



Arkanoid

The game

Staff 42 bocal@staff.42.fr

Summary: This document contains instructions for the Arkanoid.

Contents

I	Forewords	2
II	Subject	3
III	Bonus	5
IV	Instructions	6
V	Notation	7
VI	Citation	8

Chapter I

Forewords

Arkanoid is a breakout arcade game developed by Taito in 1986 and has been used on many family supports. The game is based on the same principle as Breakout by Atari Inc. (1976).

Chapter II

Subject

The purpose of this rush is to recode the Arkanoid game in C using the OpenGL Framework

You will need to read or re-read the **GLFW** doc, it's an open source framework that wraps OpenGL very well.

The principle: *The player moves from right to left a horizontal bar meant to represent a spaceship. This board, positioned at the bottom of the screen, allows you to bounce a ball that will destroy blocks at the top of the screen. The player moves to the next level when all blocks are destroyed (except some indestructible blocks); the player loses a life every time he lets the ball spin at the bottom of the screen ..*

The Blocs:

- the standard block touched by the ball is destroyed in one shot.
- the block that must be touched several times to be broken (from 1 to 3 strokes, depending on the initial state of the block).
- the indestructible block.

Technical considerations:

- You must display the score in real time on the OpenGL window! ... not in the terminal, nor at the end of the game.
- You must have a git submodule pointing to the github of the `texttt` GLFW
- A single `make` initializes the submodule, compiles the GLFW and compiles your binary.
- You must have a `texttt` levels folder that will contain all the levels in the text format you want.
- The levels will be processed at launch, ie if I do not have a level I do not run the program and I quit it cleanly.
- The ESC key must leave the game cleanly.
- You are allowed to use your `texttt` libft. You can use other functions if they are needed for your bonuses.

Chapter III

Bonus

Bonuses will only be assessed if the mandatory game is perfect.

- Blocks with bonuses
- The Multi on the same keyboard, one at the top, one at the bottom, the blocks in the middle
- Change the size of the window with a UI that adapts
- A menu (resembling a real one..)
- Management of scores in a file with nicks.

Chapter IV

Instructions

- You are free to organize and name your files as you wish.
- The executable must be called `texttt arkanoid`.
- You must make a Makefile.
- Your Makefile must compile the project, and must contain the usual rules. It should only recompile the program if necessary.
- Your Makefile must compile using the flags `-Wall`, `-Wextra` and `-Werror`.
- If you are smart and you use your libft, you must copy the sources and the associated Makefile into a folder named `libft` which should be at the root of your rendering repository. Your Makefile will compile the library by calling its Makefile before compiling your project.
- Your project must be written in accordance with the Norm, an exception will be accepted for GLFW callbacks that take `+4` arguments.
- In no case should your program quit unexpectedly (segmentation fault, bus error, double free, etc.).
- Any memory allocated on the job must be freed properly.

Chapter V

Notation

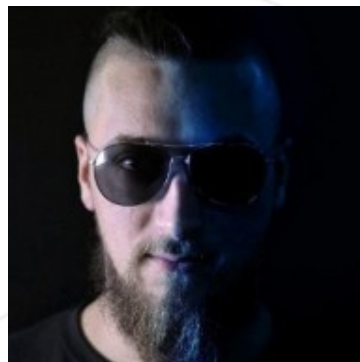
- If you miss one of the main features of the game, you risk not validating the rush.
- The two members of the pair should be able to explain a point of detail of the code or the operation of the program as a whole. The correctors are encouraged to ask questions to each of the two team members.
- We remind you that the correctors will have to scrupulously apply these rules:
 - Incomplete makefile or relink: 0;
 - Norm error, on any part of the rendered code: 0;
 - Segfault, even if caused by adding a bonus: 0;
 - Instructions in the chapter Instructions not respected: 0.

Chapter VI

Citation



@ZAZ GLOIRE AU TAS !!!!!!!!!!! #PONEY #CAT #LIKEMYEXGIRLFRIEND



Bon courage !