7-5-2023

# Arkanoid

## Semestral project

David Casajus Calvet

ZTIAP - INTRODUCTION TO INTERACTIVE APPLICATIONS DEVELOPMENT

# Content

## Basic goals and functioning of the game

The goal of the Arkanoid game is to use a paddle to bounce a ball and break all the bricks in the level without letting the ball fall off the bottom of the screen. The player must use the paddle to keep the ball in play and direct it towards the bricks, while avoiding obstacles and collecting power-ups that can help to clear the level more quickly. The game is won when all the bricks are destroyed.

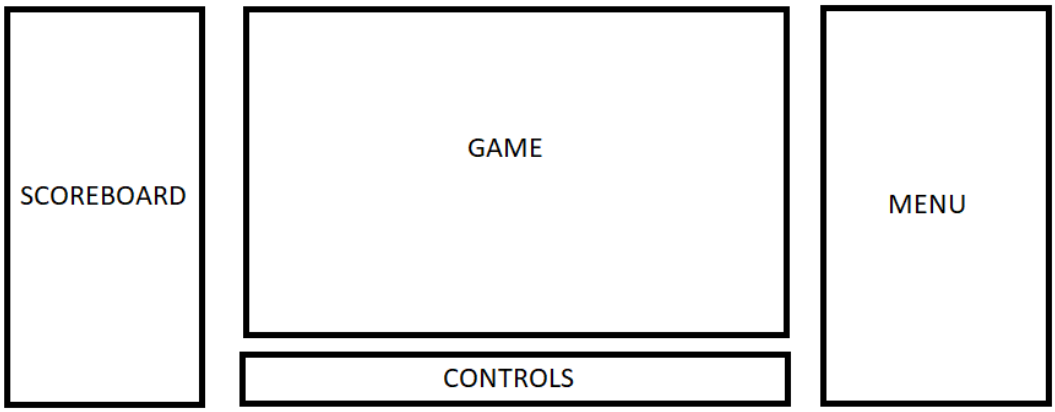
In my version of the game, I have added bricks that when broken grant bonuses to the player, such as increasing the speed of the ball and increasing or decreasing the size of the paddle.

## Description of game controls

The paddle can be controlled both with the left and right arrows on the keyboard and with the two buttons displayed on the screen.

## Design of all game screens and menu

The game has a single screen that is divided into three zones, on the left side the instructions of the game are shown. In the central part is the game itself, with the button controls at the bottom and the music controls. And on the right side is the scoreboard.




## OOP description of the game objects

- Ball: object that represents the ball of the game.
    - Attributes: radius, speed, X position,Y position, angle
    - Methods:
        - Constructor: starts in the center of the screen with random direction down
        - Draw: draws the ball in the canvas
        - RegisterObserver: Add an observer.
        - UnregisterOberserver: remove an observer.
        - NotifyObservers: notify all observers with the position of the ball.
- Objective: each brick of the game is an instance of a objective
    - Attributes: color, width, height, layersImage, upgradeImage, X position, Y position, layers, upgrade.
    - Methods:

        - Constructor: set the position of the objective, the number of layers and the upgrade if it has.
        - Draw: draws the objective in the canvas.
- Platform: represents the paddle of the game
    - Attributes: width, height, speed, imgSrc.
    - Methods:
        - Constructor: creates the platform
        - Draw: draws the platform in the canvas.
        - MovePlatformByEvent(e): moves the platform when the arrows are used.
        - MovePlatformByButton(direction): moves the platform when the buttons are used.
- Game: In charge of game logic and rendering
    - Attributes:fieldHeight, fieldWidth, objectiveRows,obectivePerRow, winScore, keepPlaying, score, damage.
    - Methods:
        - Window.onload: initialice all the objects needed and setup the game for playing.
        - Render(context, platform, objectives, ball): calls the draw method for all items needed and calls the render method. Calls requestAnimationFrame to repaint the screen. This method is called in a loop until the game is won or lost.
        - Core(context, platform, objectives, ball): this method is in charge of the game logic, calculates if there is a collision between the ball and any of the objects. Calculates the direction of the ball. Removes objectives on collision and activates the modifier if the target had it. Play the necessary sounds. Check if the user wins o losses.
        - Finish(contex, win): Ends the game if the player win/lose.
        - CreateObjectives(): create all the obectives before the game starts
        - DrawObjectives(objectives,context): draws all the objectives of the given objectives list
        - Notify(x,y): notify method that ball calls on notifyOberservers. Shows a message with the position of the ball in the console, raises the score and plays the thump sound.

# Graphic and sound elements of the game

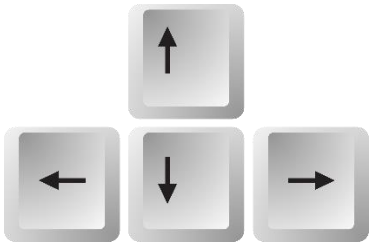
## Graphic elements

Website icon: https://www.flaticon.com/free-icon/arkanoid_2285752

Other assets: https://opengameart.org/content/basic-arkanoid-pack

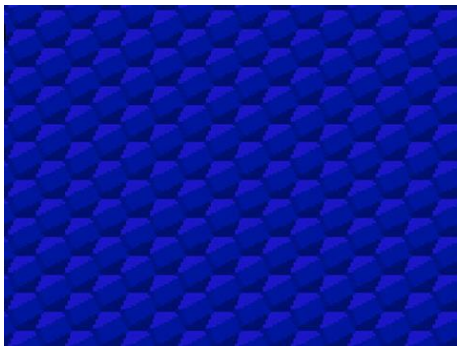
Platform

Arrows of the instructions

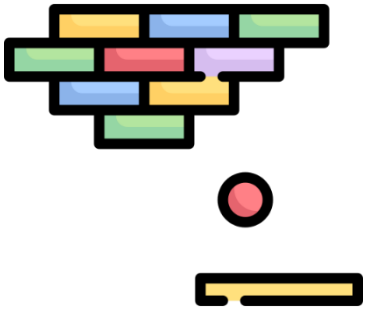


Objectives


Background image



Web icon



## Sound elements

All sound are from the next website:

https://www.sounds-resource.com/nes/arkanoid/sound/3698/

Ball collision sound


Arkanoid SFX (1).wav

Objective hit sound


Arkanoid SFX (2).wav

Fail sound


Arkanoid SFX (3).wav

Win sound


Arkanoid SFX (9).wav

Game music


Intergalactic Odyssey.ogg

# Description of interesting parts of the implementation

## MVC

I've used MVC because it's a popular software design pattern that helps separate different components of a web application. The 'M' stands for model, which represents the data and logic of the application. The 'V' stands for view, which is responsible for rendering the user interface. And the 'C' stands for controller, which handles user input and updates the model and view accordingly.

By using MVC, I can write code that's easier to maintain and modify, since each component has a clearly defined responsibility. It also helps with code organization and reusability, since I can reuse the same model and view components in different parts of my application. Overall, using MVC makes it easier to develop and scale web applications, which is why it's such a popular design pattern in the industry.

## Observer pattern

I've used the Observer pattern because it's a useful way to establish communication and synchronization between different objects in a system. The pattern involves two types of objects: Observers and Subjects. Observers are interested in some aspect of the state of the Subjects and register themselves to be notified when that state changes. When a Subject's state changes, it notifies all of its registered Observers automatically.

This makes it easier to add new functionality or modify existing behavior without having to change multiple parts of the system at once. Additionally, the Observer pattern can help improve performance, since only the relevant Observers are notified when a change occurs, rather than having to update all objects in the system indiscriminately. Overall, using the Observer pattern can lead to more flexible, modular, and maintainable software.

In my case I have used it to notify the game when the ball has collided with one of the bricks in the game, so that I can deal with what should happen, also showing the position of the ball on the console.

### Project directory structure

I have separated the files into 4 folders, one for the images used for the game elements, another where the css file with the web styles is located, another with the sounds used in the game, and finally the folder where they are found. all the javascript files that make the game work.

Also, there is only one html file, which is located in the root of the project. Within the html there are 3 sections, the left one with the game instructions and the reset button, the central section that contains the game canvas, and the right part that contains the scoreboard.

### Screen design for reduced screen

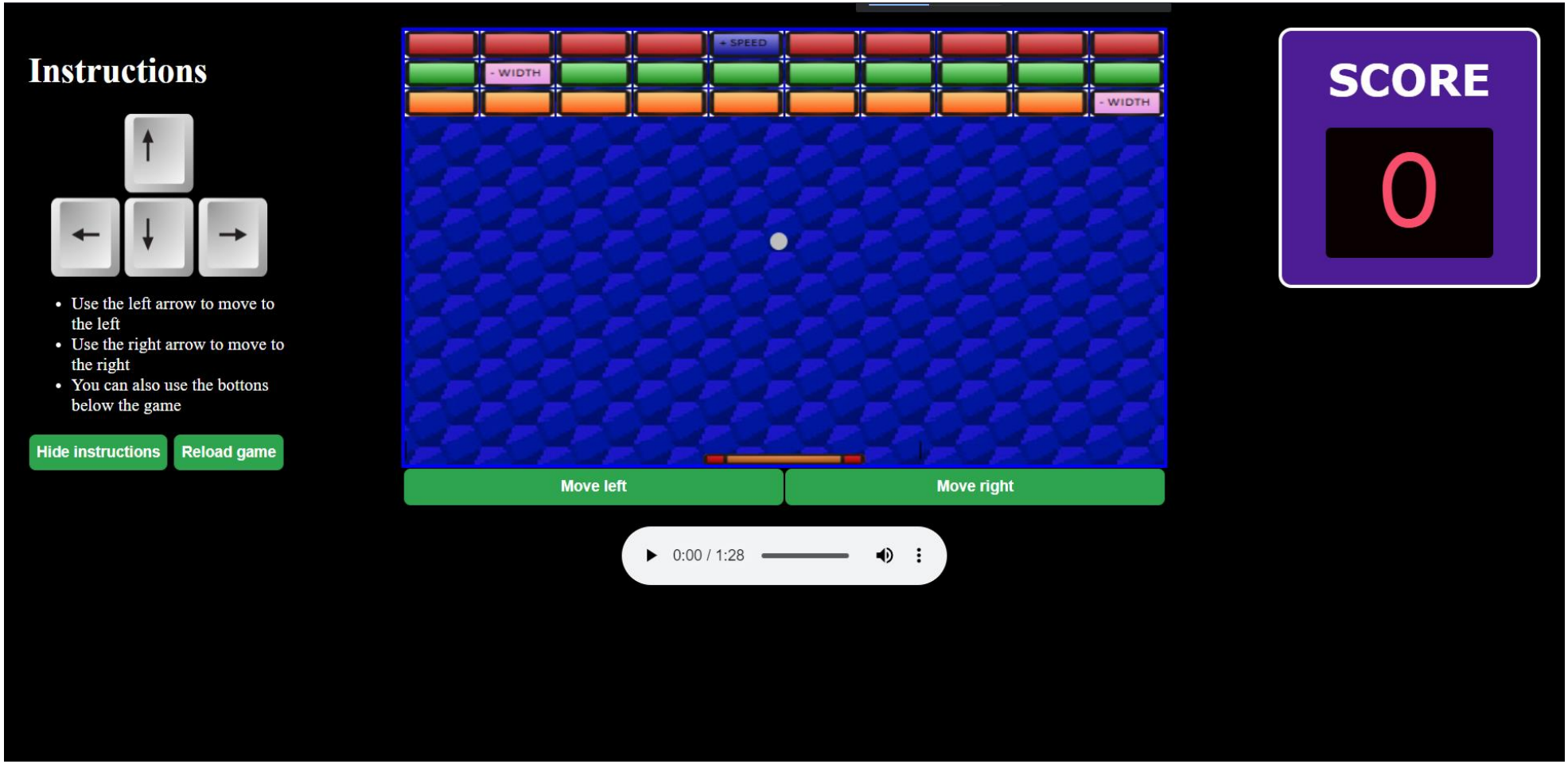
In case the screen is less than 1200px wide the side menus are hidden, so that the game fits well on the screen.

## Evaluation and comparison with the design

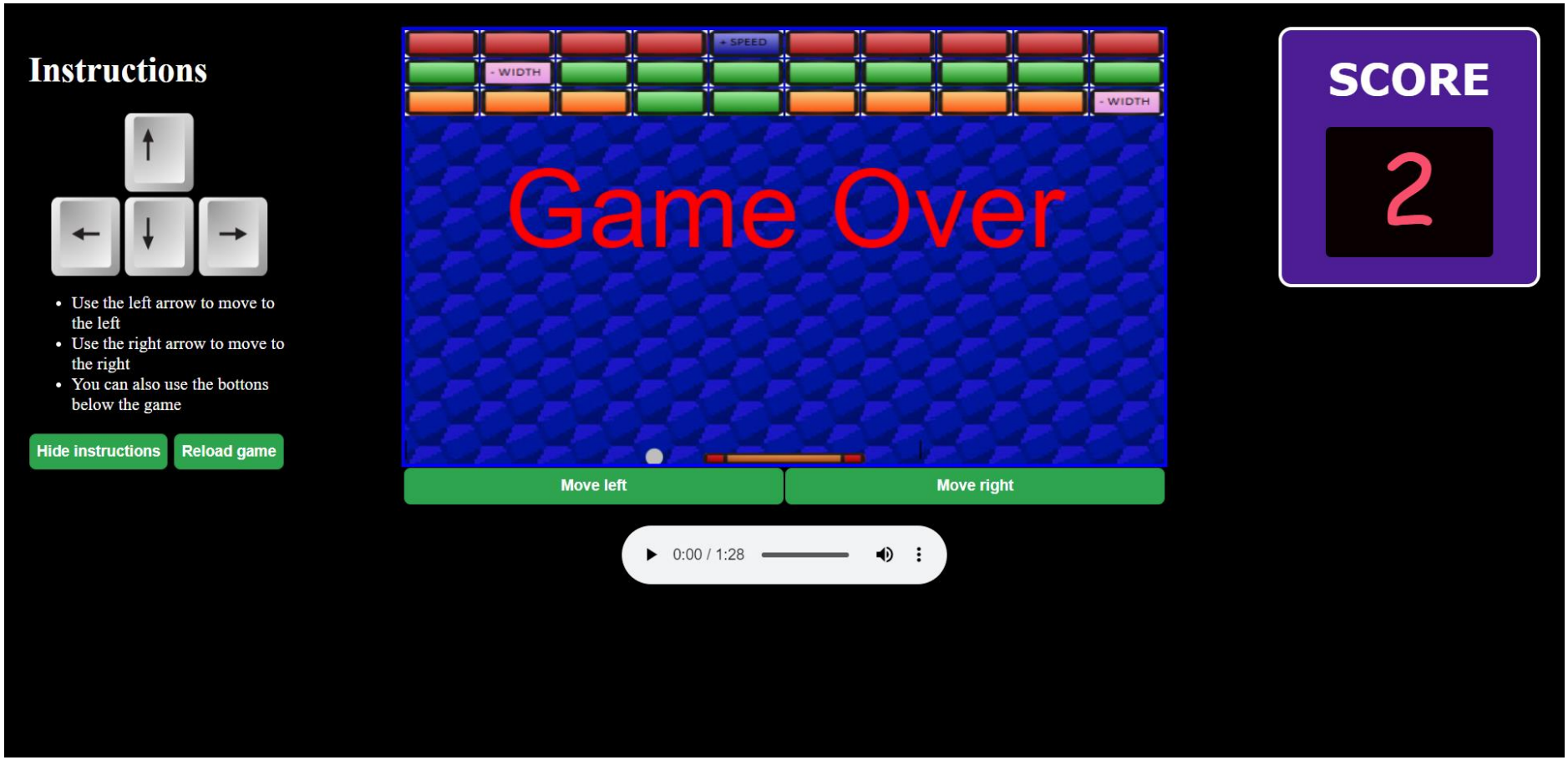
As it is a simple and well-known game, the finished game is practically the same as the initial design of the game. I have only changed the instructions and the scoreboard. Since when reading from left to right it seemed more logical to me that the first thing to see is the instructions.

The entire game runs on the same screen but changes depending on what happens. The first screen is the one that is shown during the game, and the next two are those that correspond to having lost or won the game. In addition, as I have mentioned before, in case you are playing from a small screen, the side columns are eliminated to leave only the game visible and that it can be expanded to the entire screen.

## Start screen



## Lose screen


Instructions
Use the left arrow to move to the left
Use the right arrow to move to the right
You can also use the bottons below the game
Hide instructions
Reload game
+ SPEED
- WIDTH
- WIDTH
Game Over
Move left
Move right
0:00 / 1:28
SCORE
2

## Win screen


Instructions
Use the left arrow to move to the left
Use the right arrow to move to the right
You can also use the bottons below the game
Hide instructions
Reload game
You Win
Move left
Move right
0:00 / 1:28
SCORE
57

## Game screen for reduce size of windows