

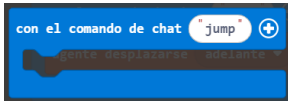

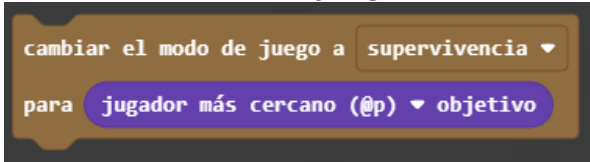

# Práctica guiada 2. Parkour


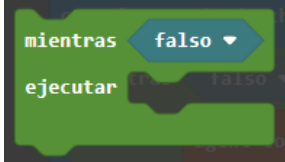

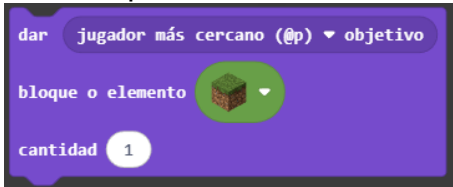



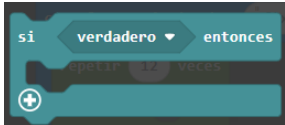

En esta práctica aprenderemos a diseñar y programar un minijuego de tipo Parkour dentro de Minecraft Education. A lo largo de esta lección, programaremos un circuito con plataformas, añadiendo un inicio y un final, crearemos un sistema de temporización y evitaremos que los jugadores hagan trampas. ¡Empezamos!

## Mecánicas de Minecraft

Mecánicas	Para ordenador
Abrir el editor de código	C
Abrir chat	T
Abrir el menú de pausa; cerrar los menús de juego	ESC
Minar un bloque	Clic Izquierdo
Interactuar con botones y con personajes	Clic derecho
Saltar	Barra espaciadora
Movimiento del jugador (Adelante/Atrás/Izquierda/Derecha)	W/S/A/D
Dirección del jugador	Movimiento del ratón

## Bloques para esta práctica

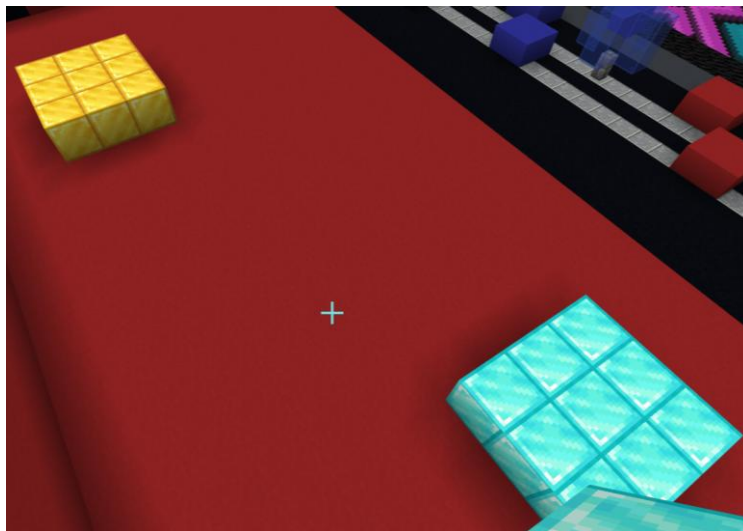
Con el comando de chat 	Evento que se activa al escribir una palabra en concreto en el chat.
Decir 	Pone el mensaje especificado en el chat de juego.
Cambiar el modo de juego 	Cambia el modo de juego entre supervivencia, creativo y aventura.
Ejecutar 	Ejecuta comandos de chat a través de código.

<p>Repetir</p> 	<p>Ejecuta la parte del programa contenida el número de veces que se le indique.</p>
<p>Mientras</p> 	<p>Mientras que no ocurra la condición, el bucle se seguirá ejecutando, una vez que ocurra, parará.</p>
<p>Agente teleportar</p> 	<p>Mueve al agente al lugar en el que se encuentra el jugador.</p>
<p>Dar bloque o elemento</p> 	<p>Coloca el bloque o elemento seleccionado, la cantidad de veces seleccionada al inventario del jugador escogido.</p>
<p>Agente inspeccionar</p> 	<p>Inspecciona un bloque en la dirección indicada, devuelve el tipo de bloque.</p>
<p>Condicional</p> 	<p>Devuelve verdadero si los dos valores de la condición son iguales.</p>
<p>Valor del bloque</p> 	<p>Devuelve un tipo concreto de bloque de Minecraft.</p>
<p>Si</p> 	<p>Si la condición se cumple, ejecuta el programa que contiene.</p>
<p>Pausa</p> 	<p>Para la actividad del Código durante los (ms) escogidos</p>

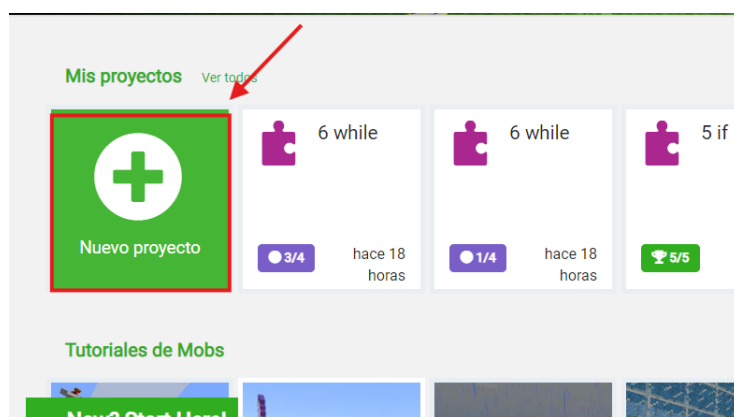
## Parte 1. Preparando el escenario

Cuando programamos, debemos recordar que vamos a estar constantemente probando nuestro código para saber si funciona correctamente, por lo que, debemos comenzar con un mapa sencillo de parkour y cuando ya tengamos todo bien programado, podremos incrementar la dificultad.

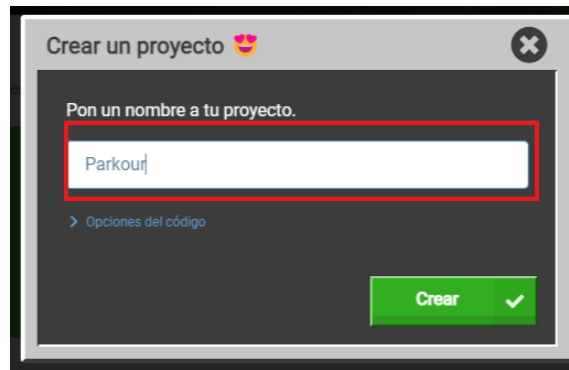
Por tanto, para poder comenzar, vamos a hacer un mapa únicamente con aquello que necesitaremos sí o sí. Necesitaremos dos áreas de diferentes bloques, una de inicio y otra de final, es importante tener dos zonas con bloques diferenciados para que el agente pueda distinguir cual es cual. Podemos comenzar con dos cuadrados planos de 3 por 3, creando uno de diamante para la zona de inicio y otro de oro para la meta, recordando dejar una distancia entre las plataformas, para poder testear bien nuestro juego. Como se muestra en la siguiente imagen.



Para este juego vamos a usar nuestro Agente como árbitro, pero para poder lograrlo, el primer paso será crear un nuevo proyecto en MakeCode.



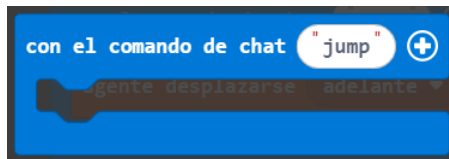
Recordad darle un buen nombre que podáis asociar con el juego que vais a crear, ya que en la próxima clase deberéis volver a usarlo y con un nombre concreto, será más fácil de localizar.



## Parte 2. Comenzando con el código

Para comenzar, vamos a hacer es programar las condiciones en las que se deberá encontrar el jugador cuando el juego comience, lo que incluye, comenzar la partida, indicar al jugador que comienza la partida, seleccionar el modo de juego del jugador y vaciar su inventario.

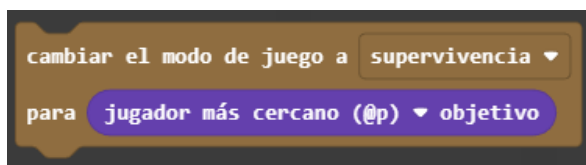
Lo primero, será crear un comando que indique a nuestro programa, que queremos comenzar a jugar. Para ello utilizaremos el bloque de eventos “Con el comando de chat ...”, y lo modificaremos de tal forma que, cuando pongamos en el chat “jugar”, el juego comience.



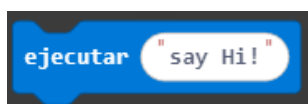
Además, una vez se vaya a activar el juego, queremos asegurarnos de que el jugador sabe que va a comenzar, por lo que, cuando activemos la partida, podemos mandarle al jugador un mensaje que indique que va a comenzar, como, por ejemplo, “preparado, listo, ¡ya!”, utilizando el bloque decir.



Ahora, queremos asegurarnos de que el jugador no está en modo creativo, para que no pueda hacer trampas, para ello utilizaremos el bloque “cambiar el modo de juego”, configurándolo para que establezca el modo supervivencia para el jugador más cercano.



Por último, queremos eliminar cualquier objeto del inventario del jugador, para ello, puesto que no tenemos un bloque concreto, podemos utilizar el bloque “ejecutar...”.



En Minecraft hay algunas ordenes que se pueden activar a través del chat utilizando “/” y la palabra adecuada, el bloque ejecutar activa ese tipo de órdenes si las escribimos en el bloque (sin el “/”). Para este caso, existe el comando para el chat “/clear” que vacía el inventario, por lo que, si ponemos “clear” en nuestro bloque, se ejecutará la orden de vaciar inventario, tal y como nosotros queremos que ocurra.

Con esto, ya podemos hacer la primera prueba de nuestro código, lanza el código, clicando en el botón verde de comenzar y comprueba que todo lo realizado anteriormente funciona.

## Parte 3. Puntuaciones en el juego

Ahora, deberemos crear una forma de puntuar mientras que el jugador está en partida. Para poder ir incrementando el valor de puntuación vamos a necesitar utilizar un bucle, hasta ahora hemos visto dos de ellos. El primero se repite un número concreto de veces, mientras que el segundo se repite hasta que se cumpla un objetivo.

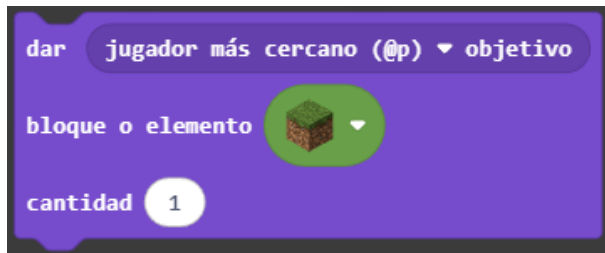


Teniendo en cuenta que nuestro bucle debe ir contando puntos hasta que el jugador llegue a la meta, decide cuál de los dos bucles anteriores es el indicado. Además, este bloque deberá activarse después de todo lo creado en la parte anterior, pero solamente cuando esté la partida activa.

Como vamos a estar utilizando nuestro agente como árbitro, queremos que este esté cerca del jugador constantemente, para ello vamos a utilizar el bloque que teletransporta al agente al lado del jugador. Ten en cuenta, para la colocación de este bloque, que el agente deberá estar constantemente teletransportándose al lado del jugador hasta el final de la partida.



Ahora deberemos buscar una forma de puntuar, por lo que necesitaremos algo con lo que se pueda ir sumando puntos. Una forma fácil de realizar este paso es añadir al inventario del jugador un mismo objeto cada cierto tiempo, de forma que, cuanto más tiempo tarde en terminar el juego, más puntos acumulados tendrá al finalizar. Para esto vamos a utilizar el bloque dar.



Además, deberemos tener en cuenta que, como no queremos que el jugador pueda hacer trampas con el objeto que le damos, deberemos cambiar este por algo que no le pueda resultar útil, como, por ejemplo, una antorcha.

Al finalizar esta parte deberíamos tener un bucle que, cada cierto tiempo, teletransporte al agente al lado del jugador y añada una antorcha al inventario de este. A continuación, crearemos el condicional que hará que el bucle finalice al llegar a la meta.

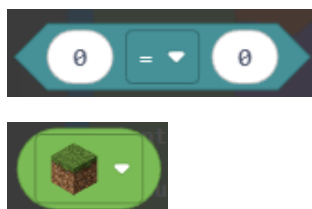
## Parte 4. Utilizando condicionales

Ahora mismo, el bucle creado en el apartado anterior se ejecuta sin pausa, pero nosotros queremos que pare cuando el jugador llegue a la meta. Para poder hacer esto, debemos poder detectar que el jugador ha llegado al final, es decir, detectar que ha llegado a los bloques de oro del final. No podemos detectar esto con ningún código del jugador, pero para eso nos hemos asegurado de que el agente le sigue constantemente, puesto que con el agente podemos hacerlo con “agente inspeccionar...”.



Este código nos devuelve que tipo de bloque tiene el agente en determinada posición, y que, para nuestro caso, será abajo, puesto que la meta es una plataforma sobre la que estará el jugador.

Aún así, aunque ya sabemos como inspeccionar sobre que bloque se encuentra nuestro jugador, con ayuda del agente, aún no tenemos forma de parar el contador de puntuación. Para ello, deberemos decirle a nuestro bucle anterior, que esté activo únicamente cuando el bloque que haya debajo del agente sea diferente (≠) de un bloque oro, para ello necesitaréis los siguientes bloques de código, con alguna variación.



Recordad que se debe colocar el resultado del condicional dentro del bucle, de forma que el bucle esté activo hasta que el agente esté encima de un bloque dorado.

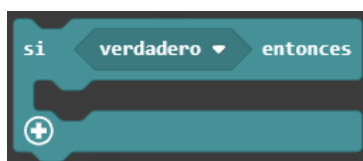
Una vez el bucle termina, podemos añadir un bloque de “decir”, como al comienzo, pero esta vez para notificar al jugador de que ha terminado la partida.

Ya podemos iniciar el código para comprobar que se para al subir a la plataforma dorada.

## Parte 5. Iniciando condicionales

Ahora que ya sabemos como identificar por código si el jugador está en la plataforma de meta, y podemos parar el contador, pero cuando intentamos jugar de nuevo, no funciona. Y esto es debido a que el agente se queda quieto encima del bloque dorado, por lo que el bucle nunca podrá iniciarse. Para poder solucionar esto debemos asegurarnos de que el agente se encuentra al lado del jugador cuando este quiere iniciar la partida, para ello deberemos teletransportar al agente al lado del jugador nada más iniciar la partida, sin modificar el teletransporte que ya se encuentra dentro del bucle.

Igualmente, el jugador podría hacer trampas y comenzar la partida a mitad circuito de parkour, por lo que debemos asegurarnos de que el jugador solo puede iniciar partida si se encuentra en la plataforma de salida. Deberemos comprobar esto justo después de mandar al agente junto al jugador por primera vez, utilizando el bloque condicional.



Dentro de este bloque deberemos colocar el resto del código, a excepción del primer teletransporte del agente, y, siguiendo la comparación del apartado anterior, pero cambiando los parámetros necesarios, se deberá conseguir que el código de dentro del condicional “si ... entonces” solo se active en caso de que el agente esté encima de un bloque de diamante. Recuerda que puedes duplicar bloques que ya tienes creados haciendo clic derecho sobre ellos, en lugar de tener que ir a buscarlos de nuevo.

Una vez lo tengas comprueba que funciona iniciando el código.

## Parte 6. Últimos retoques

Ya tenemos el juego casi completo, pero por el modo en que está programado, el jugador aún puede hacer alguna trampa.

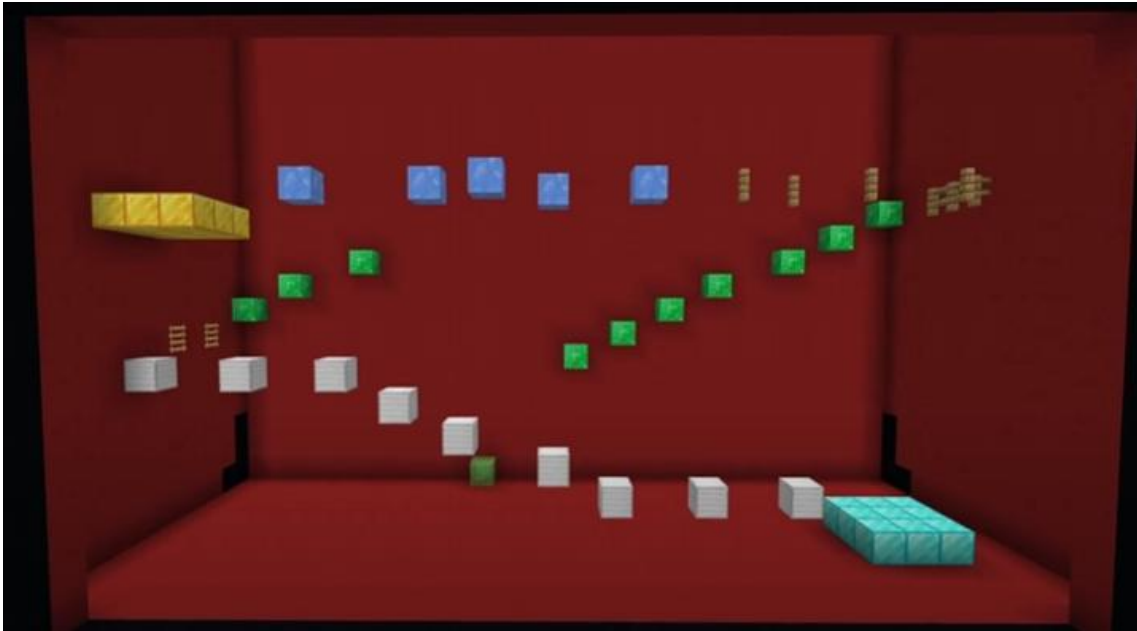
Una de ellas es que, aunque comprobamos que el jugador esté en modo supervivencia antes de comenzar a jugar, nada impide al jugador, pausar el juego y cambiarse a modo creativo para volar hasta la plataforma final, por tanto, vamos a solucionar este problema. En lugar de cambiar el modo de juego al inicio, podemos mover el código dentro del bucle de forma que constantemente se cambie a modo supervivencia.

Otra forma de hacer trampas sería ir soltando por el camino las antorchas que contabilizan la puntuación para que, al llegar a la meta, tenga menos de las que ha recibido, pero eso no podremos cambiarlo hasta alguna práctica más adelante.

Por el momento, podemos centrarnos en arreglar un último detalle, y es que las antorchas se acumulan de forma muy rápida así que vamos a cambiar esto para que las antorchas se acumulen aproximadamente cada segundo. Para ello deberemos pausar el bucle durante 500ms, colocando el siguiente bloque de código al final de este.



Y ya tendríamos finalizada la programación de nuestro juego de Parkour, ahora se puede modificar y diseñar el circuito que más os guste, como el que se muestra a continuación, por ejemplo.



¡Sed creativos!

## Entrega

Una vez hayas finalizado el circuito, asegúrate de:

1. Guardar el mundo correctamente.
2. Guarda el proyecto y realiza una captura de pantalla del código y del circuito



3. Presenta de forma oral el proyecto o haz un documento con las capturas y la explicación del código realizado.
4. Sube el archivo del proyecto y el documento, si es el caso, a Moodle.

**ANTES DE LA PRÓXIMA PRÁCTICA.**



