Cannon Duels

Game Manual

David Asta - 2018

Introduction

Cannon Duels is a 1 or 2 players game for Amstrad CPC, developed with CPCtelera and other tools for the #CPCRetroDev 2018. The Source code and assets are licensed under MIT License, as explained in the file LICENSE.

The game is quite simple. Two Cannons are located at each side of the screen and can shoot at each other. At the centre of the screen, between the Cannons, there is a group of bricks. Cannon Balls will destroy the first brick they encounter in their path or subtract a live from the opposite player if hitting the Cannon.

When hitting a Brick there is chance to get a Power Up. Each one is activated by random and have different properties, as explained in a later section of this manual.

When a Cannon is hit by a Cannon Ball, it loses a live. Reach 0 and the game is over.

A player gets an Extra Live every time that collects 464 points.

Which is your game style? Get as many points as possible and collect so many Extra Lives that you can go for lunch meanwhile your opponent tries hopelessly to destroy you? Or maybe you are so good that you don’t need Extra Lives, but you want to see if you can make the game to crash by collecting more points than an 8-bit machine can take? Or will you just aim for your opponent and destroy it before it can collect any points and never know what getting an Extra Live feels like? Do you want to beat the computer or do you prefer to play against a human opponent?

Loading the Game

Load cduels.dsk and type: *run"loader.bas*

Controls

Game Menu

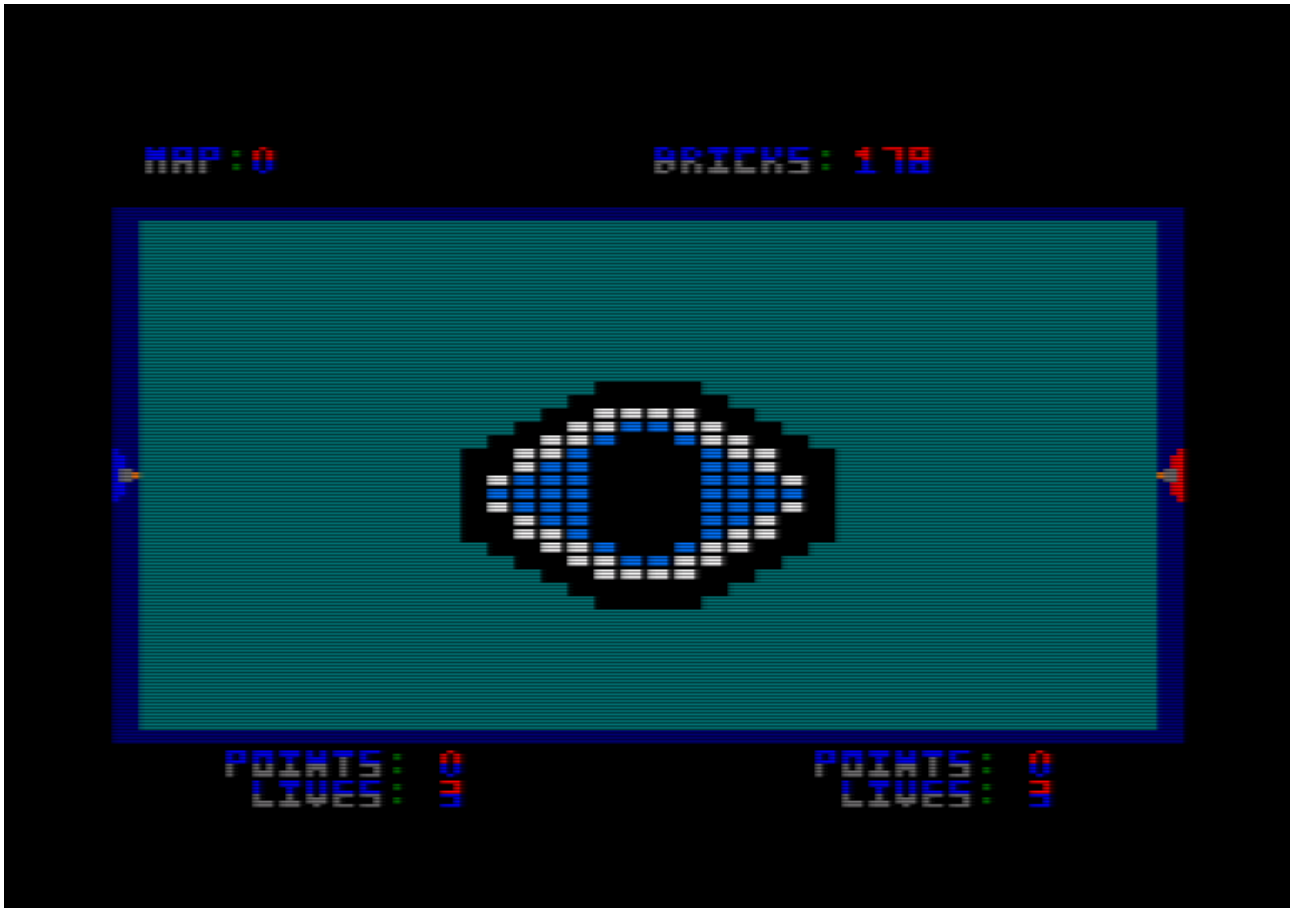
- **1** = Human Player (Left Cannon) against Computer AI (Right Cannon).
- **2** = Human Player (Left Cannon) against another Human Player (Right Cannon).

In Game

- **A** = Move left Cannon Up.
- **Z** = Move left Cannon Down.
- **X** = Shoot from left Cannon.
- **1 (numeric pad)** = Move right Cannon Up.
- **0 (numeric pad)** = Move right Cannon Down.
- **. (numeric pad)** = Shoot from right Cannon.
- **ESC** – Aborts game and returns to Game Menu.

Bricks, Points, Lives and Power Ups

At the centre of the screen there is group of Bricks forming a brick map.



When a Cannon Ball hits a Brick, the player receives a certain amount of Points depending on the Brick colour.

Black = 1 point	Blue = 2 points	Bright Blue = 3 points
Red = 4 points	Bright Red = 5 points	Green = 6 points
Cyan = 7 points	Sky Blue = 8 points	Yellow = 9 points
White = 10 points	Orange = 11 points	Bright Green = 12 points
Bright Cyan = 13 points	Bright Yellow = 14 points	Pastel yellow = 15 points
Bright White = 16 points		

Every time a player collects 464 points, gets an Extra Live.

When a Cannon Ball hits a Cannon, it will lose a live. When the number of lives reaches 0, the game ends and the standing player wins.

Also, by hitting a Brick the player have a chance to activate a Power Up for a limited amount of time:

	**Double Points**	Get double points for each Brick that is hit. Perfect for accumulating extra points quickly.
	**Triple Points**	Get triple points for each Brick that is hit. This is the best way to get to the next Extra Live.
	**Slow Cannon**	Cannon moves slower. Watch out! You are now easier to hit.
	**Freeze Cannon**	Cannon cannot move. But can still shoot. This is the moment your opponent will most certainly choose to eliminate you.
	**Jam Cannon**	Cannon cannot shoot. What could be more frustrating than not being able to collect points or destroy your opponent?
	**Slow Cannon Ball**	Cannon Balls travel slower. Those precious seconds you are loosing will certainly go in favour of your opponent.

Nod to Chicagos’30 by Topo Soft

A nod (or homage) to Chicago’s 30 by Topo Soft is included in the game, as a Game Over screen.

Big thanks to

- Amstrad Plc, for creating the best 8-bit computer ever.
- My father, for buying my first computer, an Amstrad CPC464 and later the Amstrad CPC6128 which I still use nowadays.
- My girlfriend, for her help testing and for her cool ideas.
- Fran Gallego, for creating the CPCtelera and give me the inspiration to start developing my first game.
- To all people that somehow is part of the #CPCRetroDev, for creating something so cool and give the Amstrad scene the push it needed.