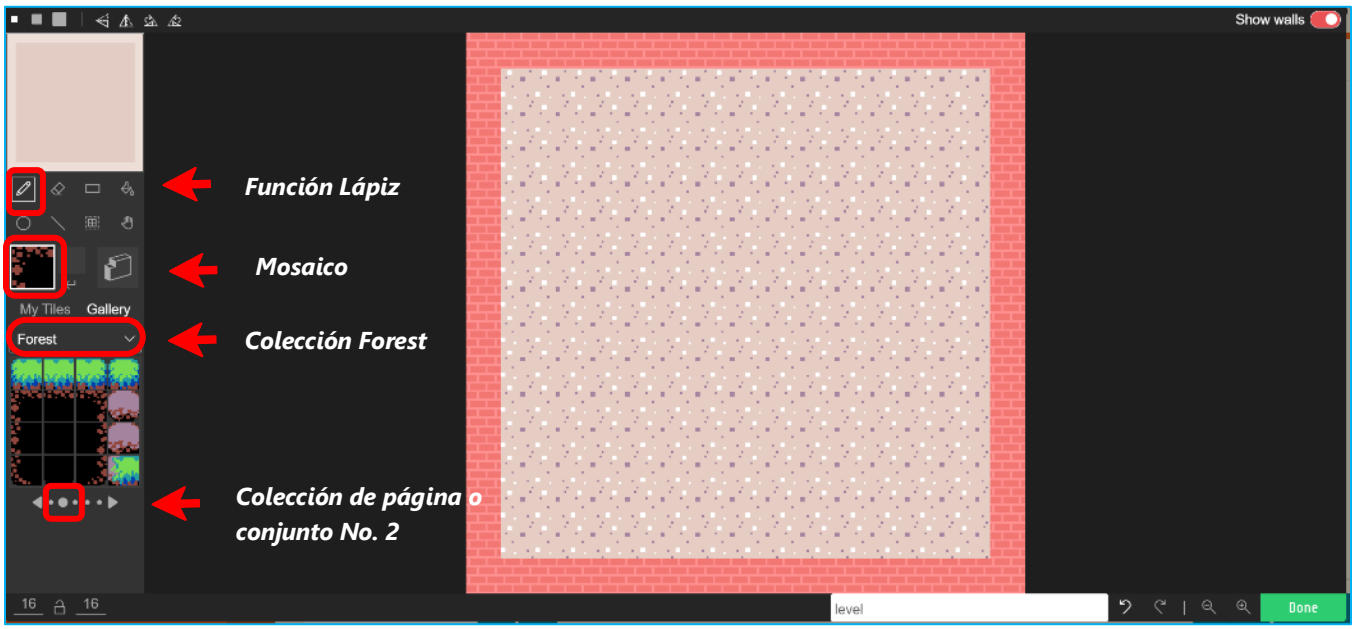
Verifique sus resultados en la siguiente figura:



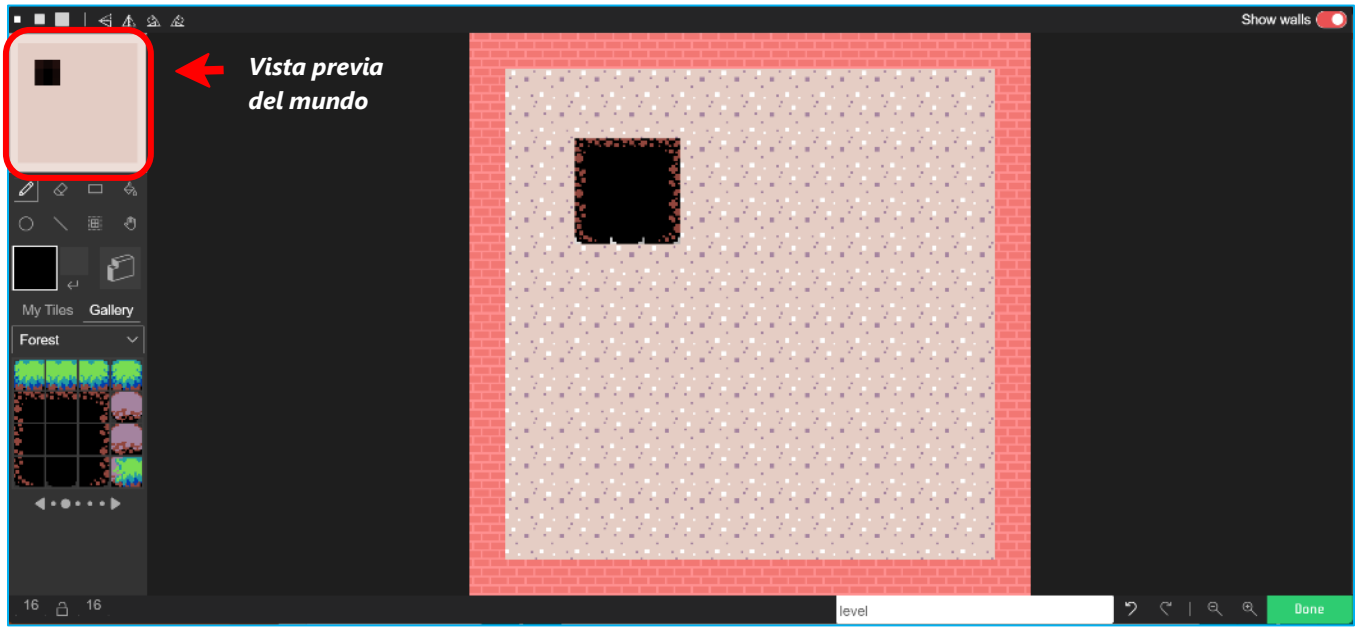
En la figura anterior podemos observar que el mosaico tipo muro lo hemos colocado sobre el marco de color rojo que delimita el juego con la intención de realizar la decoración apropiada a los límites del juego.

Obstáculos. Ahora colocaremos una serie de obstáculos que el jugador debe evitar durante el desarrollo del juego.

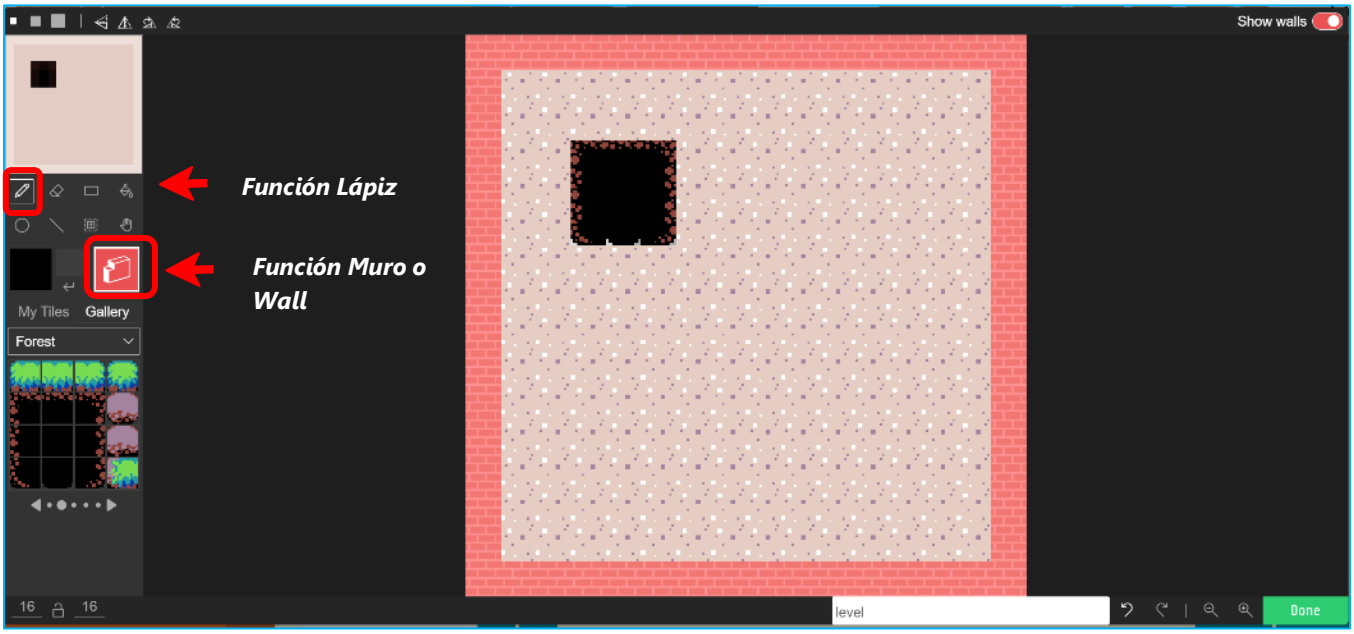
Paso 9. Utilizando nuevamente la función muro (o wall) y la colección de mosaicos Bosque (Forest), crearemos una serie de obstáculos. Seleccione la función lápiz, y la colección Forest para realizar la creación de los obstáculos tal como se muestra en la siguiente figura.



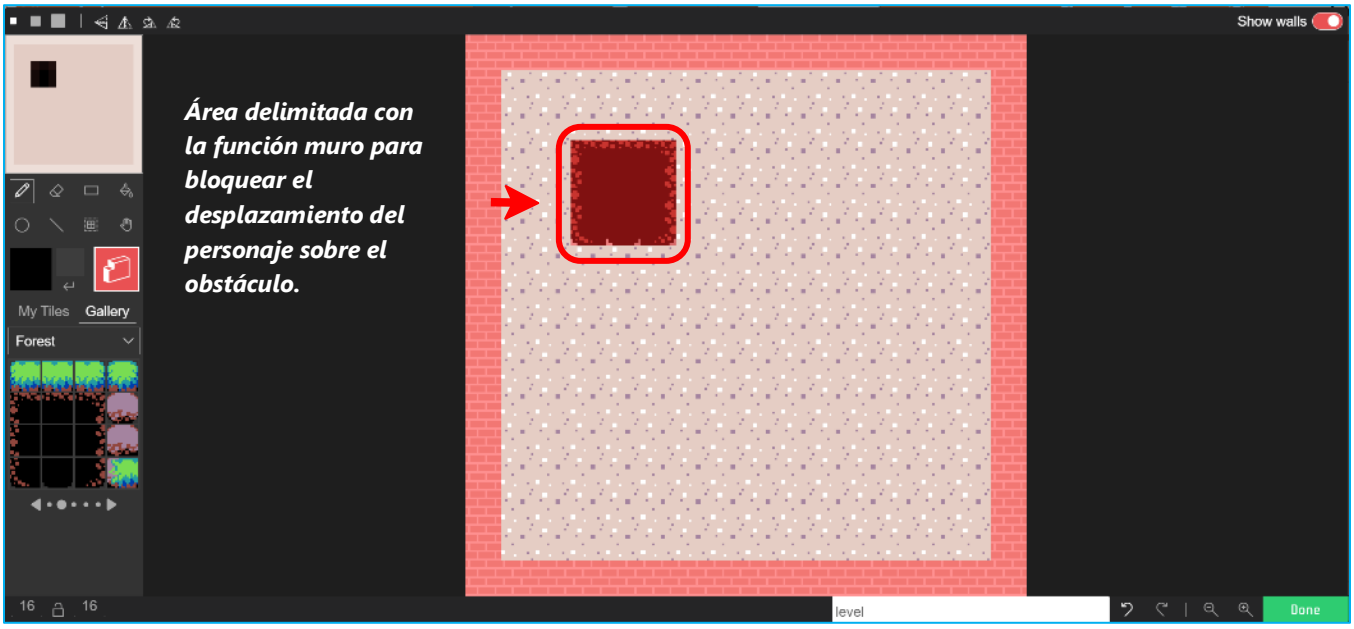
Paso 10. Utilizando la elección de mosaicos y funciones indicada, construya el primer obstáculo tal como se indica en la siguiente figura.



Hemos creado el primer obstáculo. Ahora ***debemos delimitar el área para que el personaje no se desplace sobre él***. Para ello seleccione nuevamente la función Muro (o Wall) y el lápiz, tal y como se indica a continuación.

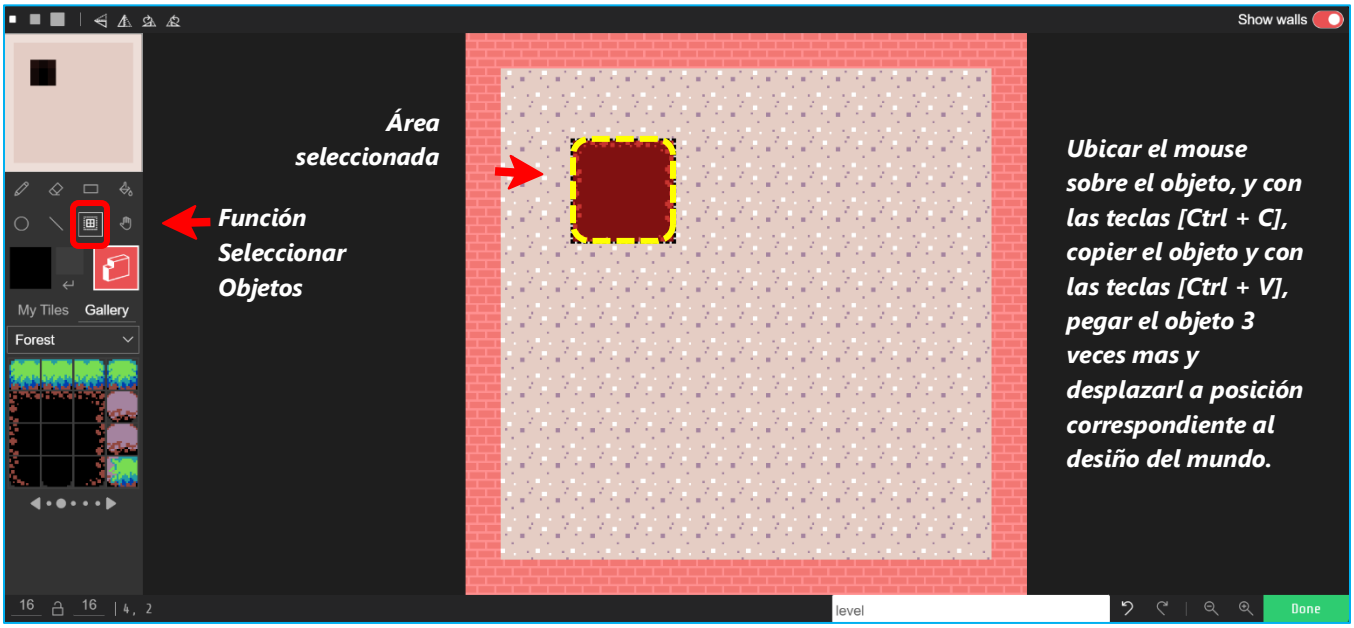


Proceda ahora a delimitar el área de desplazamiento, tal como se indica en la figura, de tal manera que se bloquee el desplazamiento del personaje sobre el obstáculo. Observe la siguiente figura y compruebe el resultado.

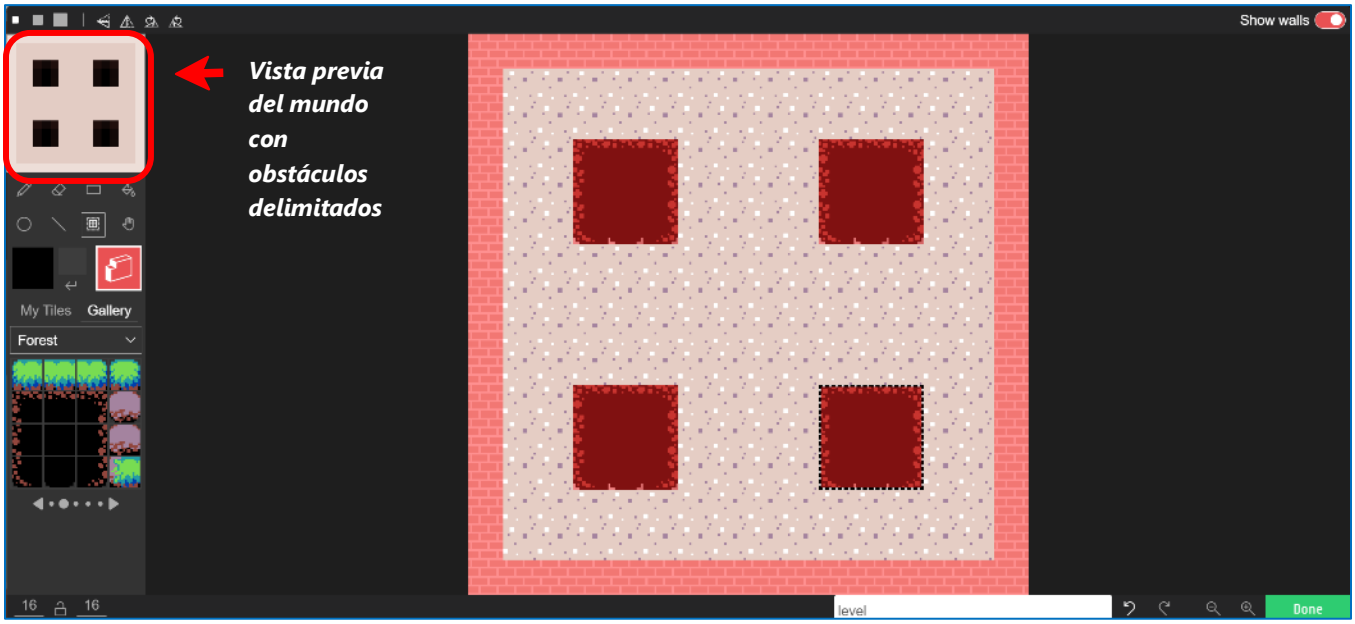


Repitamos ahora esta misa acción 3 veces mas, y coloquemos en total 4 obstáculos. Para ello utilizaremos la función de seleccionar objetos, junto con la función de copiar y pegar.

Observe la siguiente figura,



Paso 11. Ahora copiaremos el objeto una vez, y lo pegaremos 3 veces más, tal como se indica en la figura.



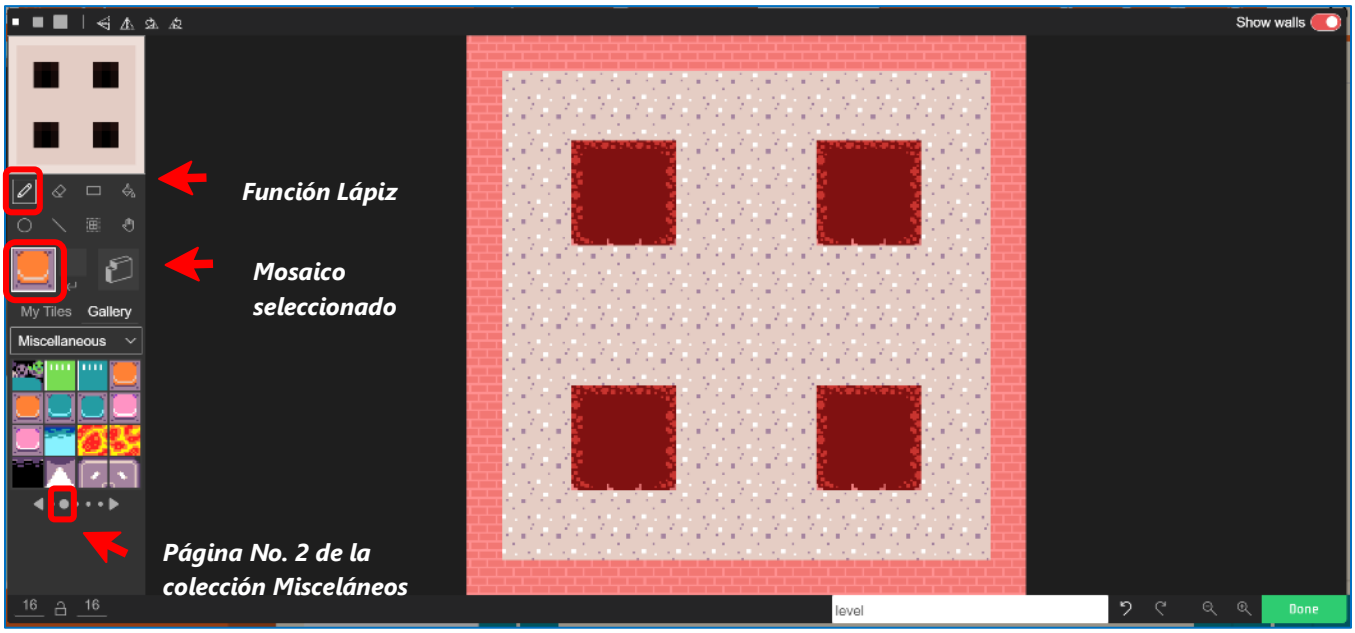
El escenario o mundo esta casi finalizado, podremos observar que el personaje podrá desplazarse en el interior del muro, y tendrá que evitar cada uno de los diferentes obstáculos.

Recordemos que la finalidad del juego es que nuestro personaje no se deje atrapar por los enemigos, y en consecuencia, el mundo es el único escenario en el cual el personaje debe desplazarse evitando a los enemigos durante un periodo de tiempo predeterminado para poder declararse como ganador.

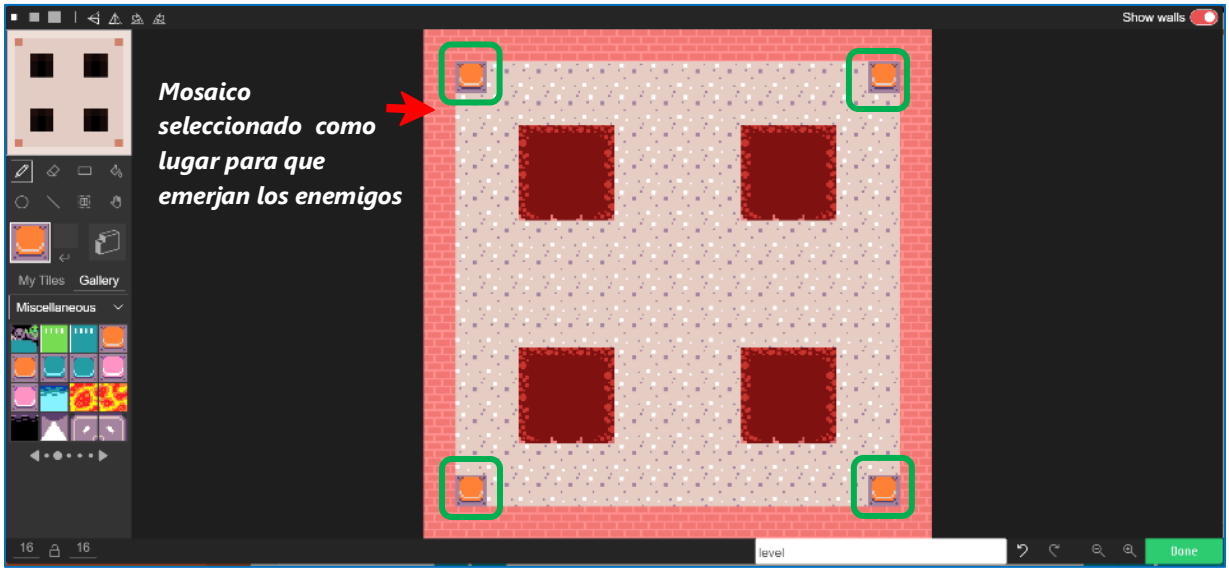
Lugar para generar enemigos en el mundo.

Para completar el mundo, colocaremos ahora *los espacios* que denominaremos ***"fuentes de enemigos"***, que será el lugar del cual emergerán los enemigos durante el desarrollo del juego.

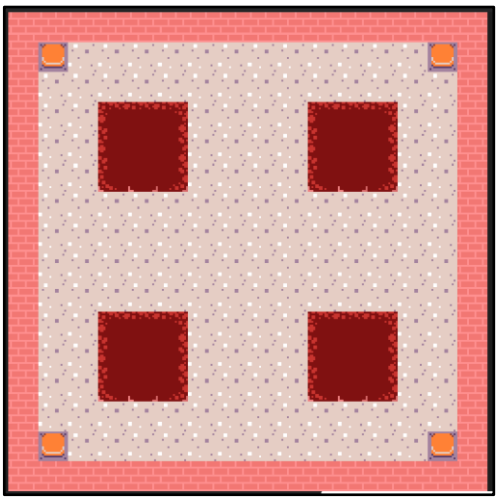
Paso 11. Para ello seleccionaremos la categoría de mosaicos "misceláneos" una vez más, la página No. 2 de esta colección mediante la selección del segundo punto indicado abajo del recuadro (observe la figura). Seleccionaremos también la función lápiz, y un mosaico que nos permita simular un botón en relieve. *Verifique esta selección de funciones en la siguiente figura*:



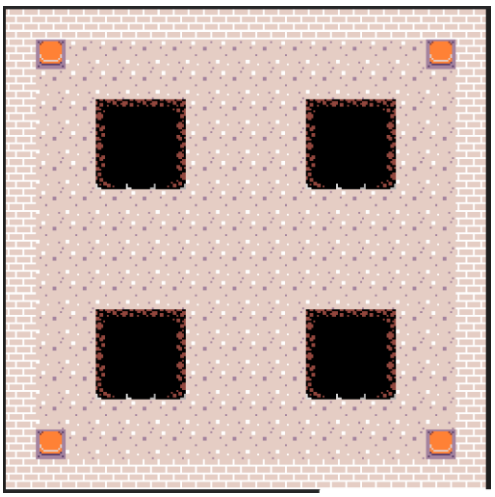
Ahora utilice la siguiente figura como guía para colocar los espacios denominados, fuentes de enemigos.



Con las acciones de diseño anteriores, el escenario o mundo de nuestro juego No Pacman, queda determinado tal como se indica a continuación.



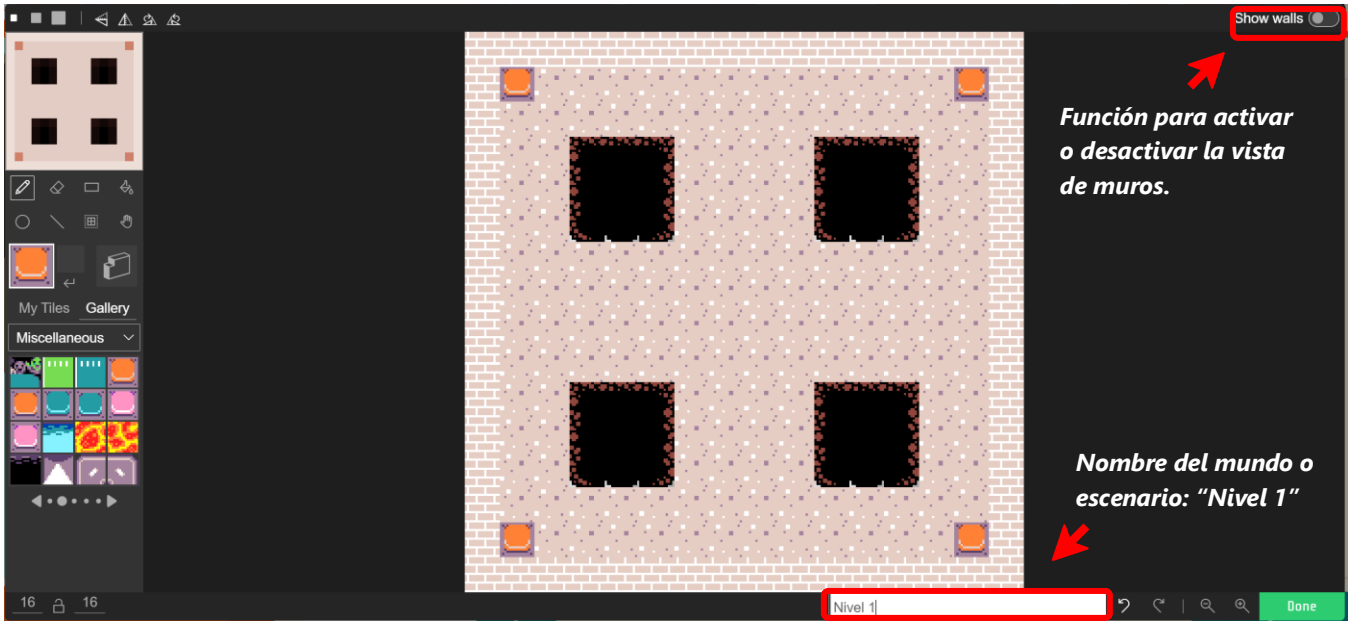
Escenario con las áreas delimitadas



Escenario con la apariencia final

Nota. Active y desactive la función "Walls" que se muestra en la esquina superior derecha de la ventana de edición de escenarios. Observe la figura que se muestra a continuación.

Paso 12. Guardar el mundo. Utilizando el cuadro texto al final de la ventana de edición para colocar el nombre; guardemos el escenario con el nombre "Nivel 1".



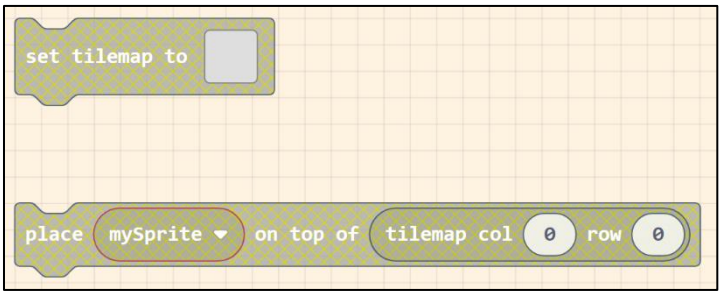
Presione ahora el botón "*Done*", y regrese al editor de programación.

Programando el escenario.

Ahora utilizaremos los bloques de la familia Scene (o escenario), los cuales nos permitirán integrar la escena al juego. Los bloques son:

- ***Set Tilemap to***: el cual permitirá integrar la escena del mosaico creado en el juego como su pantalla principal (o nivel 1).

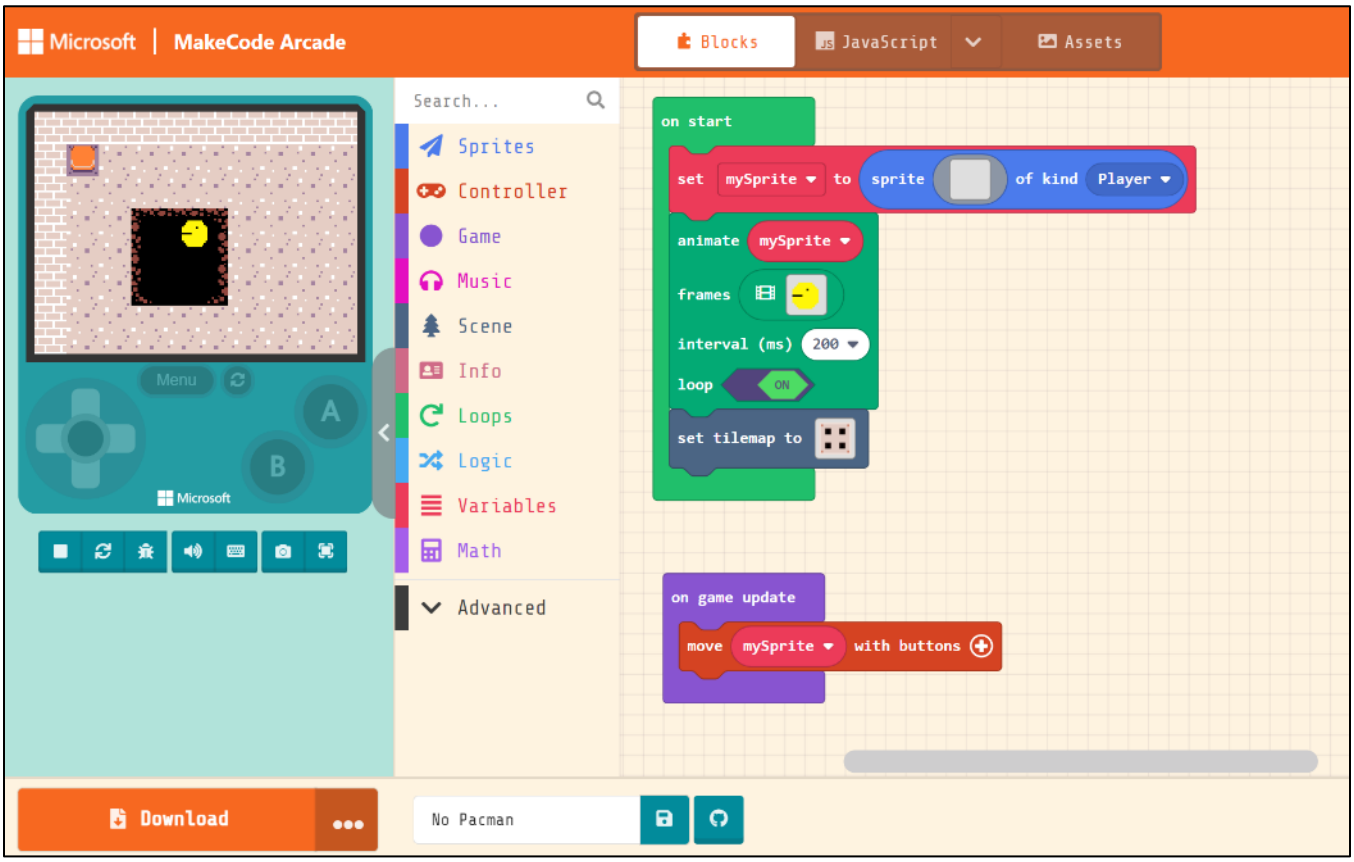
- ***Place mySprite on Top of Tilemap (x,y):*** Permitirá ubicar al personaje principal al centro de la pantalla.



Recordemos que la pantalla tiene una medida de 16 x 16 pixeles, por lo tanto, el entro se ubicará en la coordenada ***x=8; y=8***.

Paso 13. Integre la escena en la pantalla principal del juego utilizando el bloque "***Set tilemap to***", y colóquela en el bloque "***on start***" para que la pantalla se cargue al activar el juego.

Observe el resultado.



El personaje principal apareció en una posición no deseada. Por lo que es necesario reubicarlo.

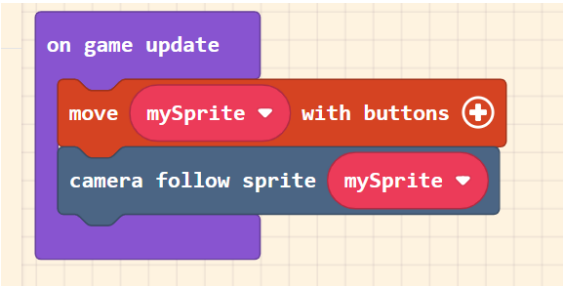
Paso 14. Coloque al personaje principal en el centro del mundo cuando se inicie el juego. Para ello utilice el siguiente bloque ***Place mySprite on Top of Tilemap (x,y). Observe el nuevo resultado.***

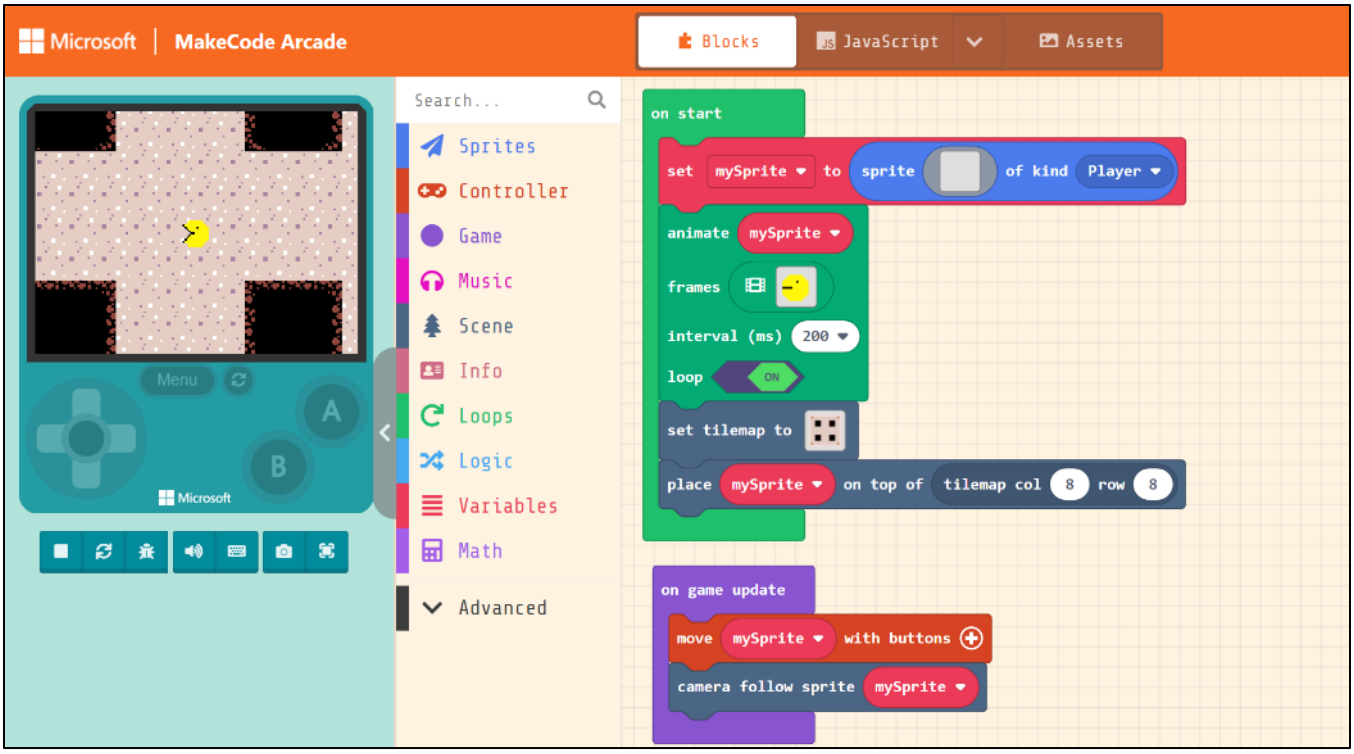
Ahora el personaje se encuentra en el centro de la pantalla, pero no lo podemos ver porque el enfoque de la cámara del juego únicamente nos muestra el sector del cuadrante superior izquierdo del mundo.

Es importante actualizar el funcionamiento de la cámara, y lograr ahora que la cámara enfoque los movimientos del personaje. Este efecto se logar con el bloque "Camara Follor MySprite" durante todo el proceso del juego.

Paso 14. Modifique el código del bloque "on game update" coloque el bloque 'camara follow Sprite". Esta modificación permitirá que por cada cambio que se realice en los parámetros del juego, constantemente la actualización permita que la cámara siga al personaje.

Observe los resultados.



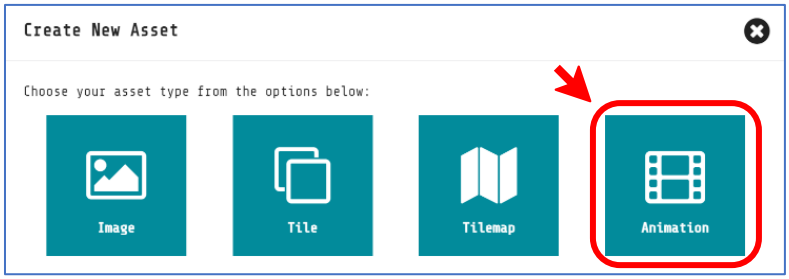


Ahora el personaje se encuentra en el centro de la pantalla del mundo y podemos contralar su desplazamiento durante todo el recorrido. Haga las pruebas de la etapa de programación hasta esta etapa.

Creando y Programando Enemigos.

Iniciaremos creando una nueva animación para el enemigo. **Nuevamente utilizaremos el menú Assets**. Para desplazaremos el seleccionador que se encuentra en la parte superior de la pantalla hacia la derecha para seleccionar la opción "Assets".

Paso 15. Selecciona el botón "*New Asset*". De las opciones que se muestran en la ventana emergente "Create New Asset", selecciona la opción "*Animation*".



Paso 16. Utilizando el editor de imágenes, construya las siguientes imágenes utilizando el editor de imágenes de 16 x 16 pixeles.

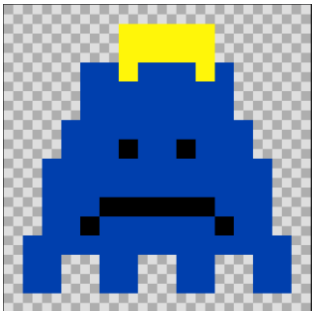


Imagen 1

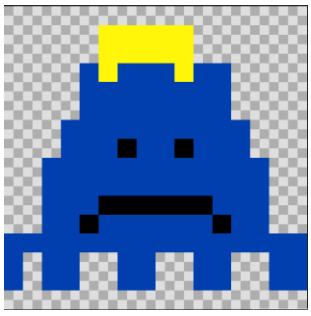


Imagen 2

Verifique su resultado en la siguiente figura.

Imagen 1

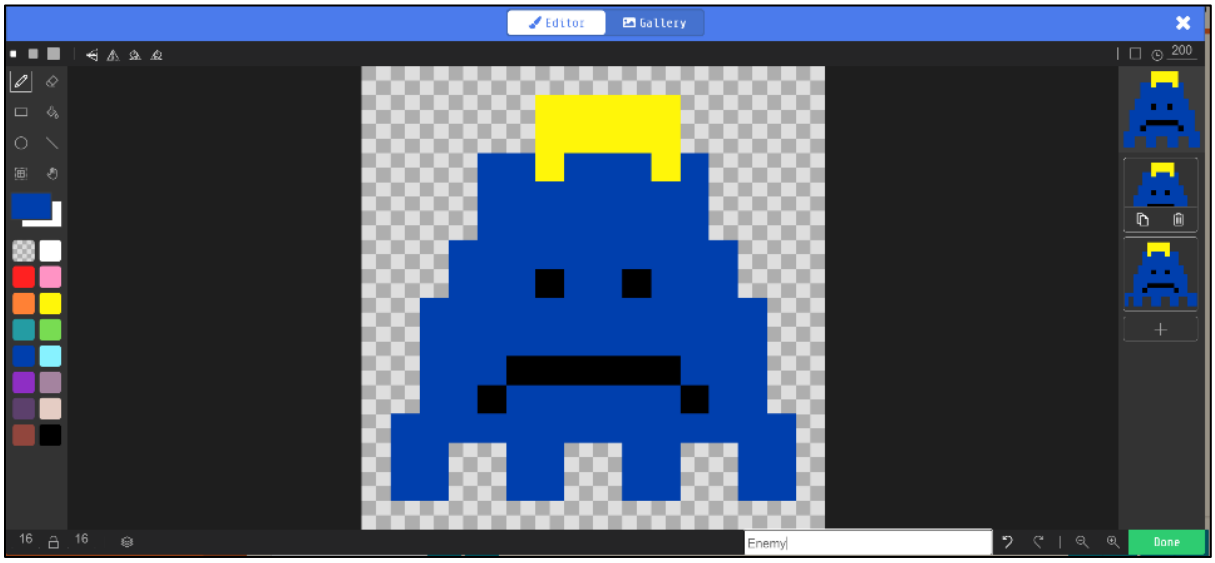
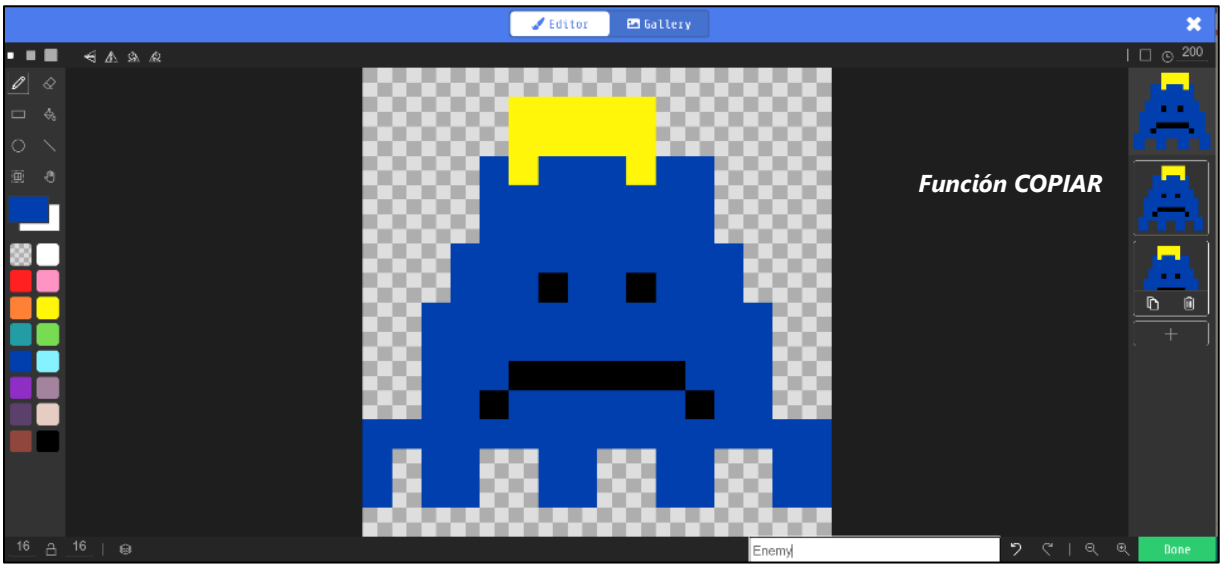


Imagen 2

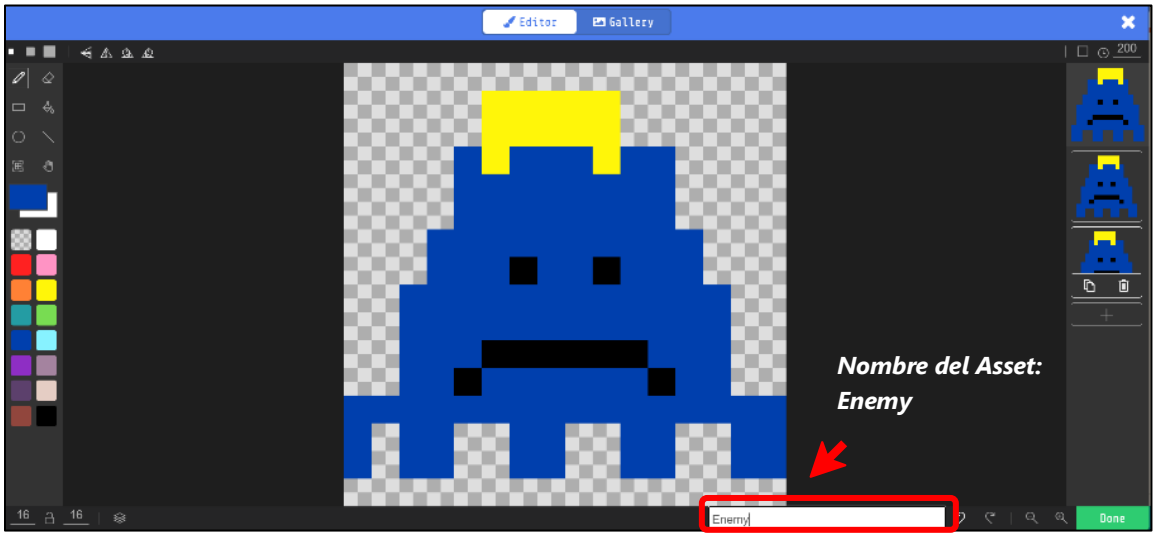


Verificando el funcionamiento de la animación.

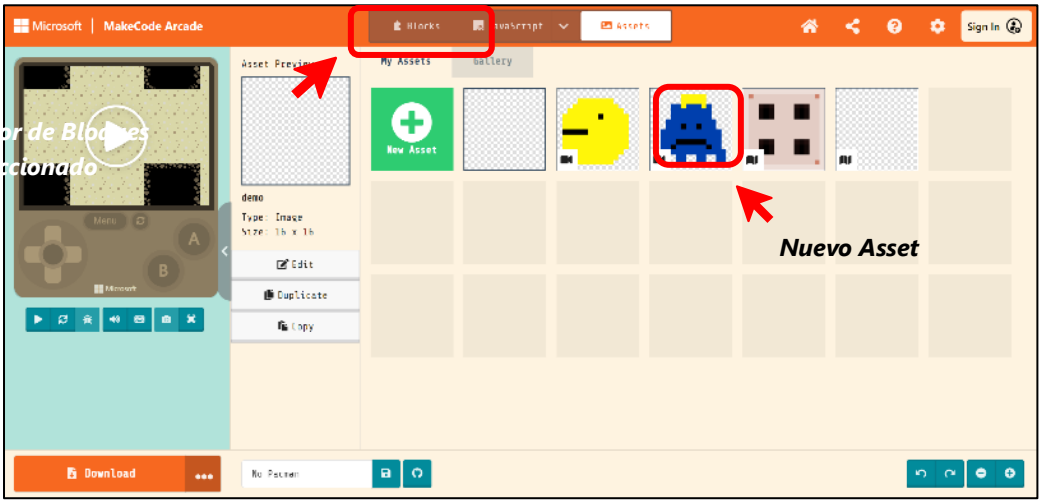
La animación estará compuesta por la superposición simultanea de ambas imágenes, logrando el efecto de desplazamiento del enemigo, estableciendo el tiempo de cambio entre las imágenes.

Presione nuevamente el botón "*play*", y compruebe el funcionamiento para un tiempo establecido de *200 milisegundos*. **Práctica**: modifique los diferentes valores de tiempo y observe los cambios.

Paso 17. Nombre a la animación. Ahora estamos listos, y debemos brindarle un nombre a la animación creada para representar al personaje de enemigo. Ubique la opción para colocar el nombre de la aplicación al final de la ventana de diseño, y coloque el nombre "***Enemy***", tal y como se indica en la figura a continuación:



Presione el botón "*Done*" y automáticamente guardamos la animación como parte de nuestra colección de animaciones o figuras.

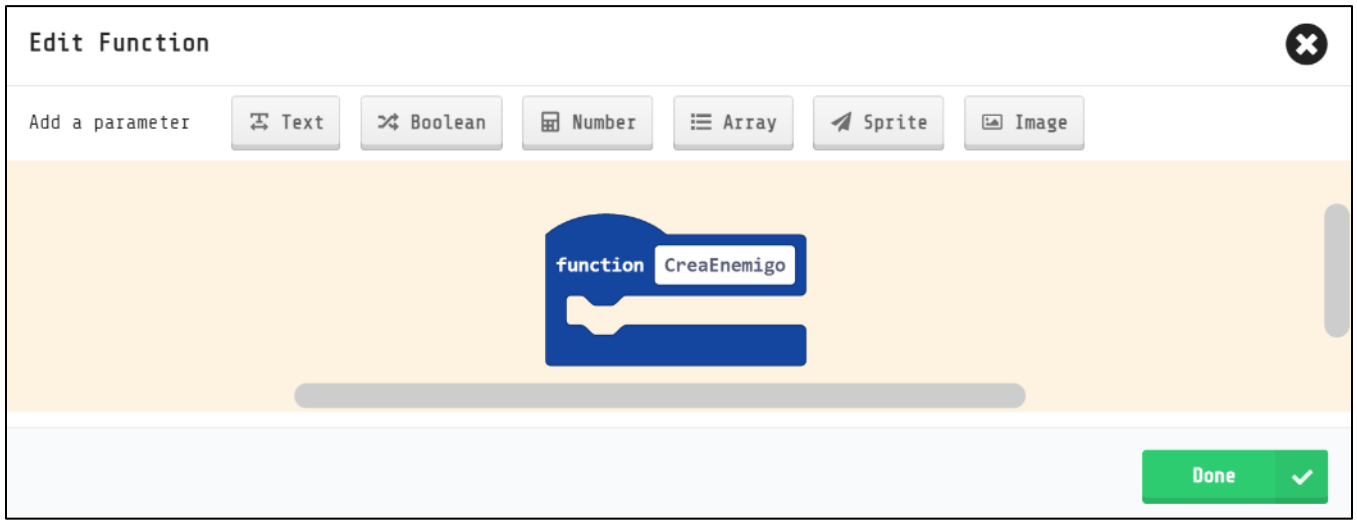


Observe ahora su colección de recursos (Assets), y verifique la nueva animación.

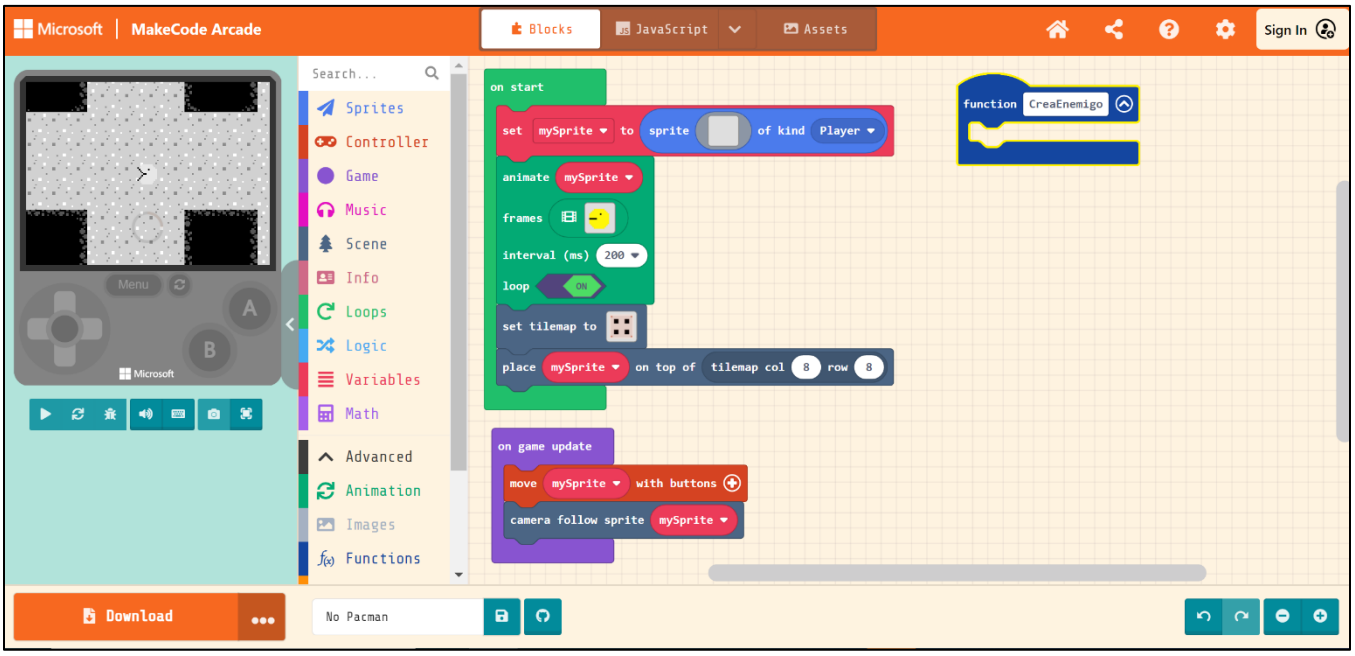
Ahora estamos listos para comenzar a programar. Para ello desplace nuevamente el seleccionador de la parte superior de la pantalla a la opción "Blocks", tal como se muestra en la figura, y regresaremos al editor de programación por bloques.

Creando Funciones.

Paso 18. Para el desarrollo de los enemigos trabajaremos con funciones. En este caso utilizaremos el bloque "*Functions*" y vamos a crear la función "***CrearEnemigo***".



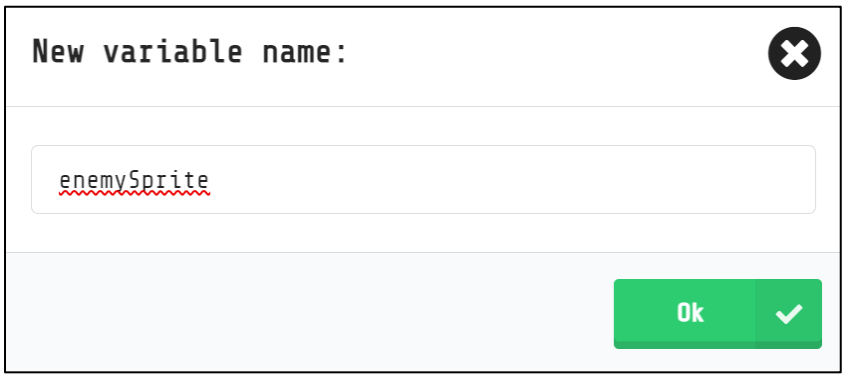
Haga clic en el botón "Done", y observe el nuevo bloque de la función "***CrearEnemigo***".



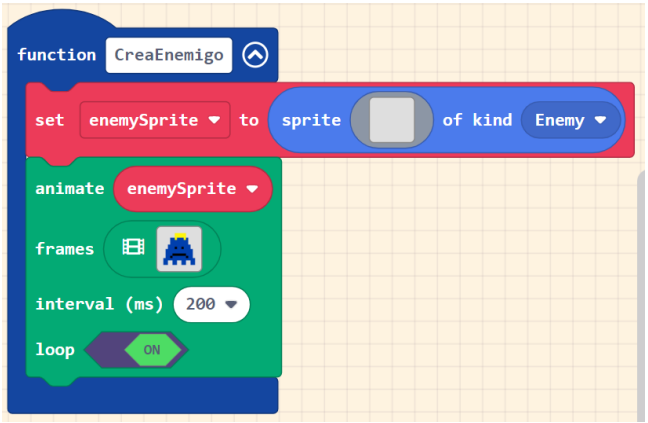
Ahora programaremos la función *CrearEnemigo*. Para ello utilizaremos los bloques ***Set mySprite to***, y el bloque ***Animante***.

Pero será importante crear una nueva variable, la cual representará al personaje que desempeña el role de enemigo. Esta variable tendrá el nombre de ***enemySprite***.

Paso 19. Selecciono el bloque variable, y defino la variable "enemySprite". Observe la siguiente ventana:



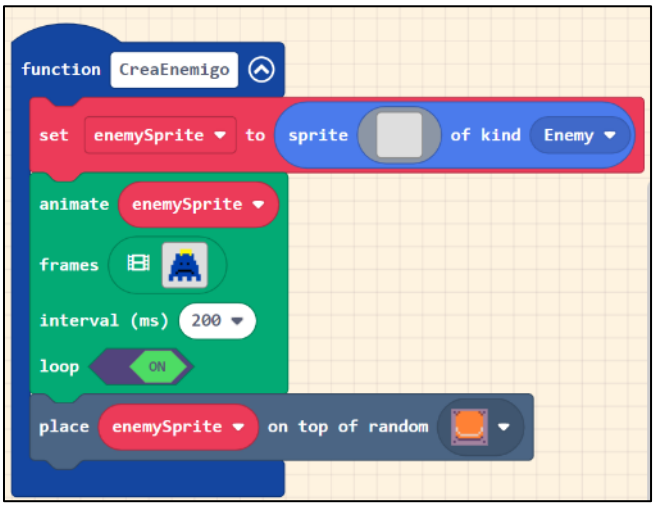
Paso 20. Ahora continue con la programación de la función nueva para la creación de enemigos. Escriba y verifique el siguiente código complementario.



Observe que el bloque, set mySprite, cambio por la nueva variable set enemySprite. Esta variable sigue siendo un Sprite, pero del tipo enemigo.

El bloque de animación ahora hace referencia a la animación ***enemy***, con una duración de 200 milisegundo y con el bucle de repetición indefinida activado.

Paso 21. **Ubicación de los enemigos en la pantalla.**

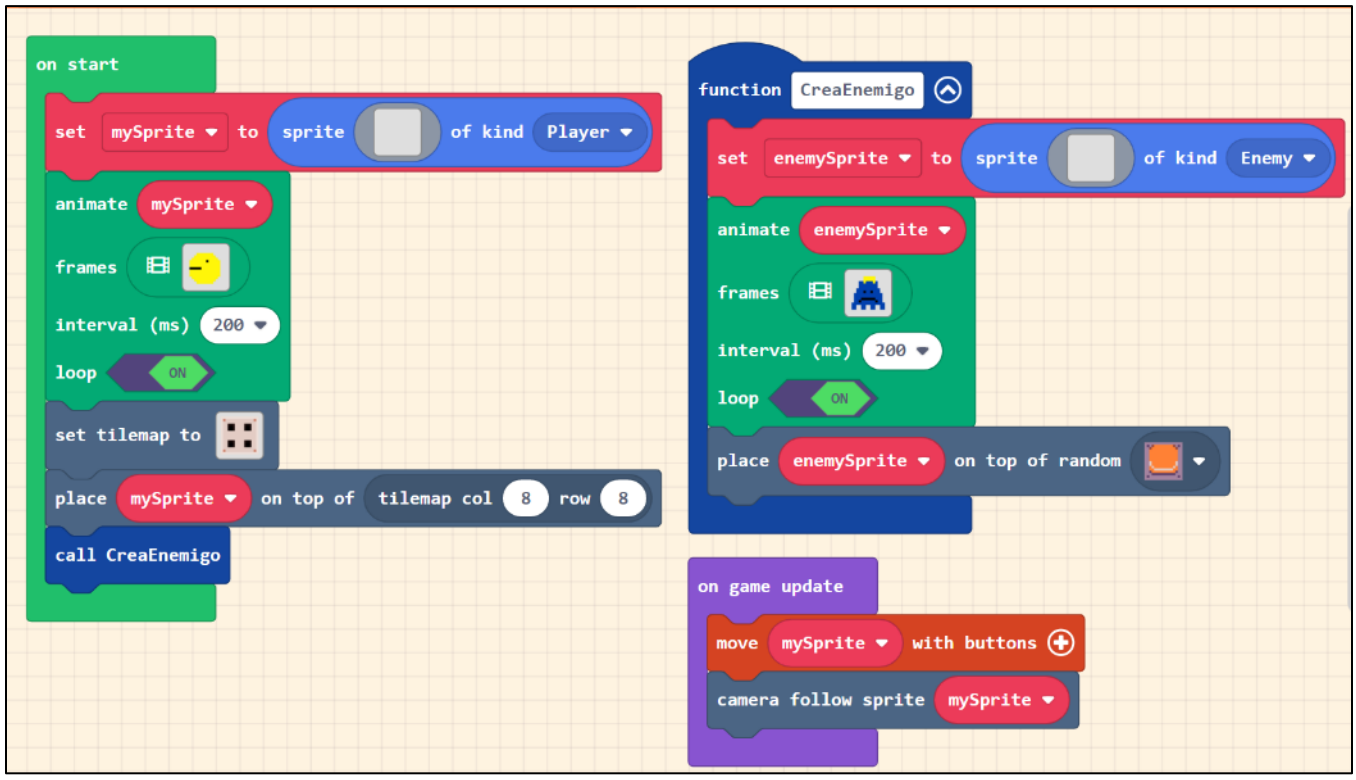


Para ubicar a los enemigos en las fuentes de enemigos, utilizaremos el bloque set ***place mySprite on top of ramdom***.

Modifique los parámetros del bloque por place ***enemySprite***, seleccione el mosaico correspondiente a lugar que denominamos fuente de enemigos, y verifique el siguiente código.

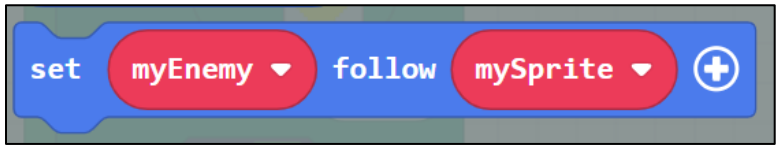
Paso 22. **Convocar la función CrearEnemigo.**

Ahora que la función esta creada y programada, es importante activar la función o convocarla desde el momento que inicia el juego. Esto lo realizaremos mediante el bloque ***call function*** o ***call CrearEnemigo***, y lo colocaremos en el bloque ***on start.*** Escriba el nuevo código para el bloque ***on start*** y verifque el resultado.



Ahora podrá moverse a lo largo del mundo y observar que el enemigo aparece una sola vez en una de las esquinas del mundo.

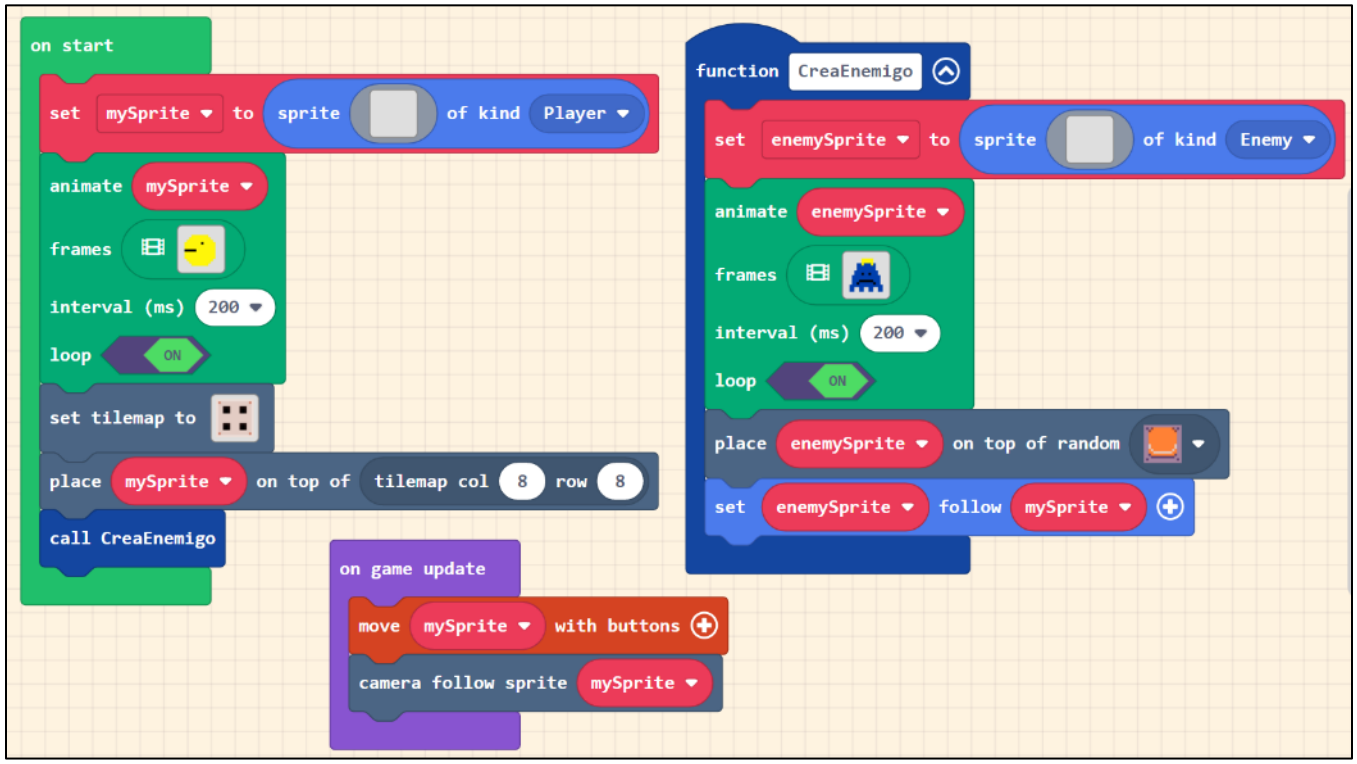
Debemos lograr ahora que el enemigo siga al personaje, y para ello utilizaremos el bloque de la familia Sprite, que se denomina set ***myEnemy to follow mySprite***.



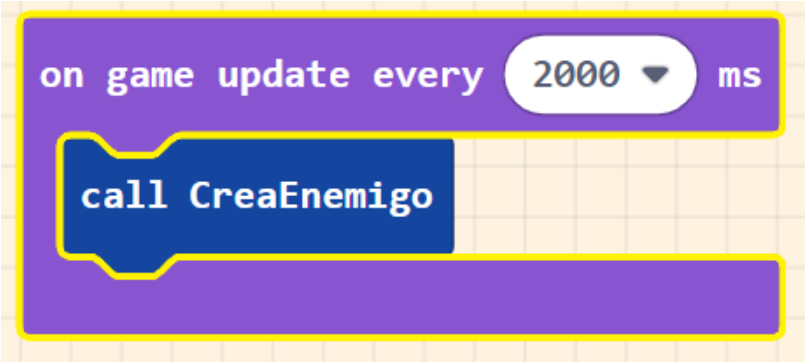
Este bloque debe ser modificado, ya que el enemigo se representa por la variable ***enemySprite***. El personaje no se modifica, en este caso esta representado por la variable ***mySprite***.

El bloque debe ser incluido como un bloque de la función ***CrearEnemigo***, de tal manera que cada enemigo al ser creado tenga la propiedad de seguir al personaje.

Escriba el nuevo programa y verifique el resultado.



Ahora que la función ***CrearEnemigo*** está completa, vamos a desarrollar un programa que permita la aparición de múltiples enemigos.



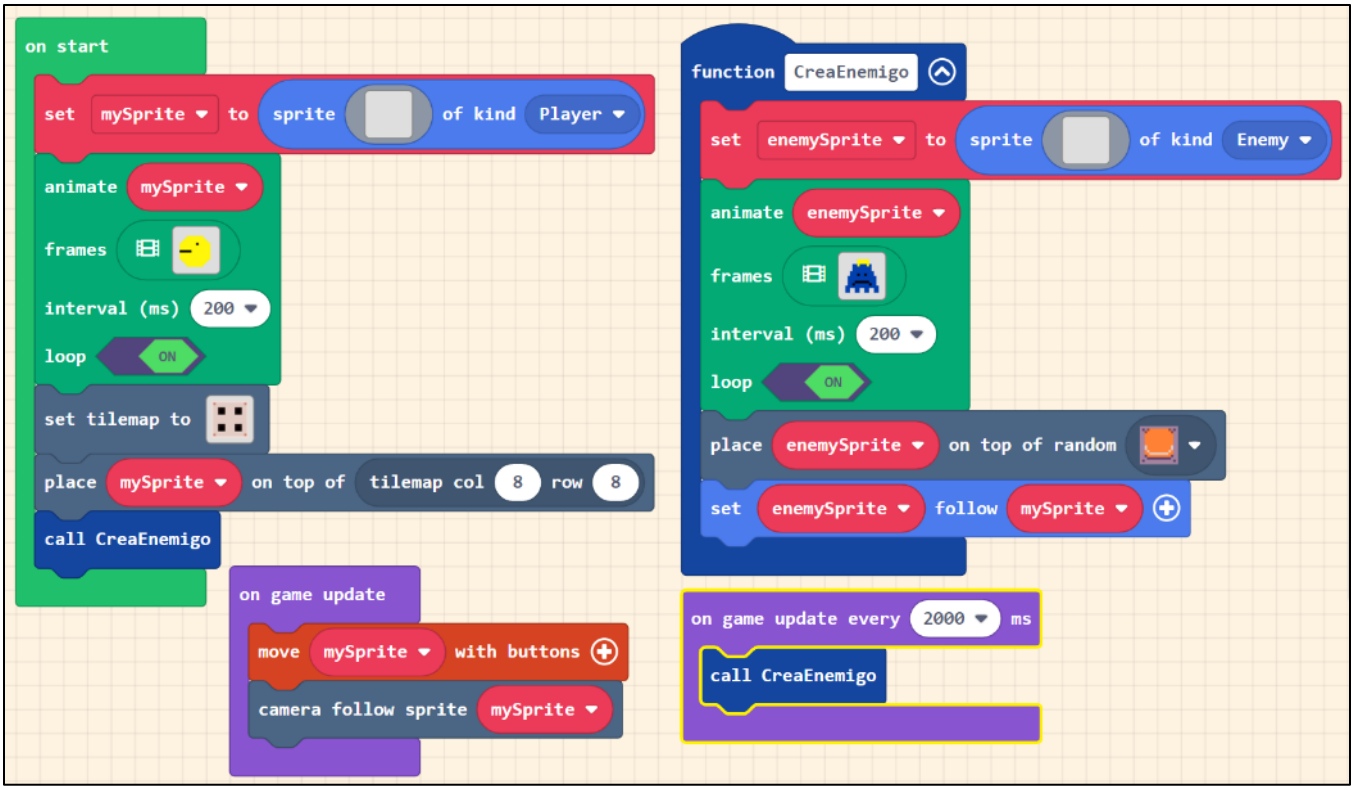
Para ello utilizaremos el bloque de actualización del juego cada cierto periodo de tiempo, de la categoría Game, denominado ***on game update every XXX ms***.

Este bloque permitirá la creación de enemigos cada 2 segundos, lo cual lograremos colocando el bloque de activación o convocatoria de función ***CreaEnemigo*** como parte de su argumento, tal como se muestra en la figura.

Con este proceso de programación, hemos terminado la etapa de los personajes, enemigos, y escenarios de un juego.

Lo que sigue ahora es determinar las reglas del juego y su proceso de puntuación, de tal manera que determinemos el momento o condición a cumplir para ganar o perder el juego.

Escriba el nuevo código de programación y verifique el resultado.



Creando y programando las reglas del juego

Regla para ganar el juego. Recordemos que, para ganar el juego, el jugador no debe dejarse atrapar o tocar por los enemigos durante un periodo determinado de tiempo.

El jugador tendrá *un máximo de 3 intentos*, y acumulará un número de puntos determinado igual al número de milisegundos que dura su participación.

En este caso entre menor sea el tiempo que el jugador debe evitar se atrapado, menor será la dificultad de ganar el juego.

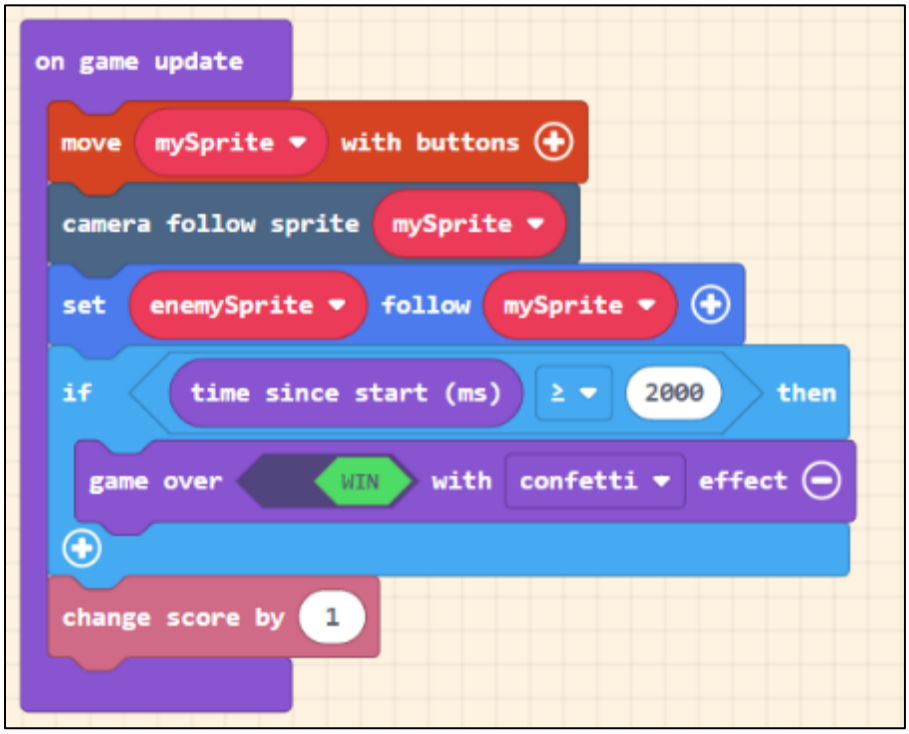
Regla para el ganador. *Utilizaremos un bloque condicional mediante el cual se cumpla lo siguiente:*

El jugador resultará victorioso, siempre y cuando el tiempo transcurrido sea mayor a 2 segundos (20,000 milisegundos), y no haya sido atrapado por algún enemigo.

Comprenderemos que sí un enemigo toca o se sobrepone sobre el jugador, entonces el jugador pierde el juego o pierde una vida.

Utilizando el bloque condicional, modifique el programa del bloque ***on game update***, con el propósito de programa la regla para acumular los puntos y establecer al ganador.

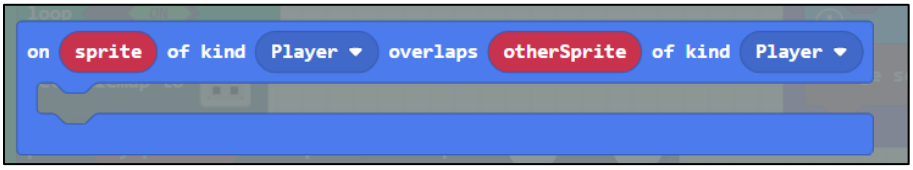
Escriba el complemento del siguiente programa y verifique su funcionamiento.



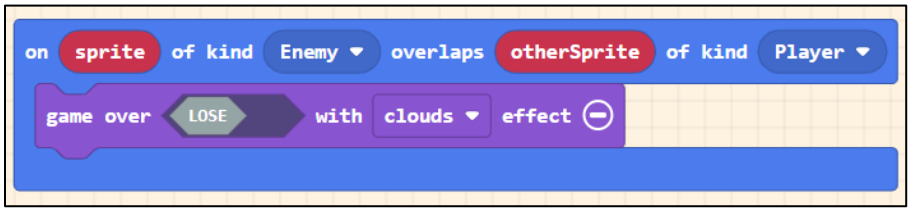
Regla para perder una vida o el juego. *Utilizaremos un bloque condicional mediante el cual se cumpla la condición de sobre posición del enemigo sobre el jugador llamado OVERLAP.*

El jugador perderá una vida o perderá el juego, siempre y cuando antes que termine el tiempo establecido a 2 segundos (20,000 milisegundos), haya sido atrapado por algún enemigo.

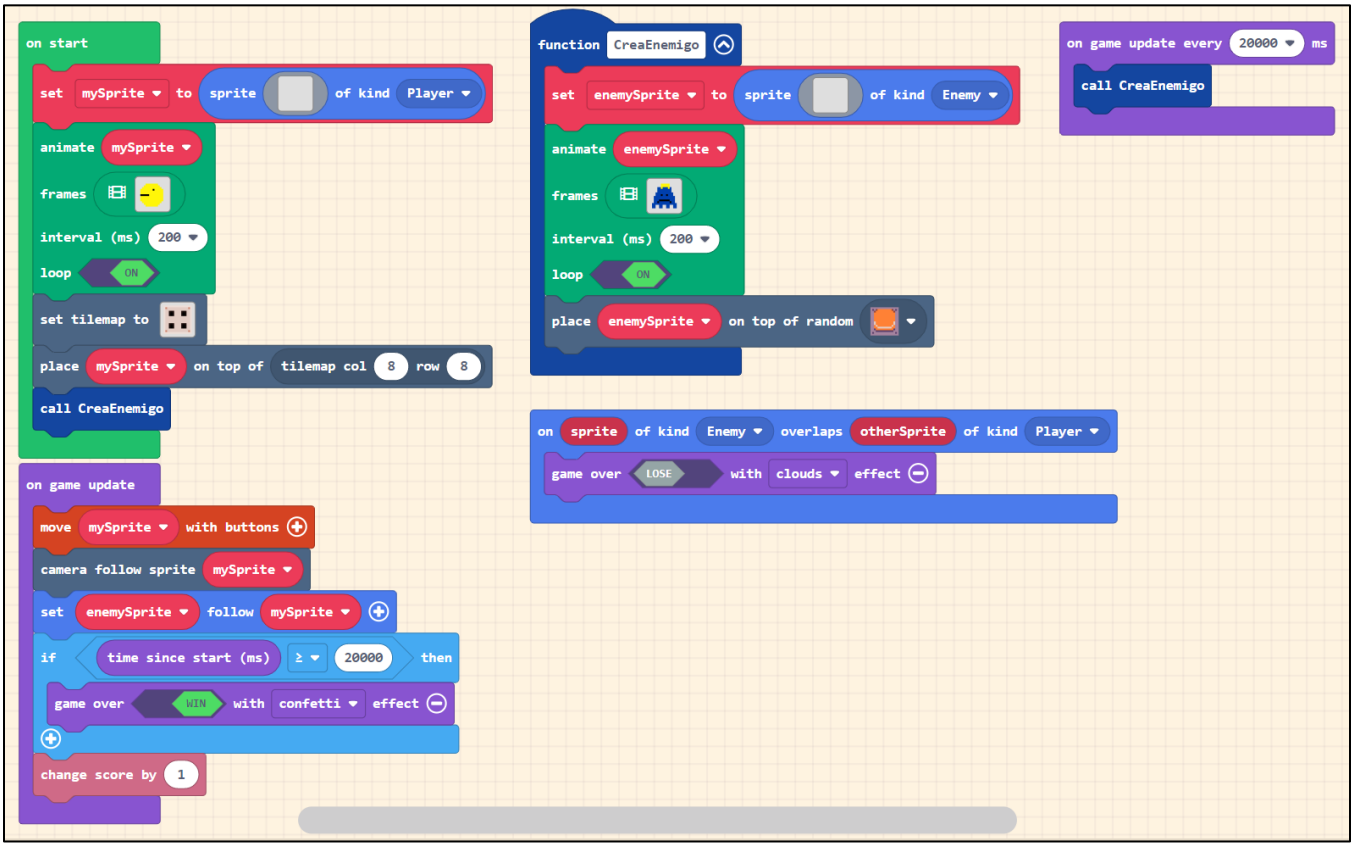
Para este fin utilizaremos el bloque especial OVERLAP, que establece la condición siguiente: "Sí un Sprite del tipo enemigo OVERLAP/SE SOBREPONE/ O SE SOBREPOSICIONA sobre un Sprite de tipo jugador, entonces se aplica una regla especifica.



En este caso cambiaremos el primer argumento de Player/Jugador, por Enemy/Enemigo, y condicionaremos a que finalice el juego una vez el enemigo haya alcanzado al jugador. Escriba el siguiente código y verifique su funcionamiento.



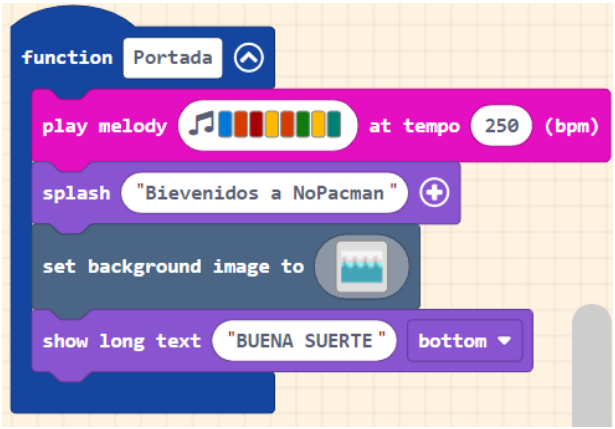
Verifique la escritura completa del nuevo programa y observe los resultados.



Pantalla de inicio.

Agregaremos una función que permita crear una pantalla de inicio o bienvenida al juego. La función se llamará ***Portada***.

Realice el siguiente código poniendo en practica todos los conocimientos adquiridos.



Verifique el funcionamiento del juego completo.

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)

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)