

Descripción

Consiste en recrear el arcade BreakOut de Atari de 1976 para el microcontrolador de 8Bits Arduino. He ido un paso más y he intentado añadir patrones en el fondo de pantalla, un marco más elaborado y distintos niveles (Como los de Arkanoid).

BreakOut es un videojuego arcade desarrollado por Atari, Inc. y lanzado al mercado el 13 de mayo de 1976. Fue creado por Nolan Bushnell, Steve Wozniak y Steve Jobs. Hay una anécdota curiosa con respecto a este juego.

Atari impuso un desafío a sus empleados. Había desarrollado BreakOut para combatir la emergente competencia en el mercado de videojuegos, pero los costos de producción eran muy altos a causa de la cantidad de chips necesarios.

La empresa ofreció **750 dólares por rediseñar el juego**, y un bono adicional de **5.000 dólares si la cuenta total era menor de 50**.

Un detalle: una de las razones por las que Jobs pudo dar la apariencia de que sí conocía de ingeniería, es porque su amigo y posterior cofundador de Apple, Steve Wozniak (que sí era un ingeniero dotado), a veces pasaba las horas del turno nocturno con él, ayudándolo a solucionar problemas técnicos.

Jobs le ofreció a Woz repartirse el dinero del bono si lo ayudaba con el tema. Ambos no solo lograron bajar la cantidad de chips desde 100, sino que la versión que **produjeron de BreakOut tenía solo 47 chips**.

Pero resulta que **Jobs no le contó a Woz acerca del bono extra de 5.000 dólares**, por lo que **solo se repartieron en partes iguales los 750 dólares** entre los dos.

Debido a la poca memoria de Arduino Uno, tan solo 2KB SRAM, hemos recreado 15 niveles. Los gráficos se han creado de forma procedural y no como bitmaps. Si bien se podría optimizar muchísimo para dar cabida a los 35 niveles de Arkanoid sin problemas, haciendo que la definición de los niveles fuera de 4bits en vez de los 8bits que estamos usando, eliminando partes que no aportan información como por ejemplo cuando no hay ladrillo, y un largo etcétera, pero entonces perdería la función didáctica y sería un código mas eficiente pero más ofuscado.

Los niveles se definen en funciones, en vez de en variables globales, por si se quiere dar un comportamiento procedural que ahorraría SRAM, pero bueno ahora mismo nadie nos paga 5.000 dólares por optimizar.

Materiales

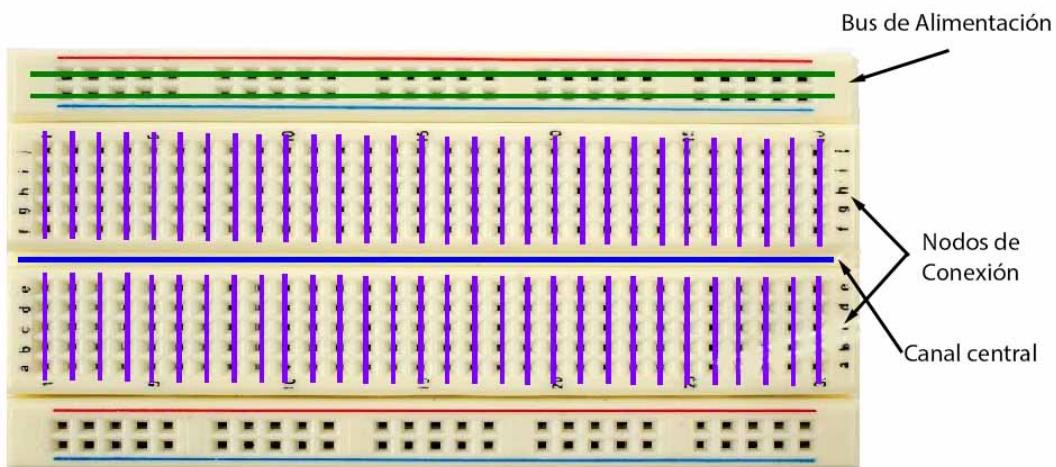
Mostramos la lista de componentes necesarios, junto a un valor aproximado de su coste y una breve descripción de su utilidad. Como podéis observar este proyecto es de muy bajo coste rondando los 15,00€ dependiendo de donde compremos los componentes. Se pueden encontrar en **Amazon** y en **AliExpress**.

Componente	Descripción	Coste
Arduino Uno	Placa microcontroladora encargada del funcionamiento del juego.	5,00 – 10,00 €
Protoboard	Placa de prototipado de conexiones simples sin soldadura.	2,00 – 5,00 €
Cables	Cables macho-macho, macho-hembra, hembra-hembra para conexiones sin soldadura.	3,00 €
Potenciómetro Lineal	Resistencia variable que se ajusta con un dial. Limitan el paso de corriente eléctrica. Tipo lineal para que el control se comporte uniformemente en todo su espectro.	1,00 €
Botón	Micro interruptor, permite el paso de corriente eléctrica. Circuito cerrado, circuito abierto.	1,00 €
Resistencia	Controla el paso de corriente eléctrica para evitar cortocircuitos. En este caso usaremos 1K para unirla al botón.	0,50 €
Buzzer Pasivo	Altavoz que permite convertir una señal electriza en una onda de sonido.	0,50 €
TFT SPI 1.8"	Pantalla TFT de 1.8" con controlador ST7735 y resolución 128x160 pixels.	4,00 - 8,00 €
AZDelivery	(https://www.amazon.es/dp/B078J5TS2G)	

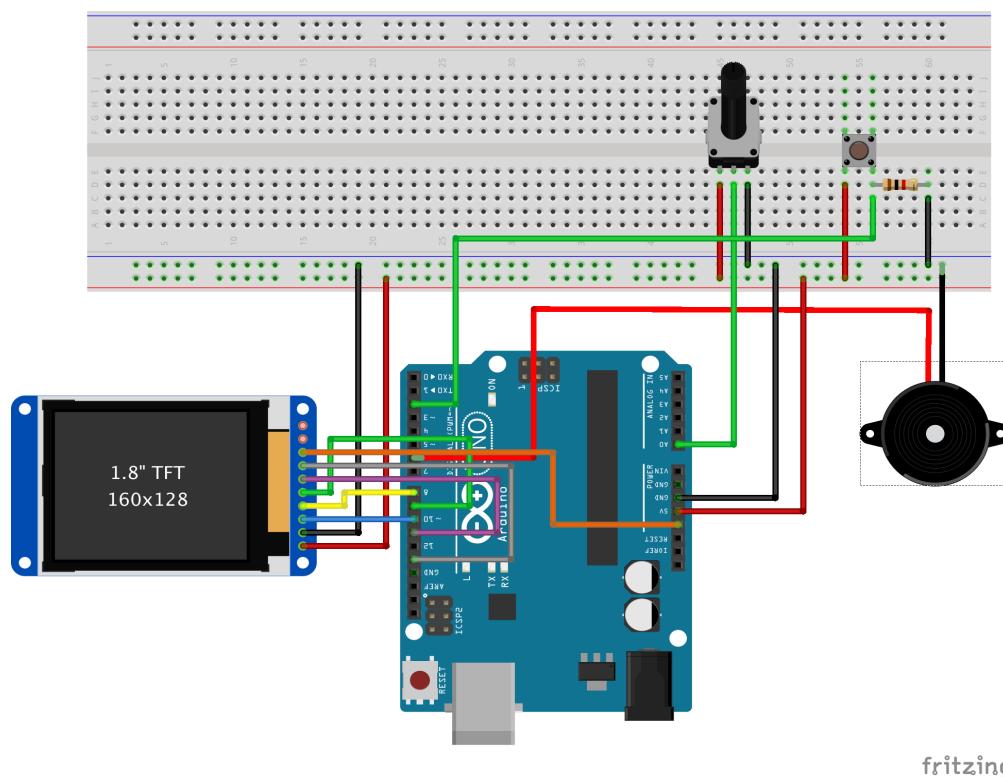
Conexiones de los componentes

Una imagen vale más que mil palabras... Os pongo una imagen del diseño de la conectividad de los componentes. Tened especial cuidado con la conexión en la Protoboard ya que si no conocéis su conectividad podéis provocar fácilmente un cortocircuito (Especial cuidado con la **conexión del botón**)

Conexiones de una Protoboard



Conexiones de BreakOut



fritzing

Programación

Tanto el código como la electrónica se han diseñado con **fines didácticos**, no se ha buscado la **eficiencia**. Simplemente se ha intentado hacer algo fácil de entender.

Siempre he pensado que el mejor maestro no es el que más sabe, sino el que mejor enseña lo que sabe.

Antes de ponernos a descargar el código deberíais tener instalado el **entorno gratuito** de desarrollo de Arduino (<https://www.arduino.cc/en/software>), este software es libre y de muy sencilla utilización.

Una vez descargado el IDE de Arduino os tendréis que descargar las librerías necesarias para la controladora de TFT, en mi caso he usado un ST7735 (si usáis un TFT distinto al que yo he usado, SPI 1.8" AZDelivery deberéis descargar otras librerías y realizar otras conexiones) En mi caso las librerías necesarias han sido:

- Adafruit ST7735 and ST7789 Version 1.6.0
- Adafruit GFX Version 1.10.5
- Adafruit BusIO Version 1.7.2

El código del juego se ha creado para poderse adaptar a resoluciones distintas de TFT es bastante sencillo y está casi todo el proceso realizado ya.

En la zona de macros encontrareis una de DIP Switches que os he puesto por si queréis modificar parámetros como la velocidad de juego, la puntuación de bola extra, ...

Para evitar que la bola se quede en un bucle del que no poder salir, se ha creado un contador que controlará el tiempo que lleva sin ser golpeada por la nave, y si pasa ese límite, nos la pondrá de nuevo en el punto inicial.

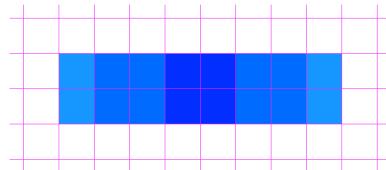
Los rebotes son poco fieles, debido a la carencia de página oculta (Framebuffers o similares), hemos adoptado una postura menos real, pero mas funcional puesto que debemos repintar la zona donde la nave, ladrillos y bola se encuentran. Para evitar parpadeo al mover la nave, se ha implementado un pequeño clipping de repintado.

Nave de 8 segmentos

Para el rebote con la nave se ha implementado el famoso esquema de 8 segmentos de Pong, si bien le hemos añadido que no pueda rebotar con inclinación cero, haciendo que el rebote sea de 22.5, 45 y 67.5 grados respectivamente.

En este diagrama podéis ver el comportamiento inicial de los 8 segmentos. La pala se dividía en 8 segmentos y cada zona provocaba un ángulo de rebote.

- Rebote 45°
- Rebote 22,5°
- Rebote Recto



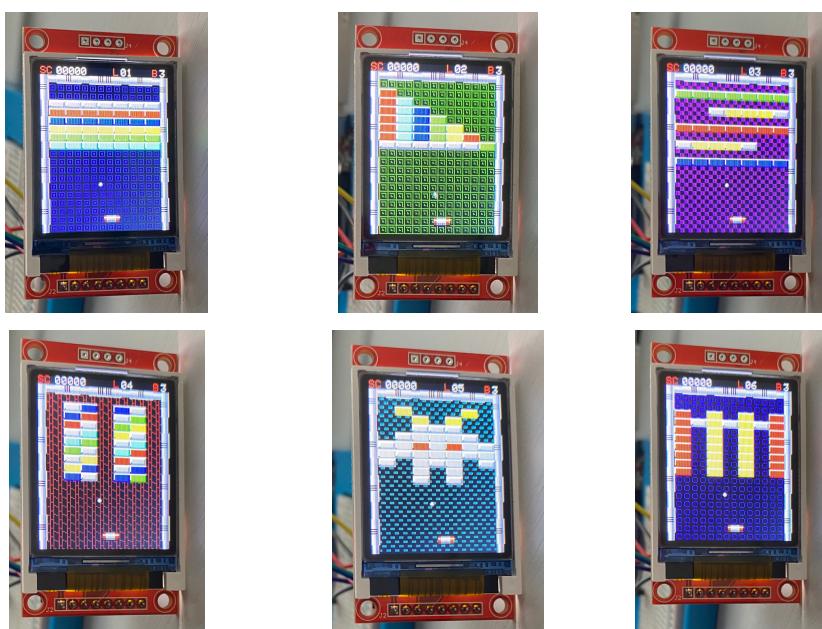
En este caso hay que sumar 22.5 grados mas para cada segmento.

Cabinet

Lo ideal para terminar este proyecto sería crear un cabinet para alojar los componentes y darle un aspecto profesional. Hoy en día con las impresoras 3D y los vinilos creo que es bastante sencillo de realizar.

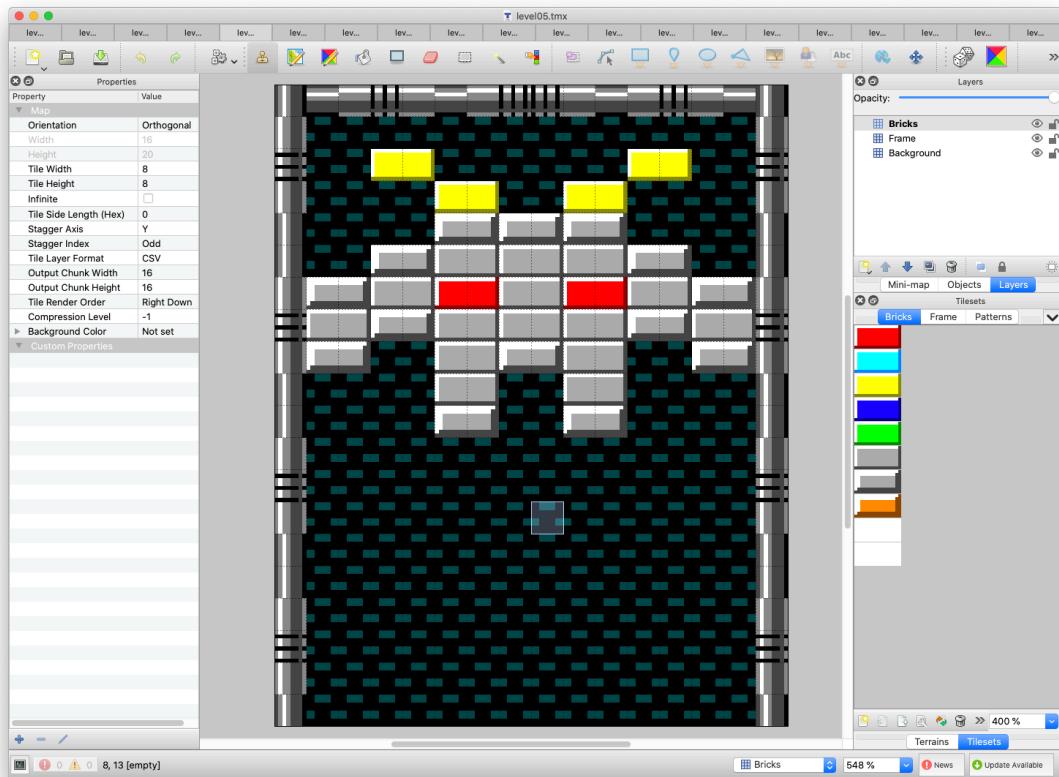
Galería de fotos

Aquí tenéis una muestra de los 6 primeros niveles y como quedan en pantalla.



Para la creación de los distintos niveles, he optado como hago siempre por Tiled una gran aplicación multiplataforma que te permite ver y tener organizado los distintos niveles de tu juego, una pequeña maravilla que os recomiendo.

Aquí os pongo un ejemplo de como se ve



Junto al código os dejo los 20 niveles que había diseñado en un primer momento en formato tmx.

Video del proyecto

Podéis ver un **video** con el prototipo en funcionamiento en mi cuenta de YouTube.

<https://youtu.be/Ju0PVx5k7nA>

Próximo proyecto

La verdad es que ha sido un proyecto breve menos que Pong, pero también breve, he tardado unas 25 horas en 7 divertidos días.

Tengo en mente Asteroids, Tetris y Plotting.

Espero que lo disfrutéis, es divertido crearlo y jugarlo.