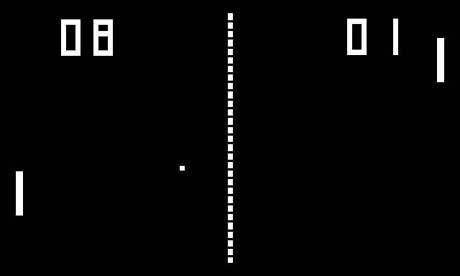
Project #2 – Pong

# Overview

Created in 1972 by Atari incorporated, Pong is one of the earliest video games. The game simulates a very simple table-tennis match involving two players. A ball (a moving pixel) moves back and forth across the screen horizontally and paddles (vertical bars) can hit the ball back and forth. <https://www.youtube.com/watch?v=5uuxFhUc8tg>



# Specifications and grading

## Detailed requirements

### Variables

* Each player’s score and the number of remaining balls are displayed.
* Each player’s score starts at 0.
* The number of balls remaining starts at 3.
* The score for each player increments every time the ball crosses the plane of the paddle (a goal for the opposing player)
* Number of balls remaining decrements every time a new ball is put in play

### Ball Sprite

* The beginning of a round
  + The ball starts each round (including the first) at the middle of the screen.
  + The ball points in a random direction between 0 and 359 (a full circle).
  + Play begins and the ball begins moving after 3 seconds.
* Movement
  + The ball moves 2 steps per frame (per iteration of forever or repeat).
* Bouncing off a paddle
  + When the ball touches a paddle, it bounces off the paddle as a typical physical ball would. Specifically, the angle of incidence of the ball approaching the paddle must equal the angle of reflection of the ball leaving the paddle.
  + The calculation of the new angle and pointing the ball at that angle **must** be implemented as a custom block named **BounceOffPaddle**.
  + Hint: Both paddles are vertical. Consider how that assumption can make the calculation simpler.
* Edge bounce
  + The ball must bounce off the top and bottom parts of the playing field.

### Human Paddle Sprite

* Position
  + The human’s paddle sprite must move within a single x position close to the right side of the stage.
* Movement
  + The human’s paddle must be controlled by the up and down buttons on the keyboard.

### Computer Paddle Sprite

* Position
  + The computer paddle sprite must move within a single x position close to the left side of the stage. (i.e. The same distance from the edge as the human’s paddle, to keep it fair.)
* Movement
  + The computer paddle must track the ball when it comes within a close range. “Track” means move toward the ball’s y position at 3 steps per frame. “Close range” means 1/3 the width of the court.

### When you miss

* Cause
  + The ball qualifies as ‘missed’ when the ball goes past a paddle. In other words, the ball is ‘missed’ when the ball’s y position exceeds a paddle’s y position. “Exceed” meaning more positive or more negative.
* Effect
  + The player who just scored (who did not miss the ball) is awarded 1 point.
  + If one or more balls remain, the ball resets to the center, waits 3 seconds, and launches again. You may recognize this from *“Ball Sprite, The beginning of a round.”*
  + If zero balls remain, “Game Over” and the winner are declared and said. (Use a say block.) The player with the most points is the winner.

### Additional extensions

Once you complete the above, you can extend your program. Some suggestions:

* Instead of a perfectly elastic reflection (the bounce), turn the ball based on where in the paddle it hit. (Hint: By comparing center positions of the ball and the paddle.) This means that you can straighten out the ball’s path to make it easier or make the ball bounce at a higher angle when it hits near the edge of the paddle to make it more difficult.
* Increment the ball speed after every 5 times the ball hits a paddle.
* Alter the computer-controlled paddle so that it can miss the ball.

## Grading criteria

The list of what we will use to grade your projects is summarized below. Please review your project before submitting it to be sure you meet all of the specifications. If you have any questions on whether you meet a requirement, **please ask us.**

|  |  |
| --- | --- |
| **Requirement** | **Possible Points** |
| Variables are initialized as specified. | 10 |
| Ball is properly initialized at start of game and after each miss. | 10 |
| When a player misses the ball, the score increments by 1. | 5 |
| The number of remaining balls decrements by 1 when a new ball is generated. | 5 |
| Ball bouncing off the paddle is implemented as a function/block and exactly as specified. | 20 |
| The ball bounces off the top and bottom of the stage. | 10 |
| When a player misses the last ball (the number of remaining balls is 0), “Game Over” and the winner is announced. | 10 |
| The human paddle sprite moves with the [up] and [down] direction keys; the computer paddle moves with the ball. | 10 |
| Good programming skill: Program is repeatable and initializes state correctly. | 10 |
| Ball bounces smoothly, overall program runs smoothly. | 10 |
| Extra Credit: Ball bounces based on where it hits the paddle | 10 |
| Extra Credit: The computer paddle can miss the ball. | 10 |
| Extra Credit: Make your own extension | 0-10 |
| **Total points** | **100** |