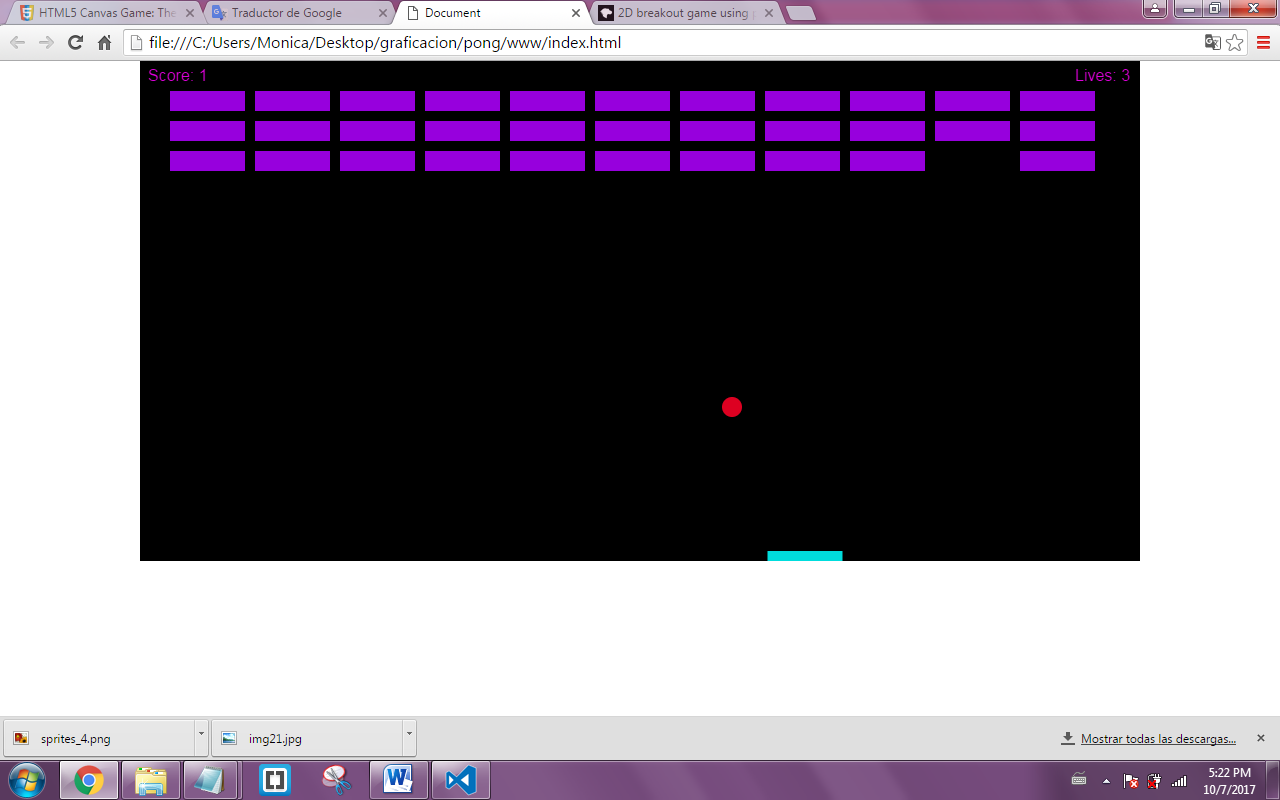
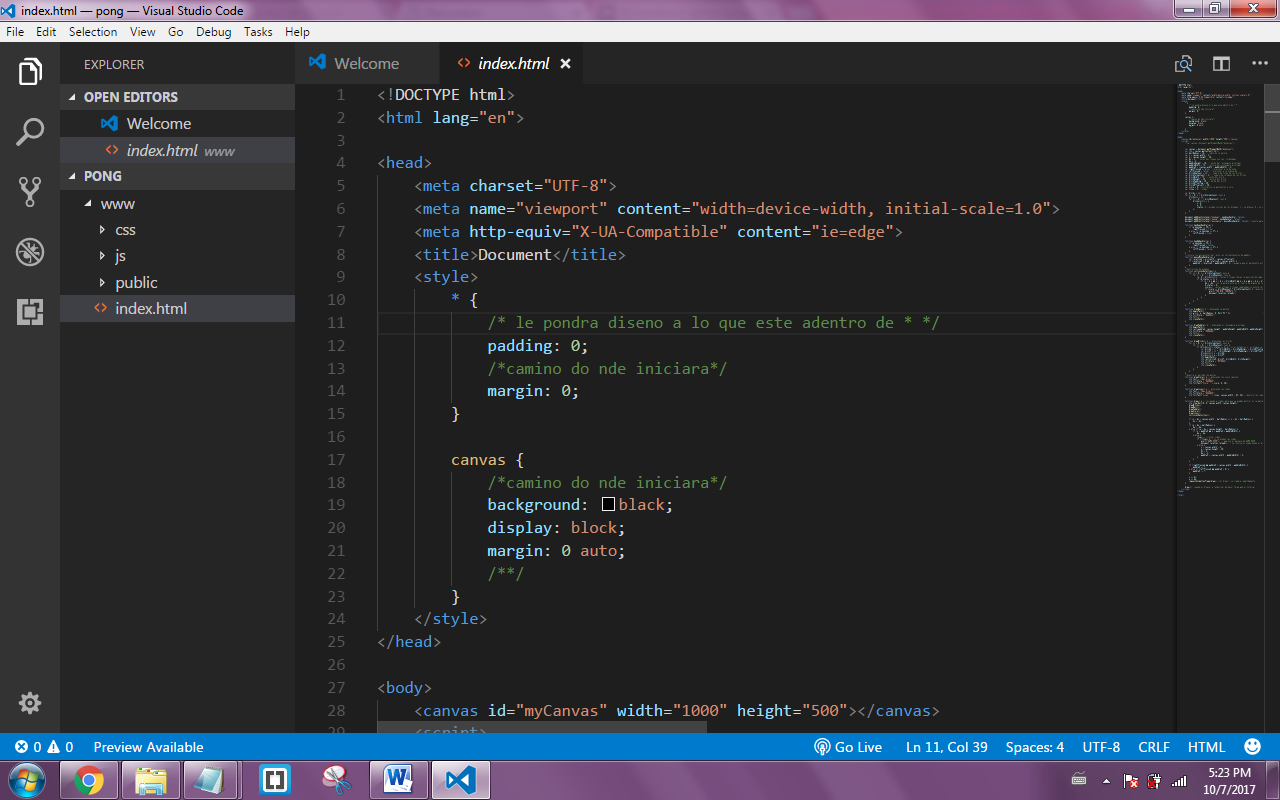
**Documentación del juego Pong**

**Usando  JavaScript pura:**

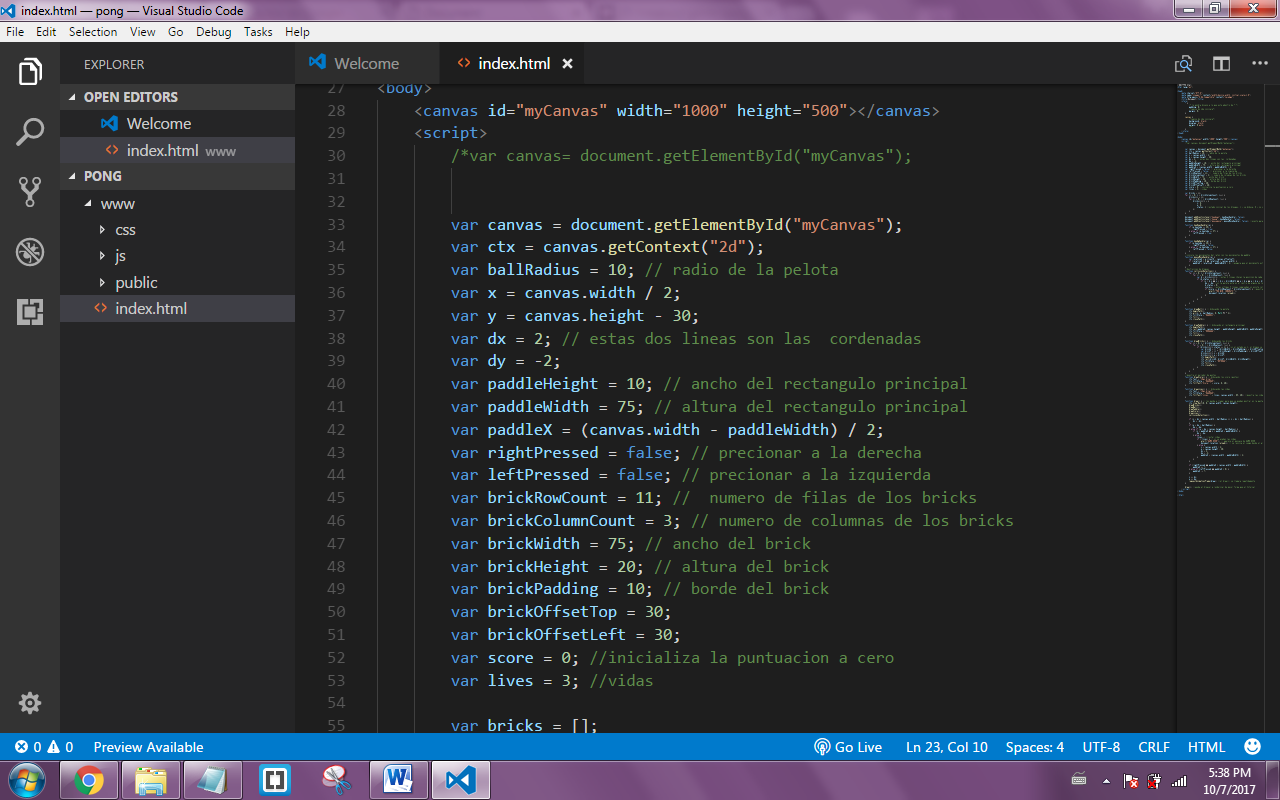


**Codigo:**



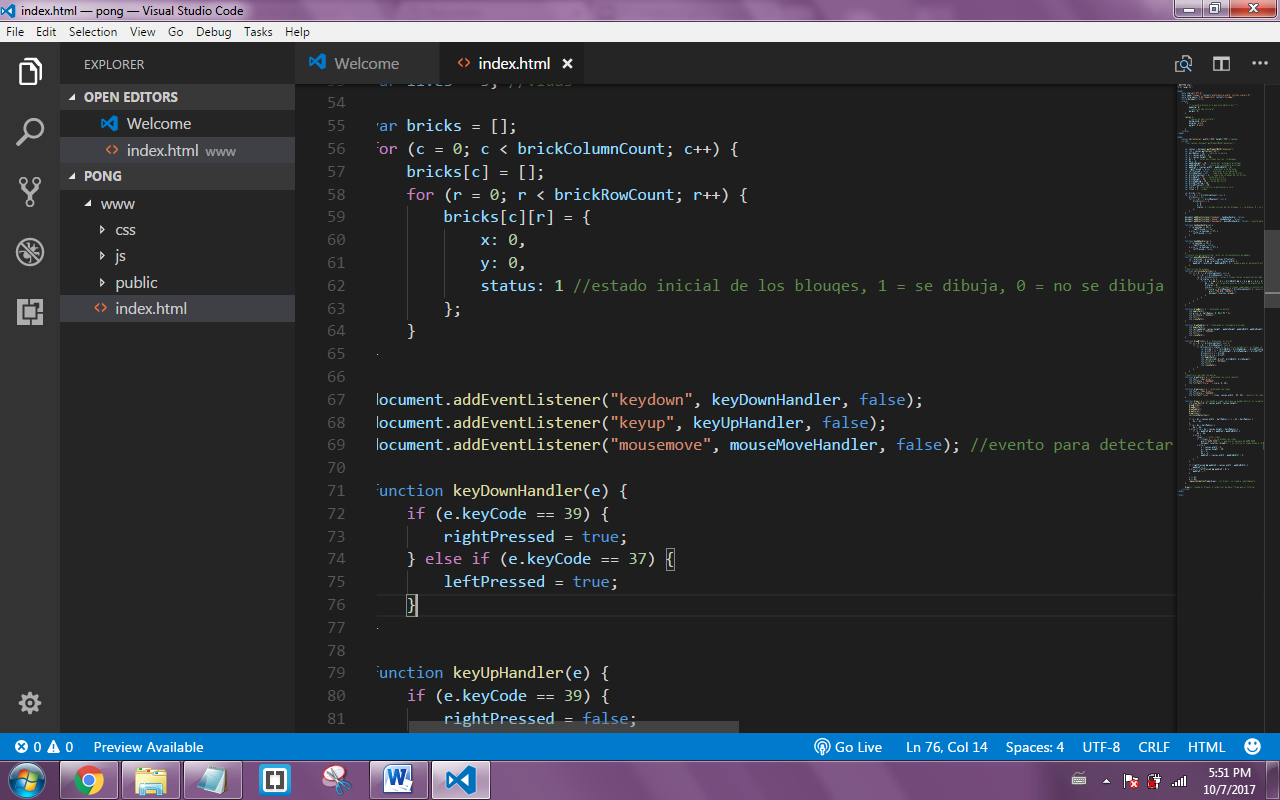
En el head tenemos el titulo del proyecto y los datos de la pagina del juego en syile le pusimos el diseño a lo que esta en adentro de “\*” donde contiene el margin y paddin de donde se inciara.

Luego esta el método canvas de nuestro juego, ahí le agregamos el color de fondo en este caso es negro, le pusimos el display que es estilo bloques y el margen en 0 auto



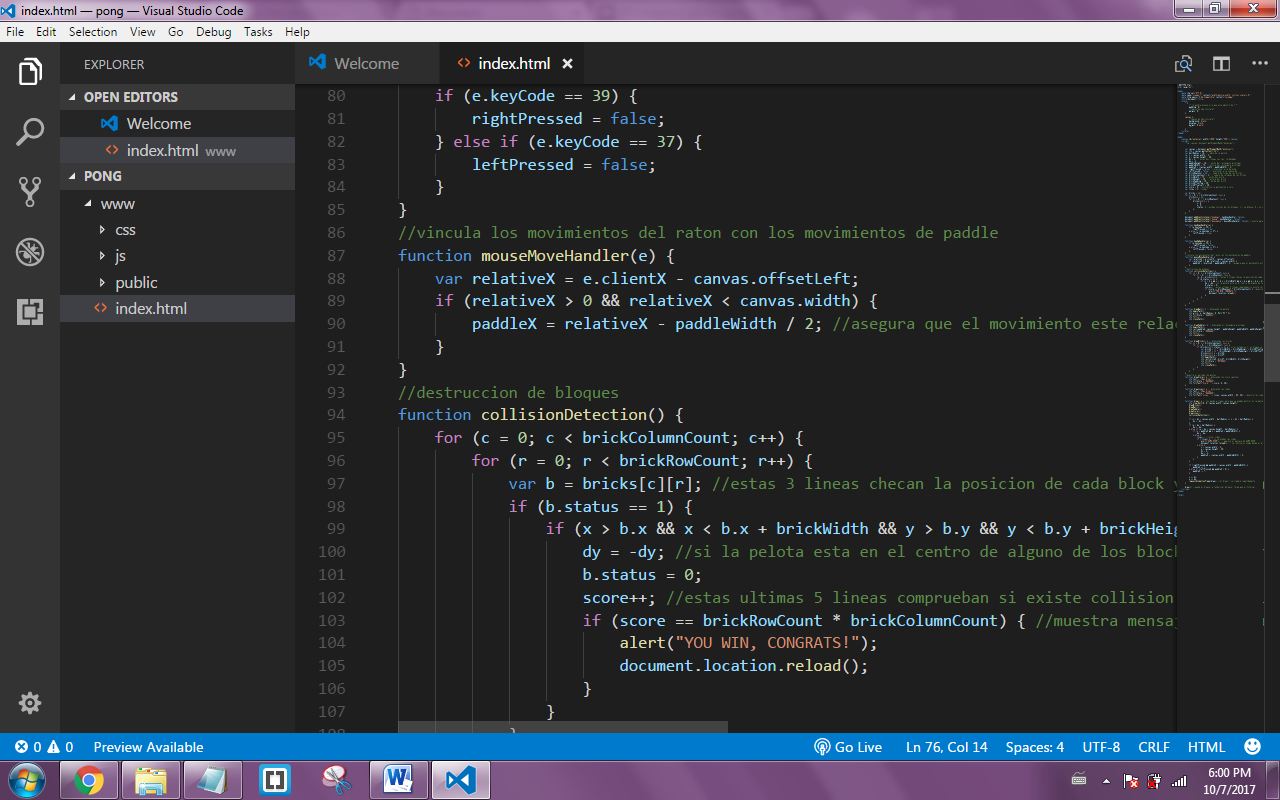
Nuestro canvas será de estilo o intepretacion 2d aquí inicializaremos los valores que iremos necesitando

Usamos el radio que le daremos a la pelota igual a 10 pixeles, después la cordenadas donde iniciara la pelota con su respectivos valores que le daremos al ancho y la altura, luego le ponemos donde se moderara la pelota en 2 en 2, luego esta el tamaño del ancho del paddle de 10 pixeles, y el ancho del paddle en 75 pixeles, luego sacamos el paddleX , luego cuando presionamos derecha o izquierda con la tecla, numero de filas de los bricks igual a 11 y numero columnas de los bricks igual a 3, ancho y altura del brick en 75 y 20 pixeles, distancia de los brick en 20, que este una distancia de los bordes de superior y inferior en 30 pixeles cada uno, los puntos (score) que inicien en 0 y por últimos las 3 vidas



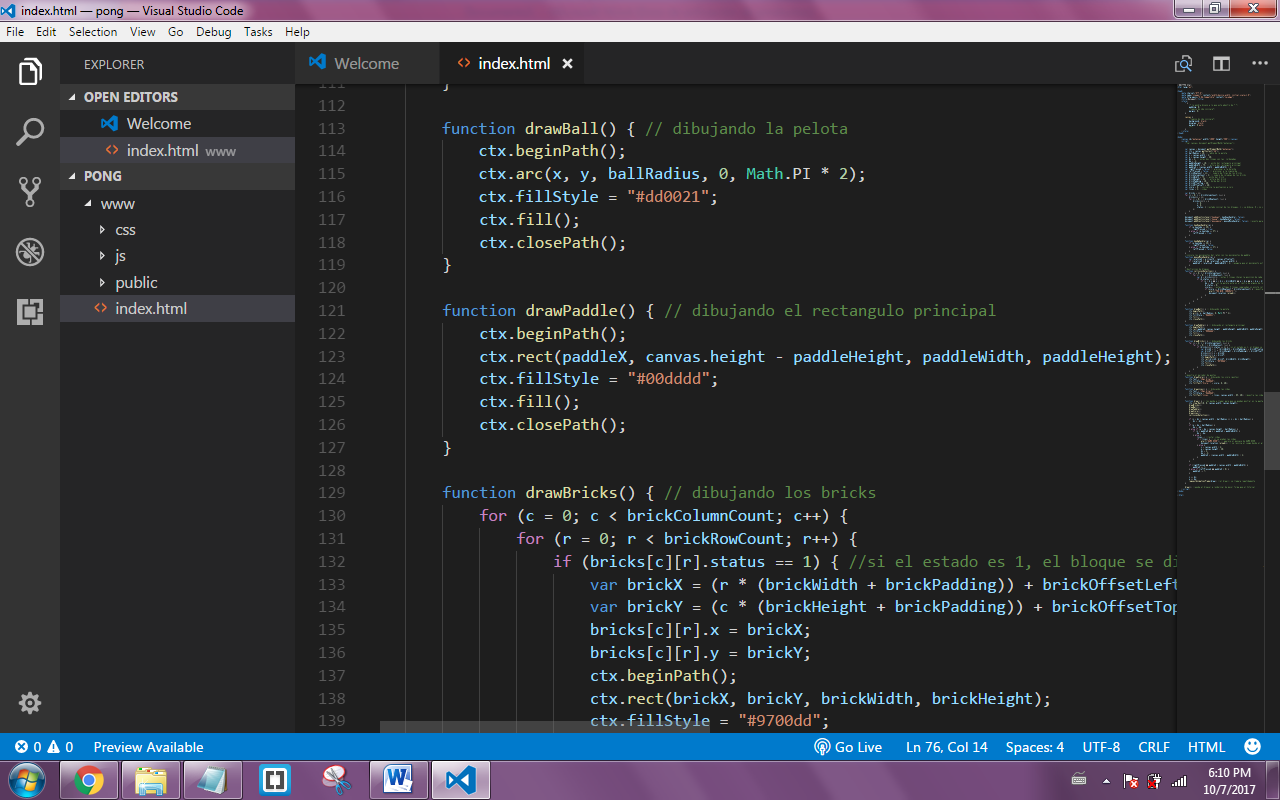
Aquí tenemos donde se va calculando la distancia entre el numero de columnas y filas de los bricks, luego esta los eventos de los movimientos de keyDownHandler que es efecto de moverse abajo, keyUpHandler que es arriba y mouseMoveHandler que es el de efecto para detectar los movimientos del raton

Luego llamamos a la función keyDown y le damos las condiciones de si apreta la tecla s y sale el efecto de presiona a la derecha y si no a la izquierda lo mismo con la función keyUp



Luego llamamos la función mouseMoveHandler que es donde pondremos las condiciones para que pueda vincular los movimientos del raton con los movimientos del paddle (rectángulo principal) y que segure que el movimiento este relacionado con el centro del paddle y no la orilla

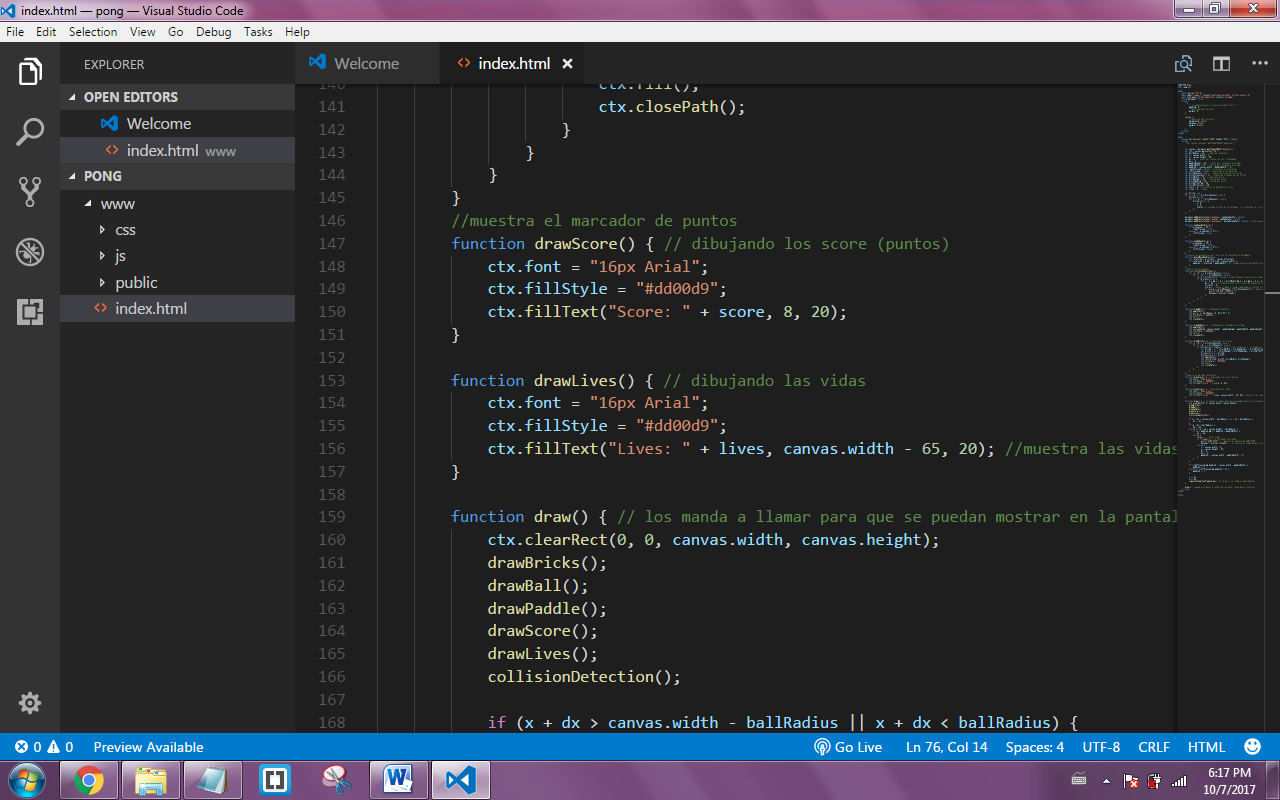
Luego tenemos la función de collisiionDetection (destrucción de bloques) , las tres primeras líneas checan la posición de cada block y la compara con la posición de la pelota , si la pelota está en el centro de alguno de los bloques, se cambiara el rumbo de la pelota , se a sumando los puntos (score por cada bloque destruido, estas ultimas tres líneas comprueba si existe colisión con el bloque y si es asi, ya al destruir todos los bloques y que no quede nada, mostrara una alerta o mensaje en la pantalla de que ganaste el juego y se reiniciará el juego en su posición que estaba antes



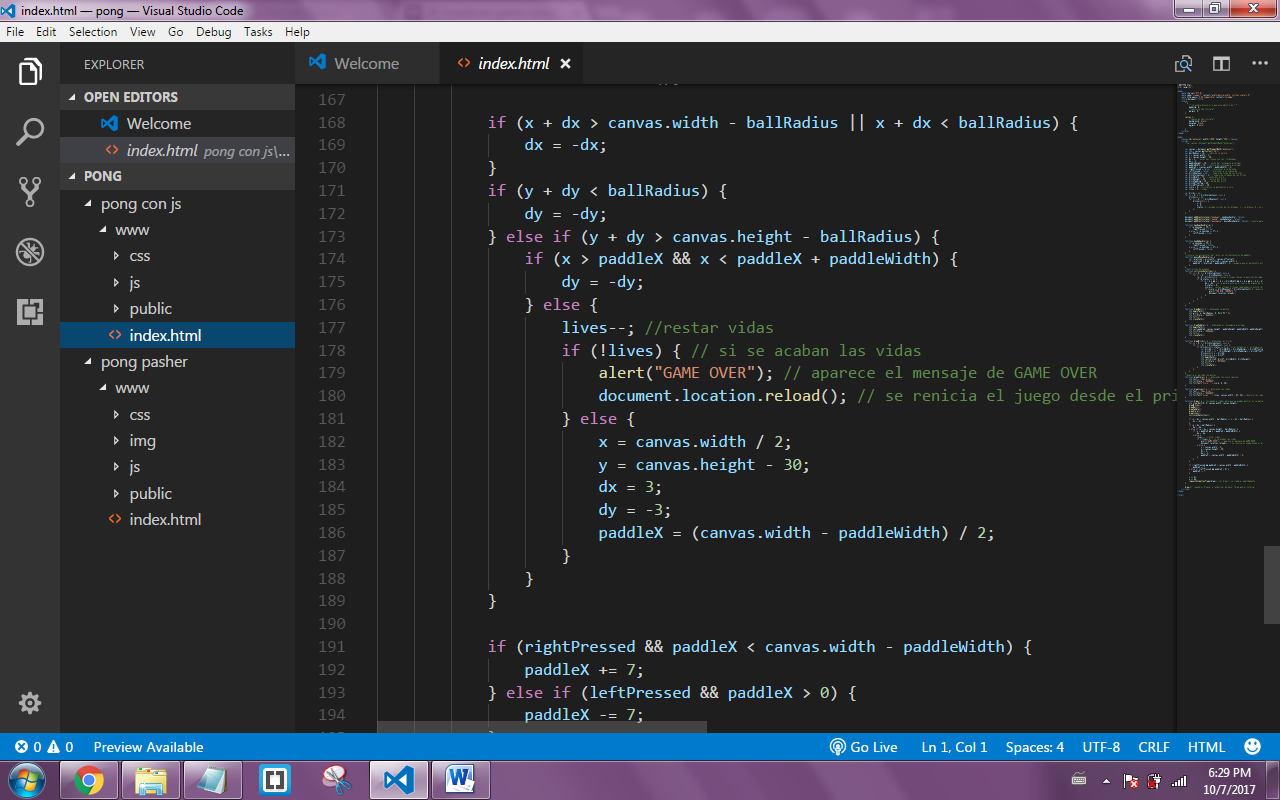
En la función drawBall se a dibujar la pelota con sus tamaños (radio), color y el relleno en este caso de color rojo

En la función drawPaddle se dibujara el rectángulo principal con su tamaño de ancho y altura y cordenadas de donde estará su posición y se pondrá un color de relleno en este caso de color azul

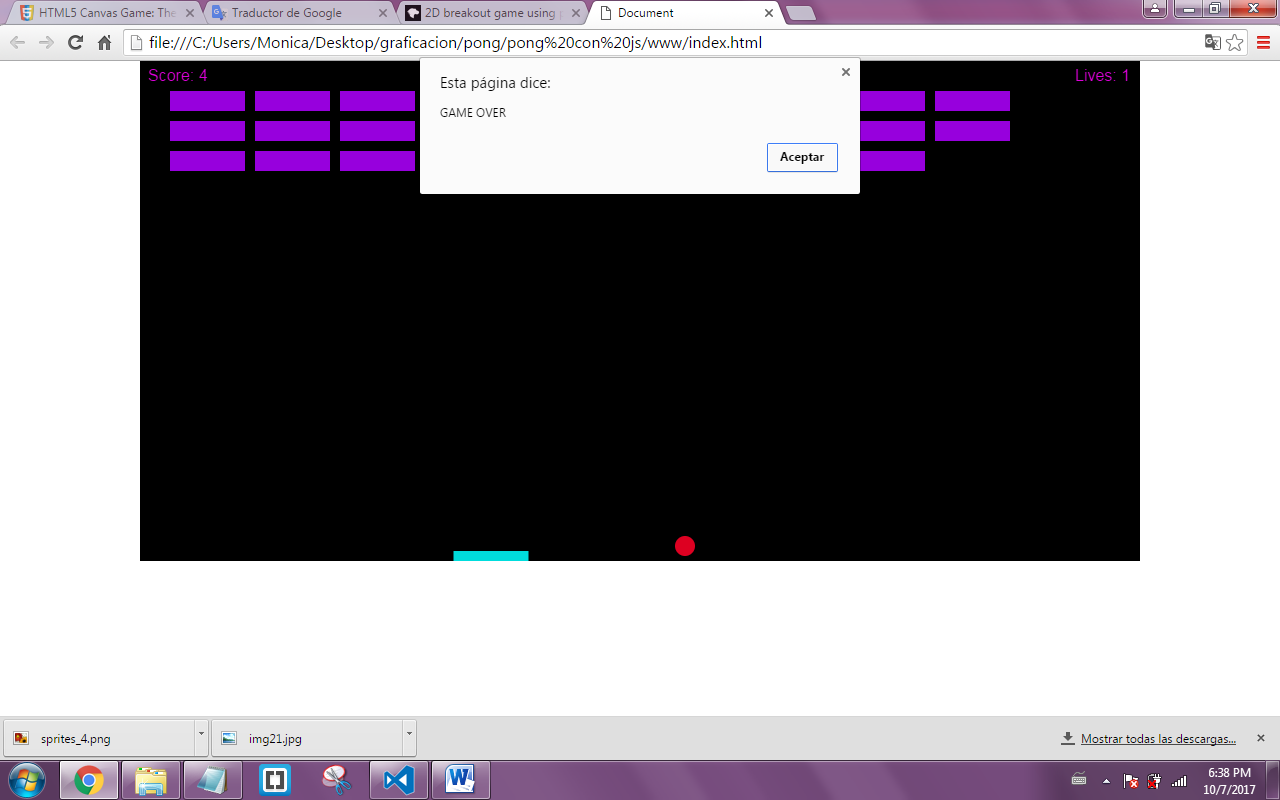
En la función drawBricks se dibujaran los bloques que destruirá la pelota igual sacando sus condiciones con respecto a los números de columnas y filas, si el estado es 1 el bloque se dibuja y si el estado es 0 no se dibuja, cuando se dibuja el bloqye o brick se pone con sus respectivos tamaños de ancho y altura y su relleno de color en este caso de color morado

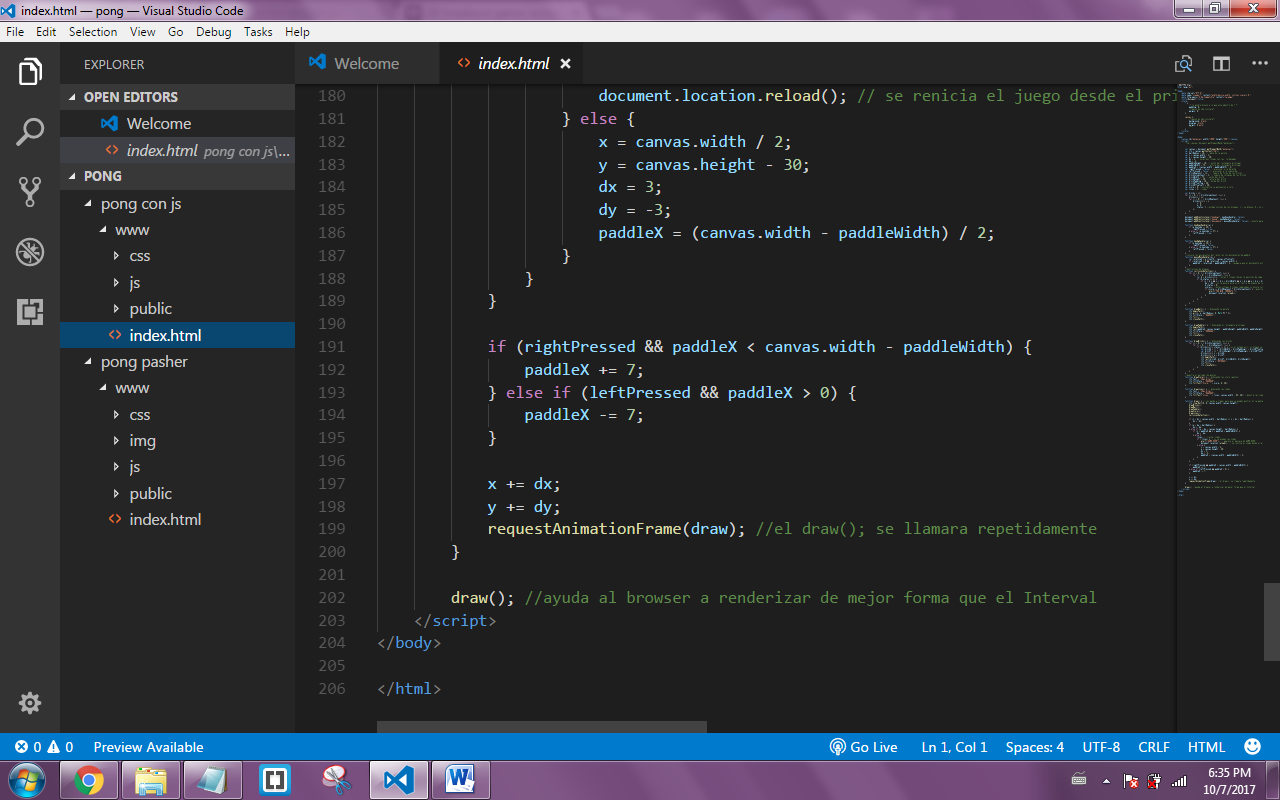


En la función drawScore se va colocando un texto donde ahí se le pone el estilo de fuente en este caso arial de 16px y que el color del texto sea morado bajito y en el texto se vaya sumando los puntos 1 en 1 y las cordenadas para que se muestre en la posición izquierda de la pantalla del juego y En la función Draw los manda a llamar para que se puedan mostrar en la pantalla del juego



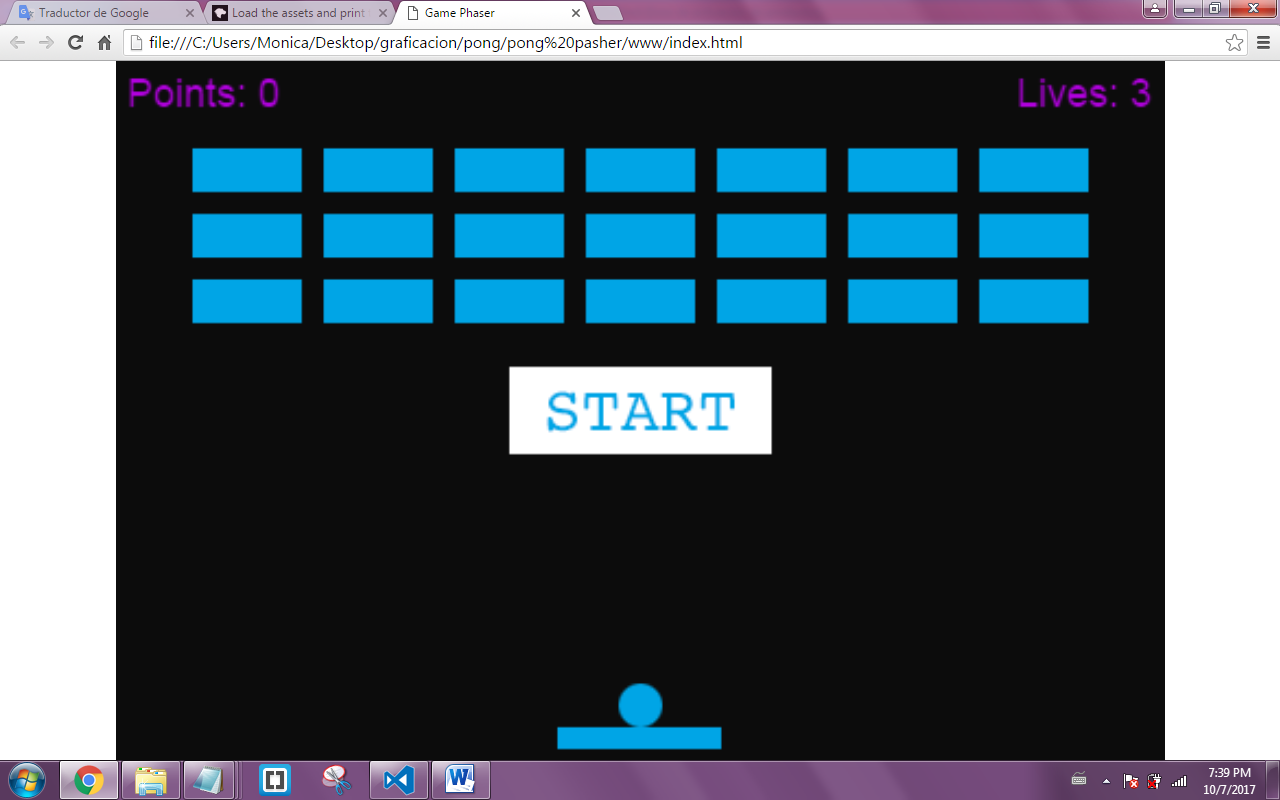
Aquí esta donde se pone lo del rebote de la pelota en parámetros de los lados exepto el de abajo de la pantalla, pero si la pelota sobrepasa el borde de abajo se ira restando una vida, si se acaban todas las vidas aparece el mensaje o alerta en la pantalla de Game Over y se reinicia el juego desde el principio con las posiciones de antes, pero si toca antes con el paddle antes de caer al borde de abajo, entonces no se quitara las vidas y seguirá continuando el juego



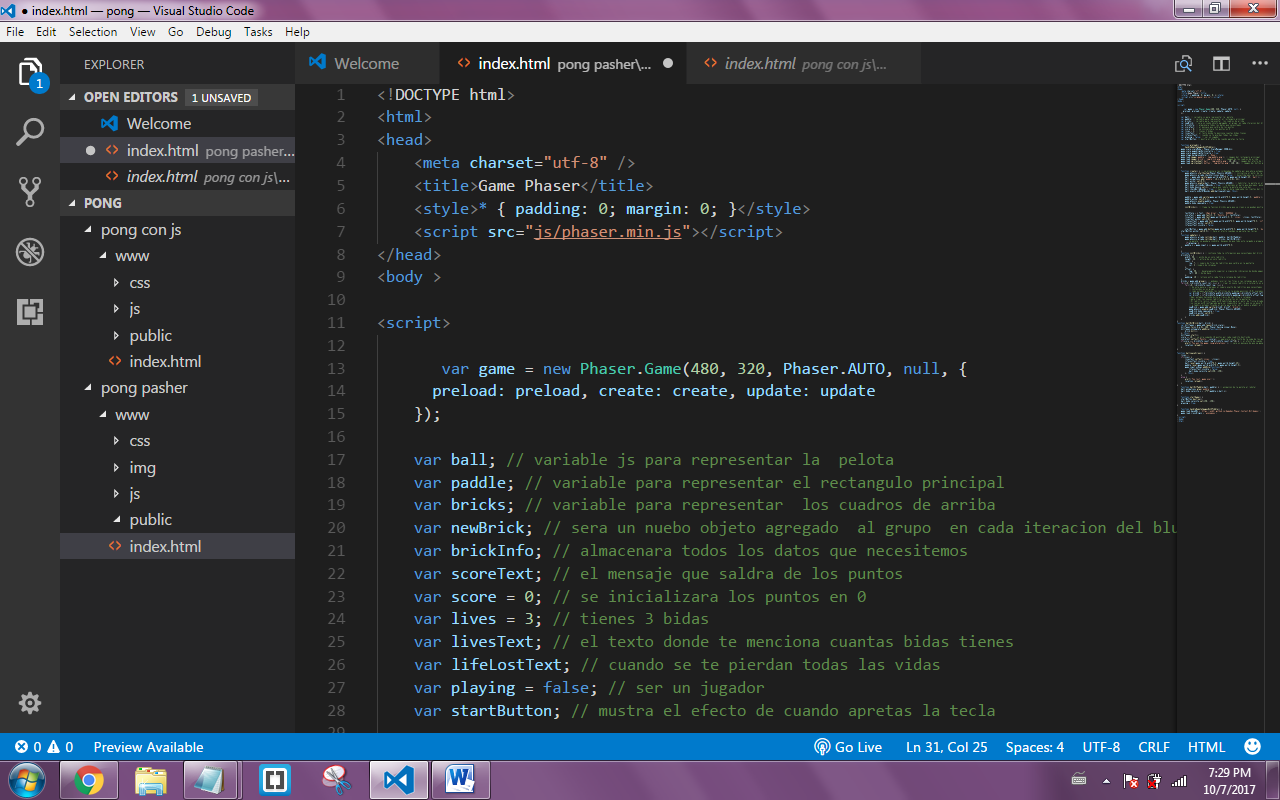


Aquí esta por ultimo lo del efecto de presar hacia la derecha e izquierda para que el padle se mueva y en el requesAnimationFrame el draw se llamara repetidamente y el draw ayuda al browser a renderizar de mejor forma que el interval

**Usando Phaser:**

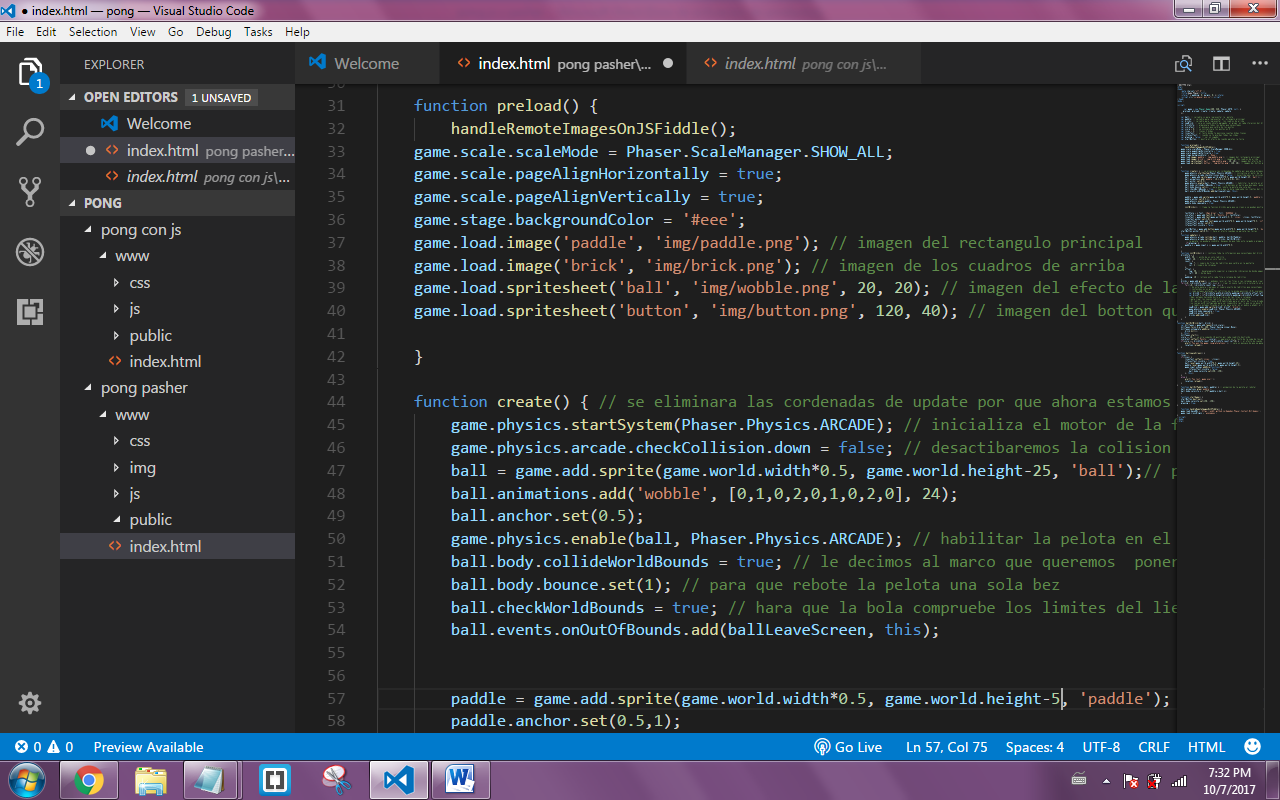


**Codigo:**



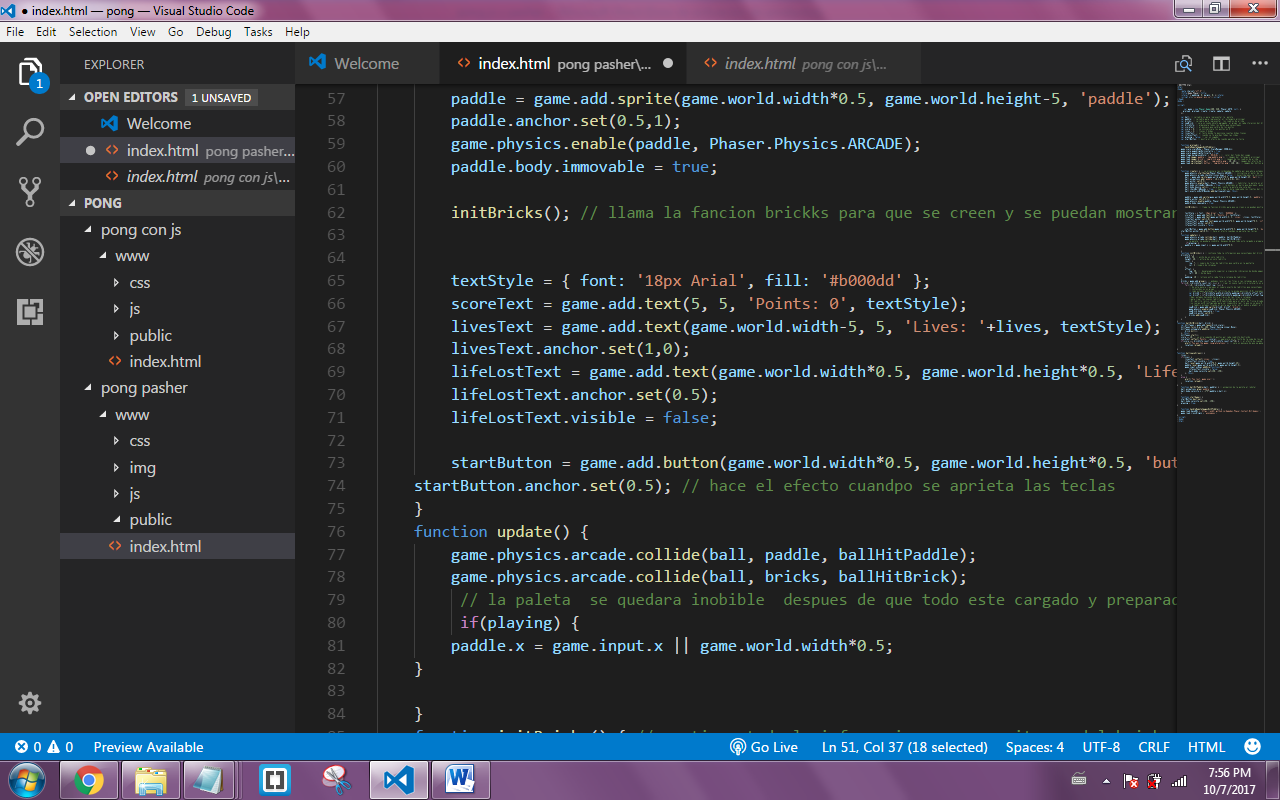
En el head tenemos los datos de la página del juego, el titulo etc y en script tenemos la carpeta del pahser que descargamos para poder usar el codigo de phaser que tenemos en la carpeta js ya en el body dentro pusimos el script donde ahí en la función game llamaremos a las funciones que necesitaremos manejándolos con el codigo de phaser

Primero llamaremos las variables que necesitamos, la variable ball que es la que representara la pelota, paddle la que representa el rectángulo principal, bricks que representa los cuadritos o ladrillos que destruiremos, newbrick ser un nuevo objeto agregado al grupo en cada iteración del blucle, brickinfo almacenara todos los datos que necesitemos, scoreText el mensaje que saldrá de los puntos, score que inicara en 0, las 3 vidas, livestext el texto donde te menciona cuántas vidas tienes, lifeLostText cuando se te pierdan todas las vidas, playing el del jugador, starButton muetsra el efecto de cuando aprietas la tecla.



Luego en Funcion preload ejecutaremos las imágenes que necesitaremos de los objetos principales, también está la posición del modo de la página del juego, horizontal y vertical, el color del fondo en este caso negro, las imágenes: de paddle, brick, la animacion de la pelota (wobble) con sus cordenadas de posición y la imagen del botton de “Start” con sus cordenadas de posición

En la función create, tenemos la inicialización del kotor de la física del juego (startSystem), checkCollision esta en false ya que desactivaremos la colision de la pelota con el borde interior de la pantalla , tenemos mpara mostrar la pelota: los dos primeros son las cordenadas del lienzo y el otro, el nombre de la imagen pelota, el uso de la animacion wobble en la pelota y el ancho de la pelota, , también habilitaremos la pelota en el sistema de la física, en collideWorldBounds está en true ya que le decimos al marco que queremos poner los límites del canvas como paredes para que la pelota no se sobrepase, luego en body.brounce para que rebote la pelota una sola vez y en check WorldBounds está en true porque hara que la pelota compruebe los límites del lienzo y se hace el evento del efecto de la pelota.

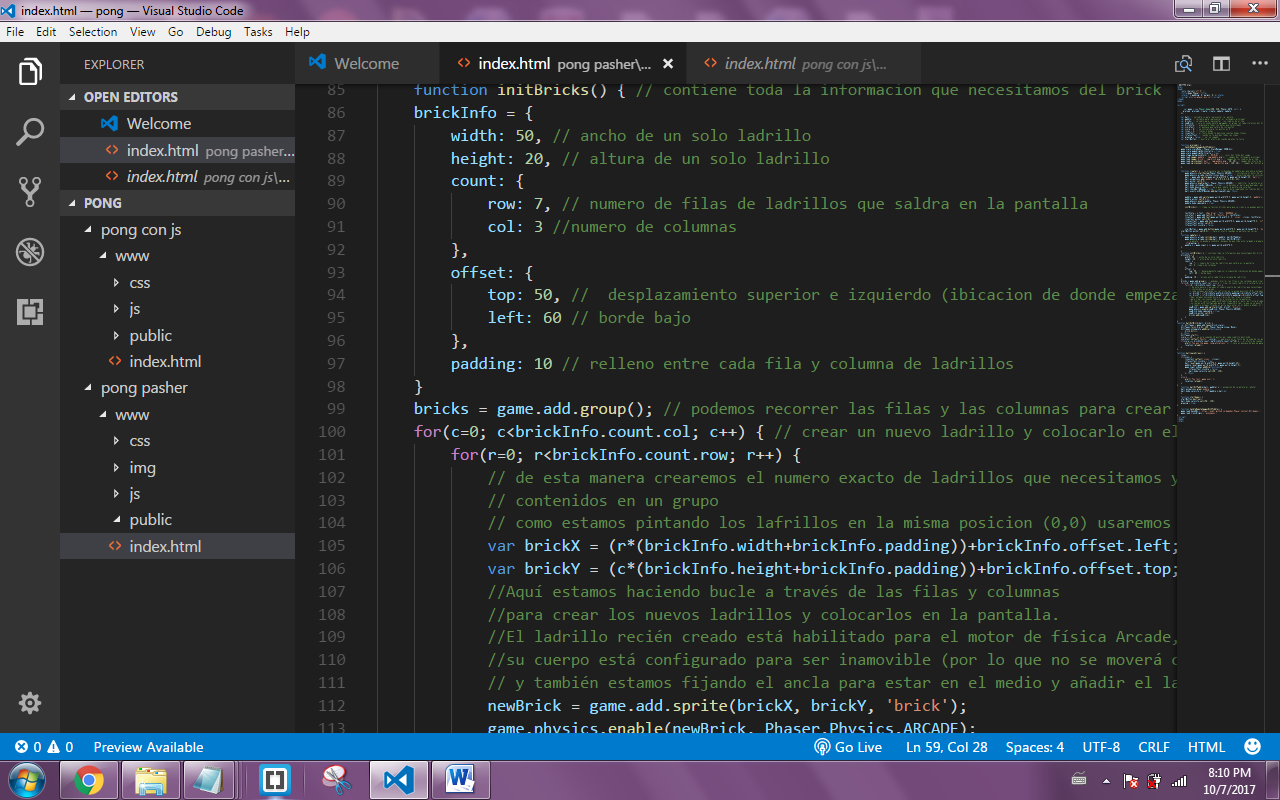


Luego en game.add. sprite, para mostrar en el rectángulo principal, las cordenadas son para posicionar el rectángulo principal donde queramos y la imagen del rectángulo, esta también el ancho del padle y habilitar el padle en el sistema de la física del juego y que sea inmóvil al iniciar el juego. initBricks es para llamar la función bricks para que se creen y s epuedan mostrar los cuadros

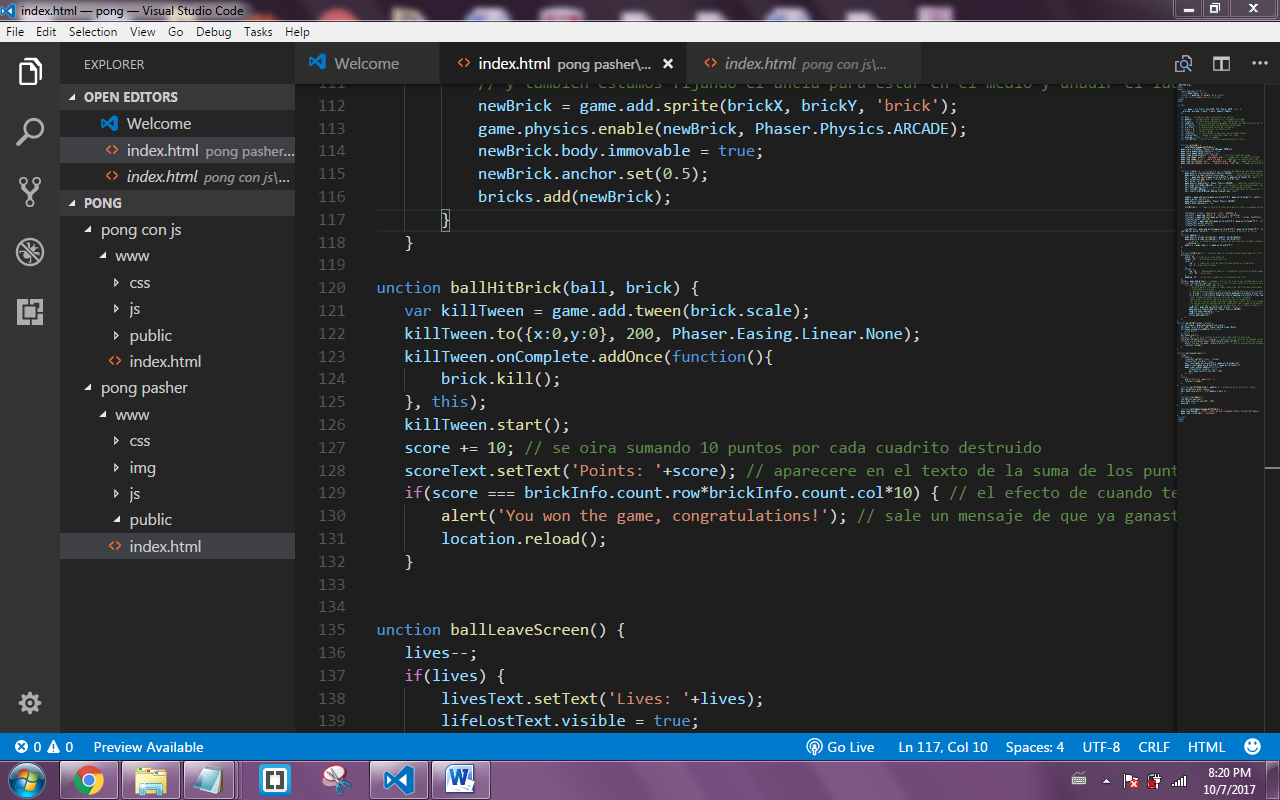
Después tenemos stulo de texto arial, tamaño 18 pixeles, color del texto morado, ahí pondremos el txto para los puntos por cada ladrillo destruido y la posición del lado derecho de la pantalla y el texto donde se mencionara las vidas que tenemos igual con su posición del lado izquierdo de la pantalla, el ancho del texto de las vidas. Y también el texto que saldrá en el mensaje cuando pierdas todas las vidas , el ancho del mensaje cuando pietrdes las vidas y que se pueda mostrar en la pantalla

Luego inicializamos el botton de start que aparece al principio de la pantralla ante sde que inicie el juego con su posición etc y hace el efecto cuando se preciona el botón para que inicie el juego y el ancho del botón

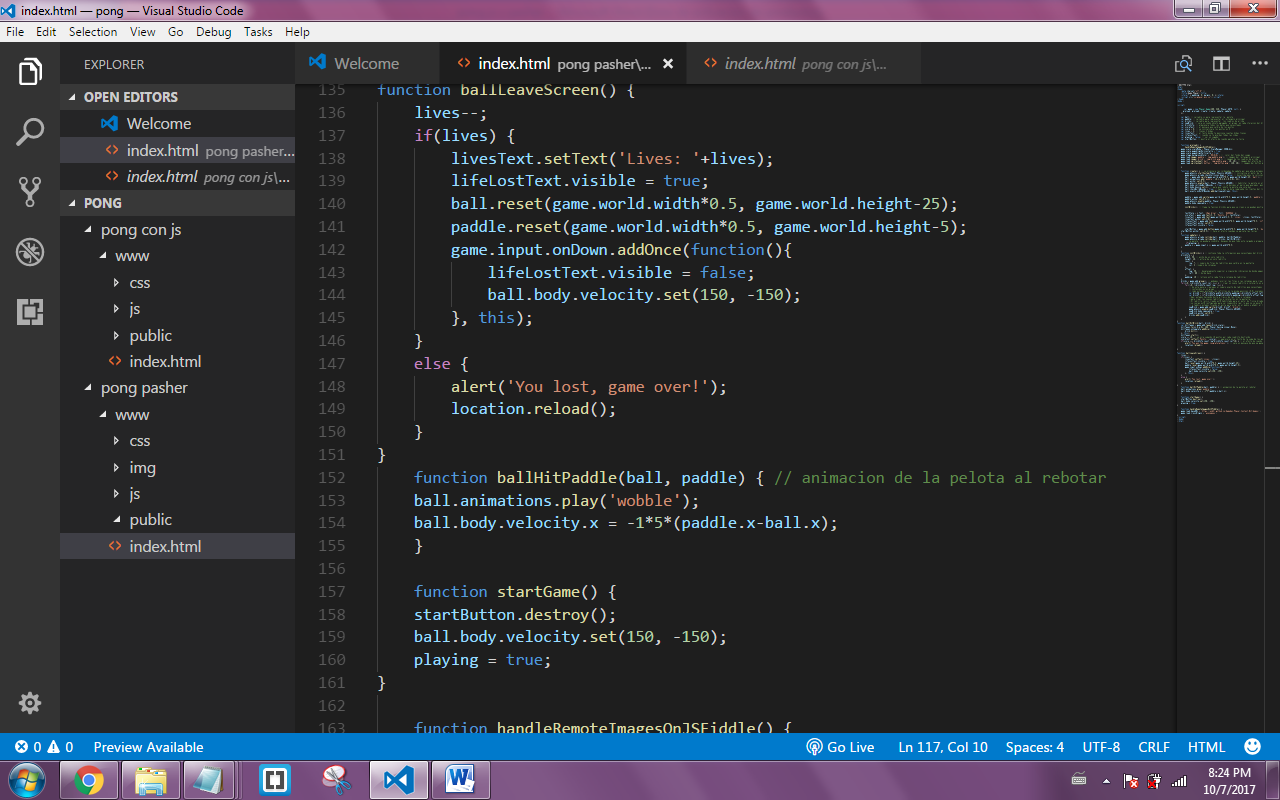
En la función Update tenemos el efecto de que la pelota se queden inmóviles después de que todo este cargado y preparado antes del inicio del juego y que al momento de que el jugador le de al botón comience el juego



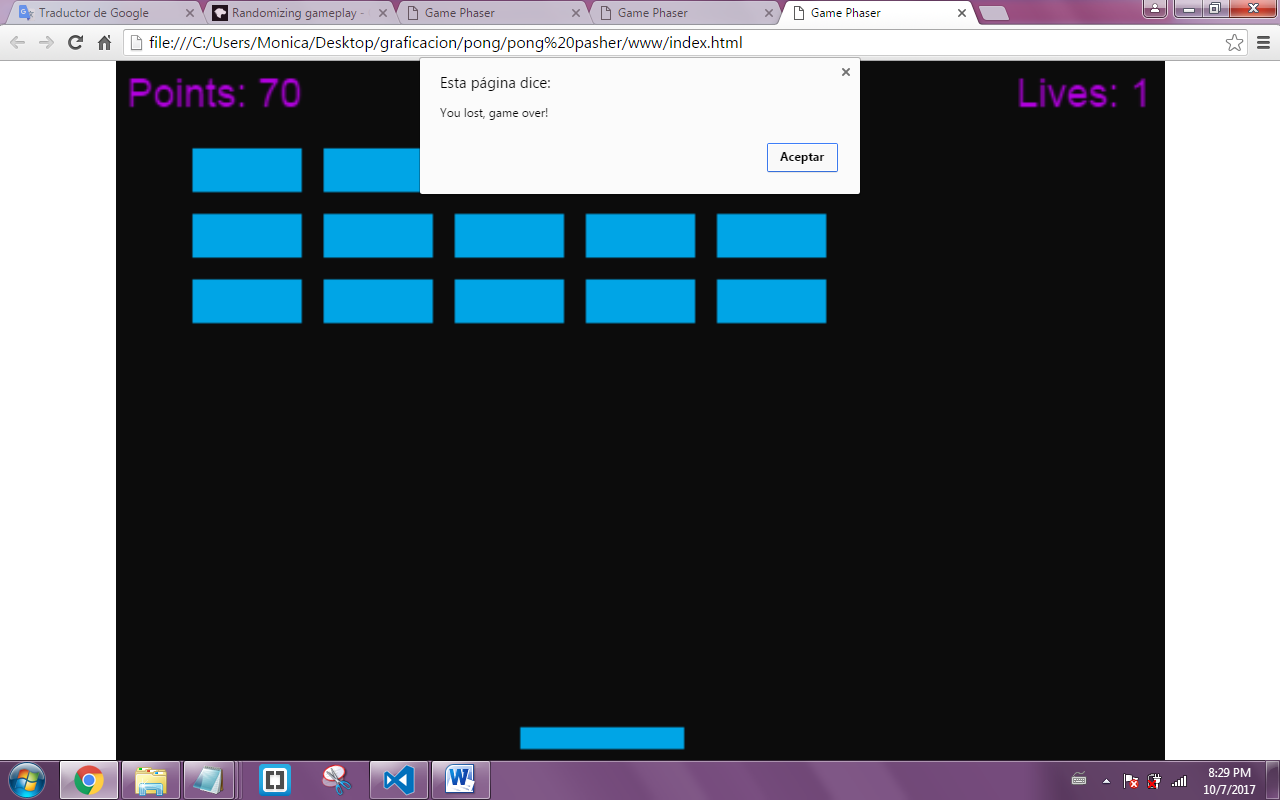
En la función InitBricks contiene toda la información que necesitamos del brick , ahí tenemos el ancho de 50 y altura de 20 de un solo ladrillo, el numero de filas 7 y clumnas 3 de ladrillos que se mostraran, desplazamiento superior e izquierdo ( ubicación de donde empezamos a dibujar los ladrillos y relleno entre cada fila y columna de ladrillos en game.add.group podemos recorrer las filas y columnas para creare un nuevo ladrillo en cada iteración luego crear un ladrillo y colocarlo en el grupo y asi creamos el numeroi exacto de ladrillos que necesitamos pero como los estampos poniendo en la misma posición (0,0) le ponemos el brickX y brickY para ponerkos en su propia cordenada distinta cada ladrillo en las siguientes 5 lineas , aquí estamos haciendo bucle atrabez de las filas y columnas para crear los nuevos ladrillos y colocarlos en la pnatalla , el ladrillo recién creado esta habilitado para el motor de física arcade, su cuerpo esta configurado para ser inmovilizado ( por lo que no se mobera cuando se golpee por la pelota) y también estamos fijando el ancla para estar en el medio y anadir el ladrillo al grupo

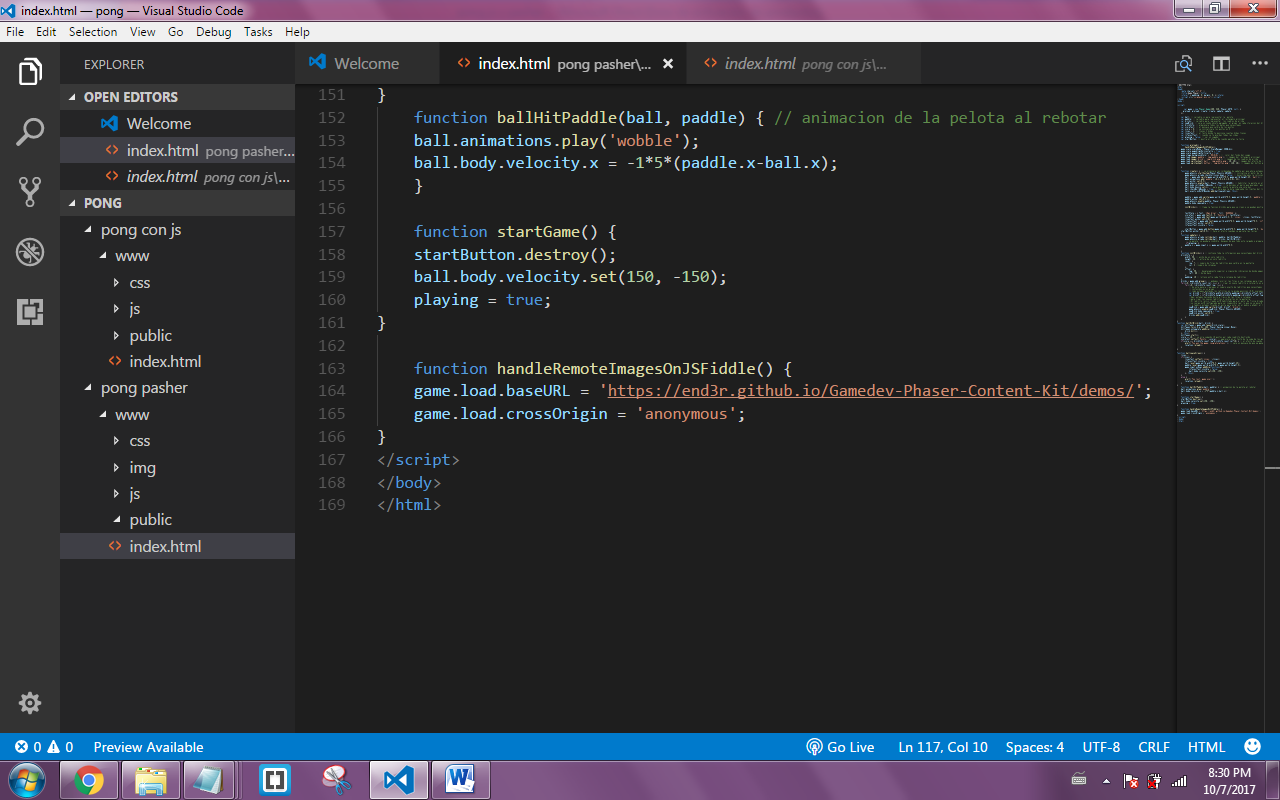


En la función ballHitBrick haremos el efecto de la pelota y el ladrillo donde haremos de que cuando la pelota toque el ladrillo desaparece y que por cada ladrillo destruido se este sumando 10 puntos en el score y cuando ya no este ningún ladrillo en la pantalla aparecerá un mensaje o alerta de que ganaste el juego y todos reiniciara en la posición que estaba al inicio



En la función ballleaveScreen tenemos de que se mostrara en el texto del lado derecho las vidas que tenemos que se inician con 3, también esta la velocidad de la pelota al moverse y si la pelota se sobre pasa del borde de abajo, se iran perdiendo vidas y si se pierden todas aparecerá un mensaje de que perdiste el juego y todo se reiniciará a la posición que estaba al inicio





En la función ballHitpaddle tenemos el efecto de la pelota que cuando esta rebota en el paddle sale la animacion de wobble que parece que tiembla al rebotar. En la función startGame es cuando el jugador aprieta el botón de start y inicia el juego y la velocidad de la pelota al iniciar el juego y por último en la función handRemoteImagesonJsFiddle es donde se carga el url de github de la pagina de pasher y su origen que es anonymous