# 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.

Una pantalla de un video juego

Descripción generada automáticamente con confianza media

# 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
* 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

# Graphic and sound elements of the game

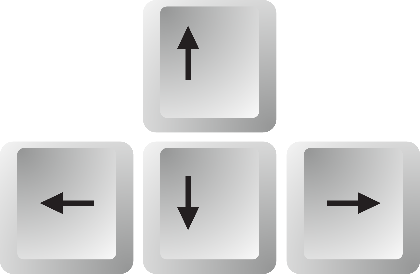
## Graphic elements

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

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

Platform

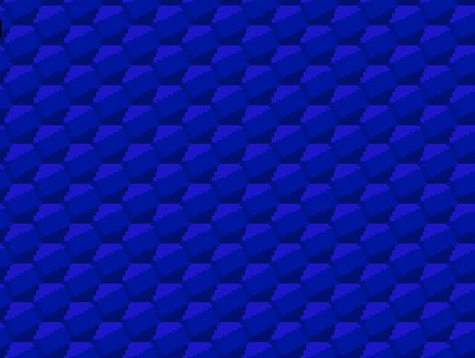
Arrows of the instructions



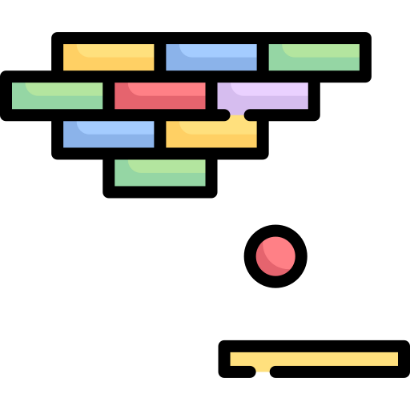
Objectives



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



Objective hit sound



Fail sound

### 

Win sound



Game music

