

ANDRES FELIPE BETANCURT RIVERA

BRANDON CARVAJAL GONZALEZ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MANUAL DE PROGRAMADOR

2015

SHOOTER

Bienvenido Usuario agradecemos que haya escogido nuestro juego para su entretenimiento, a continuación

Describiremos nuestro producto y enseñaremos como funciona nuestro juego.

"SHOOTER" es un juego de disparos 2D que consiste en buscar en una pantalla, con una mira, una diana que está en una posición aleatoria de la pantalla, el juego tiene una puntuación la cual aumenta si el jugador acierta en la diana y disminuye si este dispara aleatoriamente sin buscar la diana.

Nuestro juego se compone de tres niveles de dificultad que varían con respecto al tiempo, entre más dificultad

Menor tiempo de sumar puntos.

A continuación daremos un recorrido por el código del juego.

INDICE

PRECAUCIONES………………………………………………………………………………………………………………….....3

Manual Programador……………………………………………………………………………………………………………..4

Lista de archivos……………………………………………………………………………………………………………………10

PRECAUCIONES

* No borrar las librerías.
* Ejecutar el programa en el lenguaje “#lang racket”.
* Leer los comentarios puestos sobre cada función.

Manual Programador

Al abrir el programa debe cerciorarse que el lenguaje es el correcto para pasar a ejecutar, en caso de tener algún problema o si desea modificar los valores de tiempo, vamos a ver que hace cada función del código:

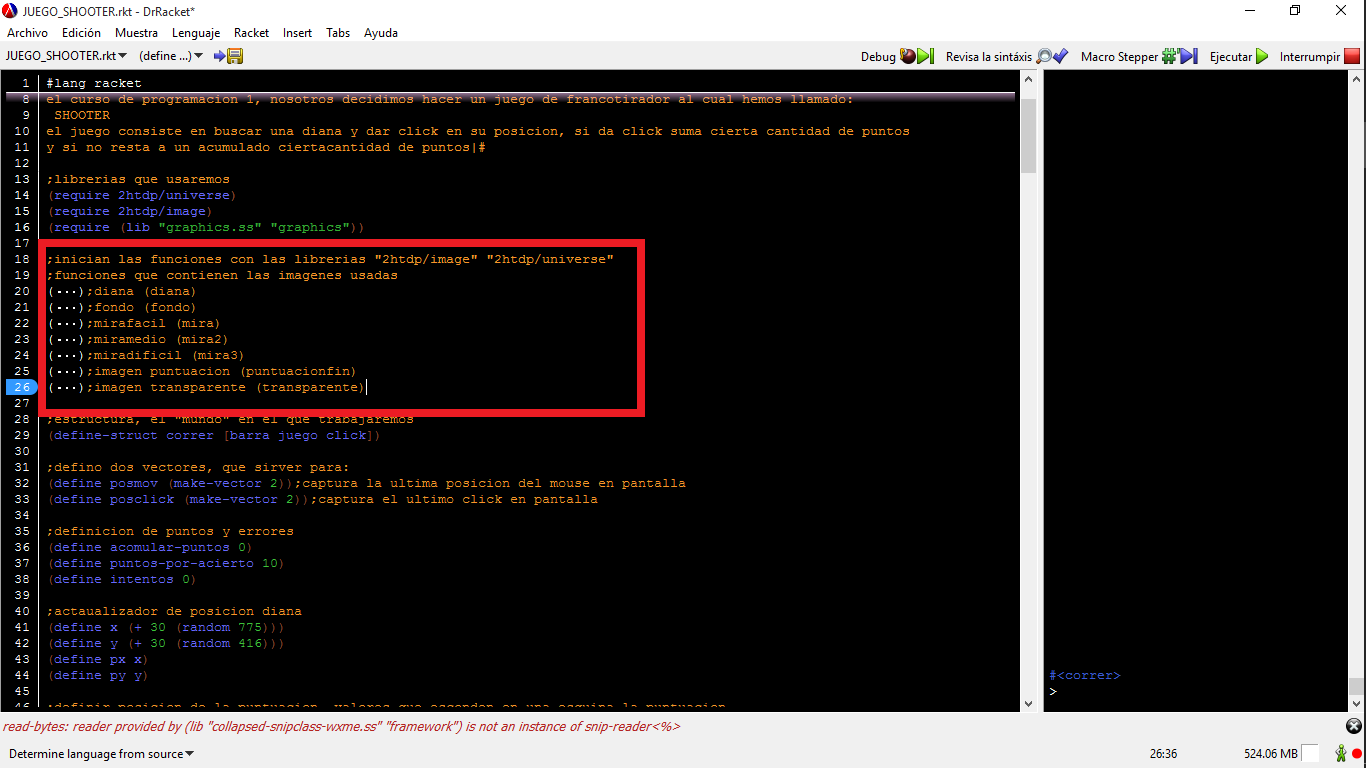
Primero nos encontramos con las librerías que estamos utilizando, debe verificar que sean las librerías correrctas, en este caso:

“(require 2htdp/universe)

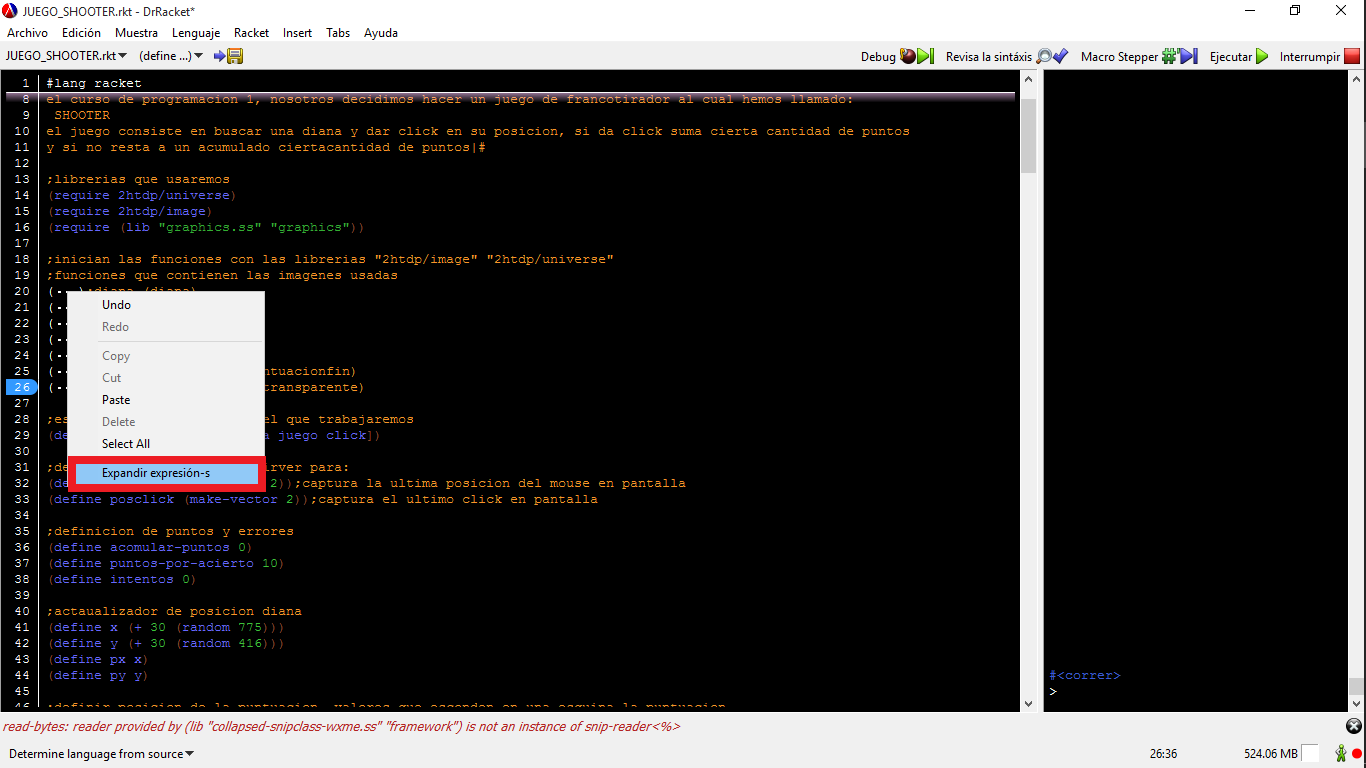
(require 2htdp/image)

(require (lib "graphics.ss" "graphics"))”

Nos encontramos después con las imágenes que usaremos a lo largo del código:



Si desea ver o modificar las imágenes debe hacer clic derecho sobre la función que desea ver y seleccionar la opción “Expandir expresión-s”



De este modo podrá modificar la función borrando o cambiando la imagen allí definida.

Esta función trabaja con una estructura de valores que está definida como “correr” y contiene “barra juego click”, cada uno de estos tres captura diferentes valores, barra se encarga de recibir los valores que hacen que la función que simula el tiempo se ejecute correctamente, juego se encarga de capturar las posiciones que tiene el mouse en sus dos estados, moviéndose u haciendo clic, y por ultimo click captura un valor verdadero o falso si se hace clic en pantalla.

Los vectores “posmov” y “posclick” me capturan la última posición del mouse en pantalla y la posición donde se hizo clic respectivamente, de modo que es más fácil acceder a esta información de esta manera.

Defino los acumuladores que me van a servir para llevar el recuento de aciertos y de fallos que tenga el jugador, “puntos-por-acierto” es la función que determina la cantidad de puntos a sumar si el clic acierta en la diana. Definición de posición en x “px” y posición en y “py” utilizando random ya que usaremos estas posiciones para ubicar la diana por primera vez en pantalla.

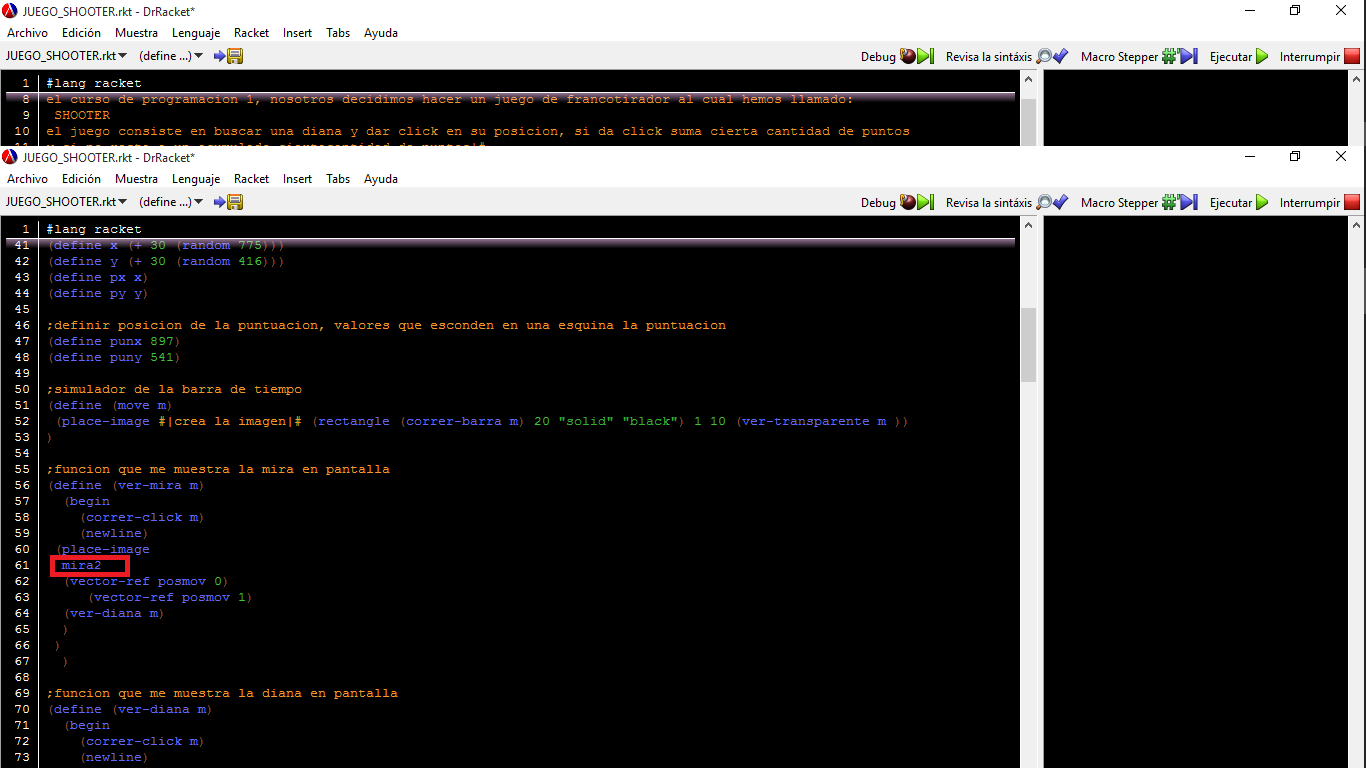
Defino otros dos puntos “x” y “y” que me sirven para “esconder” la puntuacion final en la inferion derecha, “punx” y “puny” respectivamente.

La función “(move m)” se encarga de dibujar en pantalla la barra que simula el tiempo y la ubica encima de una cadena de funciones que contienen las imágenes, la función “(ver-transparente m)” ubica la imagen “transparente.png” sobre la función “(ver-puntos m)” que a su vez ubica la puntuación final como una imagen en pantalla(aquí la función “posx” “posy” se utilizan para esconder dicho puntaje) sobre otra función, “(puntos m)”, que pone un texto como una imagen sobe otra función, etc… así hasta llegar a la función “(ver-diana m)”, que lo que hace es ubicar la diana sobre el fondo que ya habíamos definido.

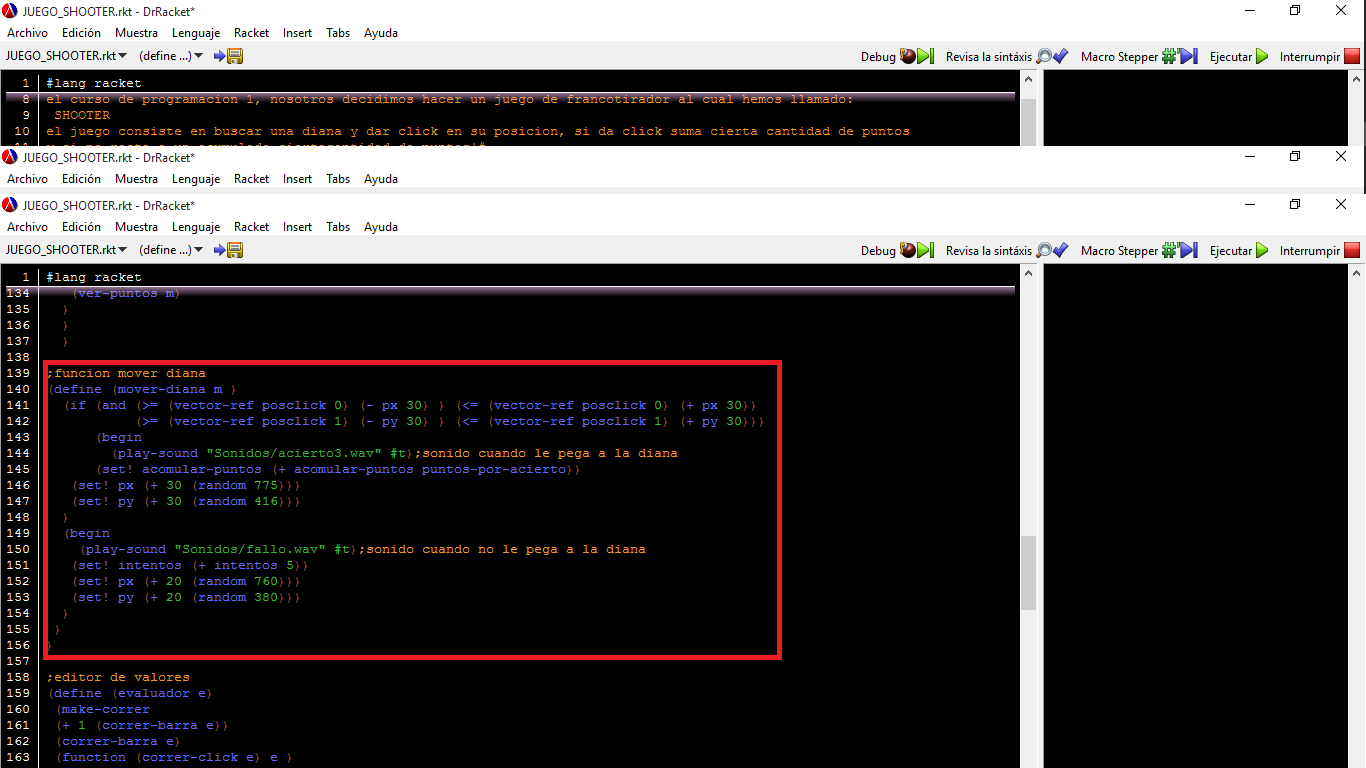
De modo que lo anterior lo que hace es trabajar con las imágenes haciéndolas poner unas sobre otras como una especie de capas en la que el rectángulo que simula el tiempo siempre estará arriba y después va una imagen transparente que se pone sobre un texto que a su vez se pone sobre otro texto que se pone sobre otro texto que se pone sobre la imagen definida como “mira” que se pone sobre la imagen definida como “diana” que se pone finalmente sobre la imagen definida como “fondo”, lo que permite que podamos ver unas imágenes y otras no.

Estamos ofreciendo tres imágenes referentes a la mira, son tres tamaños que damos a escoger y que el programador puede escoger al gusto siendo la imagen definida como “mira” la más grande y la imagen definida como “mira3” la más pequeña.

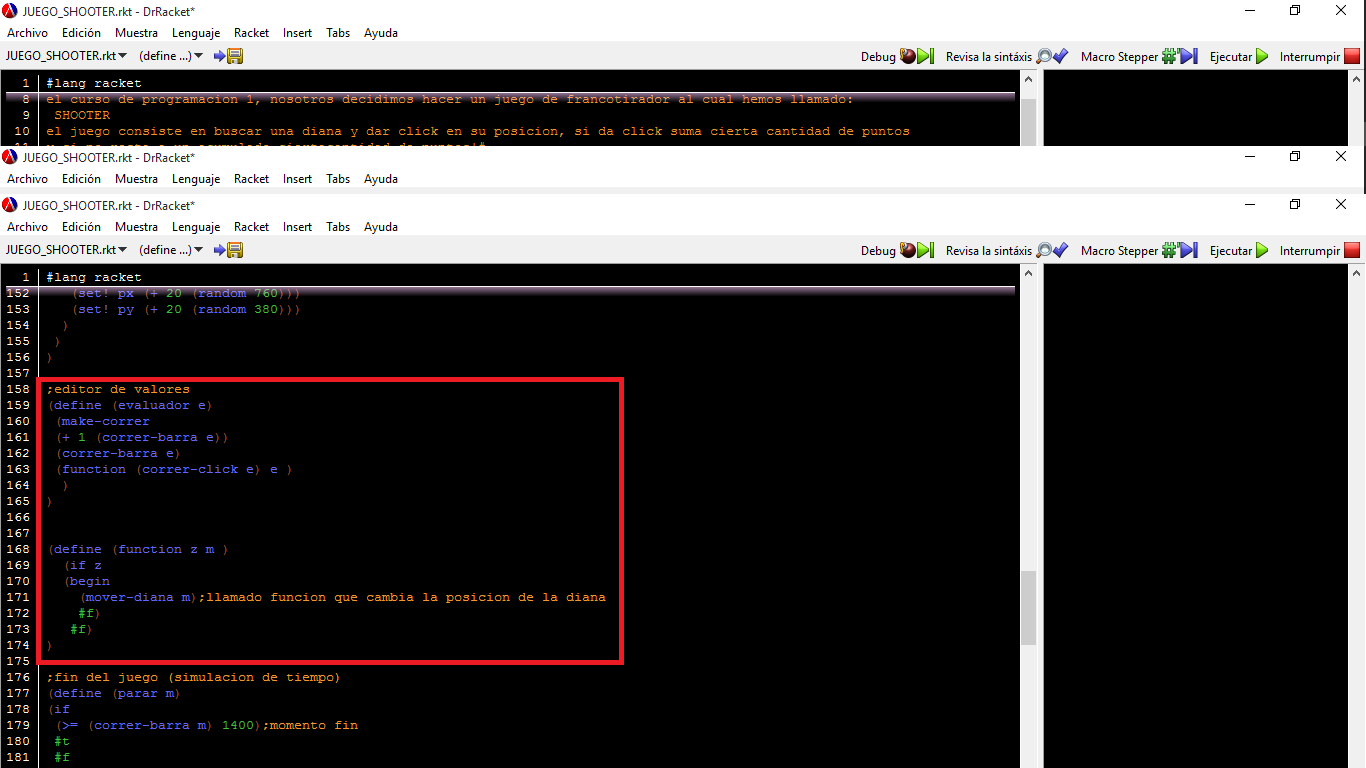
Si deseamos cambiar la mira lo que debemos hacer es ir a la función “(ver-mira m)” y cambiar “mira2” por una de las tres imágenes de la mira definidas (“mira” “mira2” “mira3”), teniendo en cuenta que al cambiar esta mira hay que cambiarla en todas las funciones en la que usemos la mira.



La función que se encarga de evaluar si el clic hecho esta en el rango de la diana, es decir si el clic hizo “impacto” en la diana, es la función definida como “(mover-diana m)” teniendo en cuenta que “m” es la estructura que va cambiando, esta función lo que hace es evaluar si la posición al hacer clic, obtenida en el vector “posclick”, esta dentro del rango de la diana, se evalúa si la posición en x es mayor o igual que la resta de la posición en x definida como “px” y 30 y que si es menor o igual a la suma de la posición en x definida como “px” y 30, al mismo tiempo evalúa en y si es mayor o igual que la resta de la posición en y definida como “py” y 30 y que si es menor o igual a la suma de la posición en y definida como “py” y 30, de modo que si cumple estas cuatro condiciones se hizo clic dentro de la diana, lo que hace que se sumen los puntos correspondientes en el acumulado definido como “acomular-puntos” y que se reproduzca el sonido de acierto ubicado en la carpeta Sonidos “acierto.wav” a la vez que modifica los valores “px” “py” para que de esta forma la diana cambie de posición y se proceda a buscarla de nuevo.



La función que se encarga de hacer funcionar la barra y de determinar si se hizo o no clic es la función “(evaluador e)” donde “e” es la estructura a modificar, lo que hace esta función es que cada vez que se ejecute sume 1 a la posición del rectángulo que simula el tiempo y por esto el rectángulo parece que se mueve, esta función evalúa si se hace o no clic devolviendo #f y #t siendo #t un clic en pantalla, entonces es ahora donde se hace el llamado a “(mover-diana m)” para que cambie de posición a la diana y haga lo correspondiente si se hizo o no clic en la diana, este llamado se hace a través de la función definida como “(function z m)” donde “m” es la estructura y “z” el valor de verdad (#t).



Las funciones que hacen que el programa termine son las funciones “(parar m)” y “(dar-fin f)” siendo “m” la estructura y “f” también. Lo que hace “(parar m)” es evaluar si la posición de la estructura “correr” conocida como “barra” es menor o igual a un número que se considere que de buen tiempo, en este caso el número es 1400 dando un máximo de tiempo de 1:10 minutos, este valor (enmarcado en azul) se puede modificar al gusto del jugador. Y cuando la función “(parar m)” sea verdadera la función “(dar-fin f)” se ejecutara y terminara el proceso haciendo los siguientes cambios:

* Cambiará “posx” y “posy” a una posición en la pantalla donde se puedan ver (recordemos que “posx” y “posy” eran los encargados de tener la puntuación total en la esquina inferior derecha para que no se viera).
* Cambiará la mira por la imagen transparente para poder ver la imagen que contiene el texto “PUNTUACION:” definida como “puntuacionfin”.
* Cambiará la imagen definida como “transparente” por la imagen definida como “puntuacionfin”.

De este modo se observara una pantalla en la que se ve todo el fondo, la última diana ubicada y la puntuación en mitad de pantalla.

Imagen en el juego.

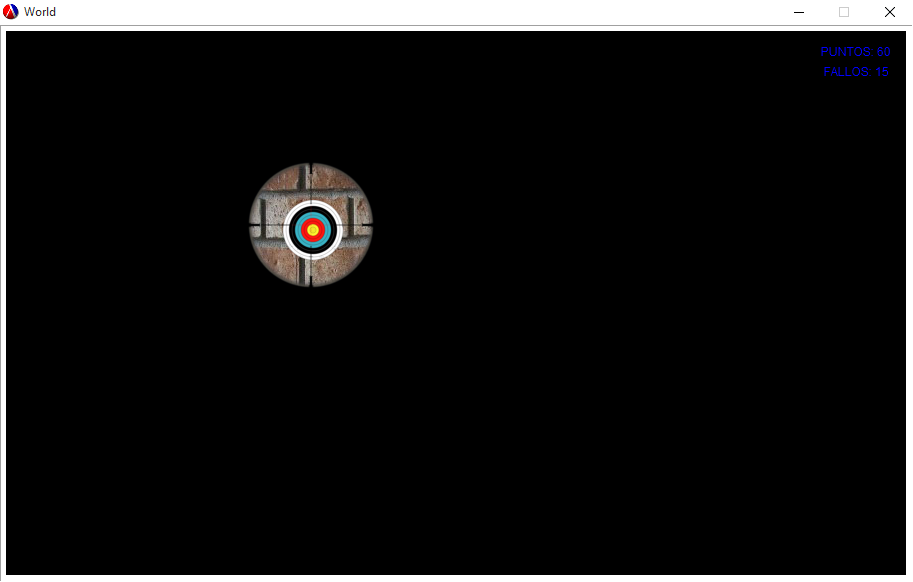
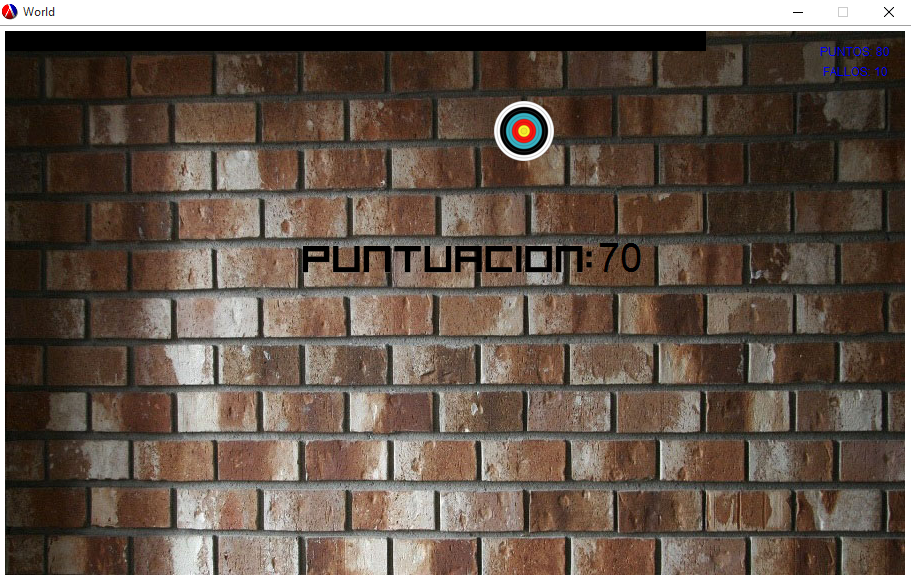
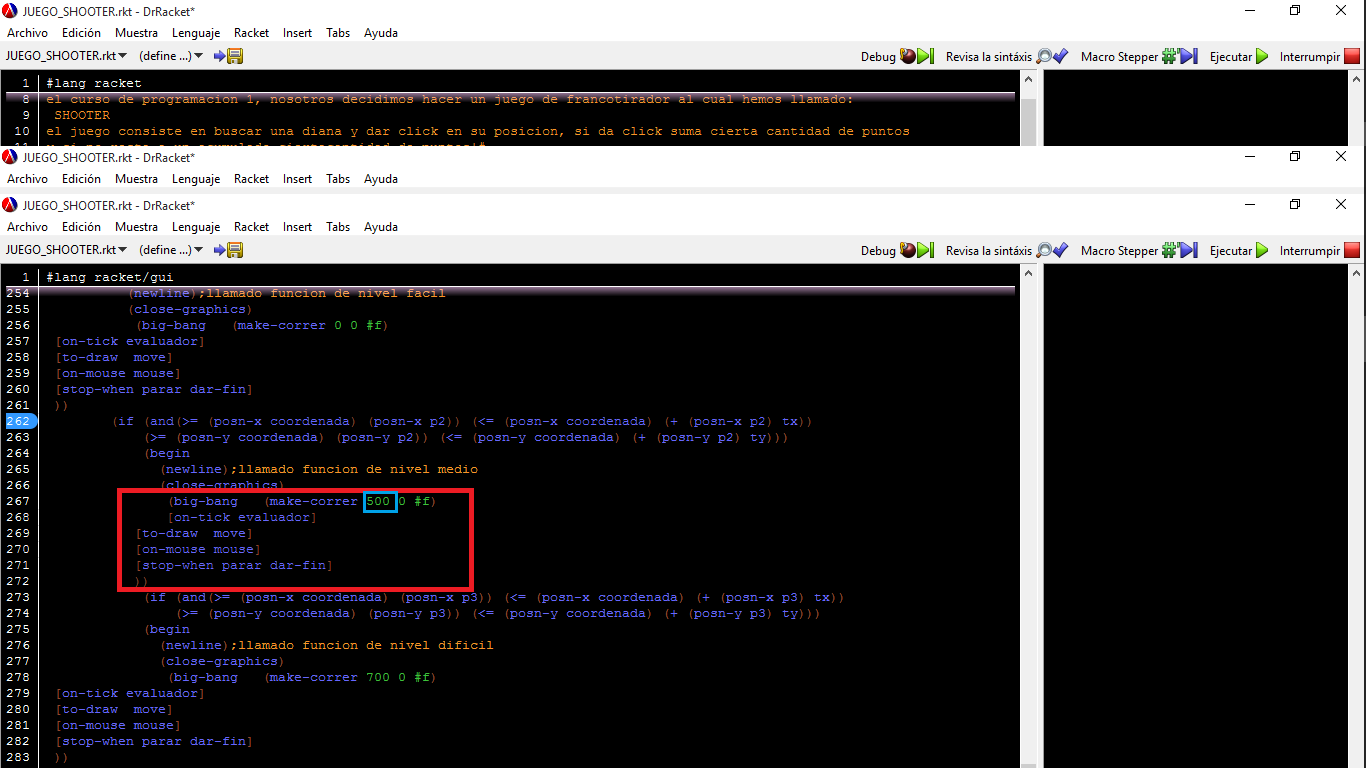


Imagen fin del juego.



La función que captura todo lo relacionado con el mouse, sin decir que la más importante, es la función “(mouse m x y estado)” y lo que hace esta función es que evalúa el estado del mouse, si está moviéndose por la imagen o si se hizo clic en esta, si el estado es “move” (moviéndose en pantalla) la función captura la posición del cursor y lo envía al vector “posmov” y los otros valores de la estructura quedan igual, y si el mouse hizo clic en la imagen lo que hace es capturar esa posición y enviarla al vector “posclick” y cambia “correr-click” por un #t diciendo así que se hizo clic en la imagen y dando paso a las otras funciones que necesitaban saber si se estaba moviendo o se estaba haciendo clic. Por ultimo esta función si no se está moviendo el mouse o no se hace click no hace nada con la estructura y la devuelve tal y como llego.

El llamado para estas funciones se hace dentro del menú que está definido por la función “(clic-rectangulo tx ty col p)” el llamado de estas funciones se hace con un big-bang que es una función recursiva que permite ejecutar todo lo dicho anteriormente al mismo tiempo, entonces el llamado se hace de la siguiente forma:



El recuadro azul está seleccionando “correr-barra” (el primer “correr-barra” que recibe la función), es decir el momento de inicio de la barra de tiempo así entre este número esté más cerca del límite que se definió anteriormente menor es el tiempo, “(to-draw move)” se encarga de los dibujos, en este caso del rectángulo que simula el tiempo y de las imágenes puestas en capas “(on-mouse mouse)” es el encargado de leer las operaciones del mouse, “(on-tick evaluador)” es el encargado de enviar la estructura a la función que los modifica (“evaluador”)

Y por último “(stop-when parar dar-fin)” esta función lo que hace es que da fin cuando la función “parar” sea verdadera, por lo tanto llama la función “dar-fin” y ejecuta todo lo relacionado con esta función, todo esto se hace al mismo tiempo gracias a la función “big-bang” y a la librería “2htdp/universo”

El llamado principal es el menú hecho con graficos que evalua si se hace click dentro de los cuatro cuadros que hay, la función “(hacer-click-dentro tx ty p p2 p3 coordenada)” evalua si el clic esta dentro de uno de los cuatro botones del menu, de ser asi ejecuta el “big-bang” acorde al nivel o cierra todo si la opción fu salir, también cierra los graficos dando paso solo a las imágenes.

La función que hace el llamado principal es entonces “(click-rectangulo tx ty col p)” que da inicio al mouse y captura la posicion del clic para que la función “(hacer-click-dentro tx ty p p2 p3 coordenada)” pueda funcionar.

Lo único que queda es dar click en ejecutar y disfrutar del juego SHOOTER.

Lista de archivos

Al descargar el juego usted obtendrá los siguientes archivos

* JUEGO\_SHOOTER.rkt
* Carpeta: captcha
* captctcha1.png
* captctcha2.png
* captctcha3.png
* captctcha4.png
* captctcha5.png
* captctcha6.png
* captctcha7.png
* captctcha8.png
* captctcha9.png
* captctcha10.png
* captctcha11.png
* captctcha12.png
* captctcha11.png
* captctcha11.png
* captctcha13.png
* captctcha14.png
* captctcha15.png
* captctcha16.png
* captctcha17.png
* ShooterLogoWhite.png
* Carpeta: imágenes
* Diana.png
* fondo.png
* miradificil1.png
* mirafacil.png
* miramedio.png
* pared-de-ladrillo-173533.jpg
* puntuacion-4.png
* transparente.png
* Variable\_Zoom\_Scope\_Retricle\_BOII.png
* Carpeta: manuals
* Manual usuario.docx
* Manual programador.docx
* Carpeta: Sonidos
* acierto.wav
* acierto2.wav
* acierto3.wav
* fallo.wav