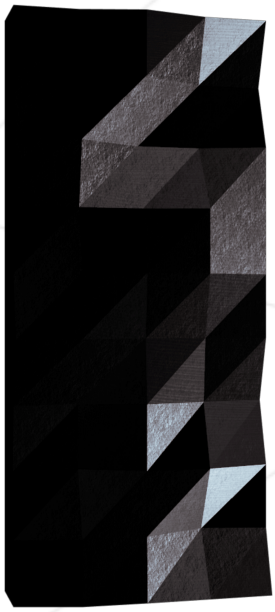

Introduction to Programming - 42

Day 06

Kai kai@42.us.org
Gaetan gaetan@42.us.org

Summary: This document is the subject of the day 06 of the introduction to programming piscine.

Contents

Chapter I

Guidelines

- Corrections will take place in the last hour of the day. Each person will correct another person accoring to the peer-corrections model.
- Questions? Ask the neighbor on your right. Next, ask the neighbor on your left.
- Read the examples carefully. The exercises might require things that are not specified in the subject...
- Your reference manual is called Google / "Read the Manual!" / the Internet / ...

Chapter II

Preamble

Chapter III

Goals

Games are one of the oldest motivations for computer science and still drive the industries of processors and microchips today. What is the goal of learning Ruby other than to make the world your own arcade game?

Connect 4 is one of the many simple grid-based games which we can easily play at the terminal window. You could do the same thing with Tetris, or Sudoku, or word searches, or Checkers... but today we're playing C4.

Can you code a sophisticated Connect 4 AI?

Chapter IV

General Instructions

Create a program `connect4.rb` which allows two users to play Connect 4 on the command line.

After a welcome screen and instructions, your output should look something like this:

```
=== *Connect4* ===
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
Player 1, go!

Player 1 drops a piece on column 1.
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . o . . . . . |
Player 2, go!

Player 2 drops a piece on column 7.
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . o . . . . x |
Player 1, go!

Player 1 drops a piece on column 1.
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . o . . . . . |
| . o . . . . x |
Player 2, go!

Player 2 drops a piece on column 1.
| . . . . . . . |
| . . . . . . . |
| . . . . . . . |
| . x . . . . . |
| . o . . . . . |
| . o . . . . x |
```

... and so on.

Chapter V

Mandatory Part

- The default board has 6 rows and 7 columns.
- The board should display in its updated state after every play.
- Allow two human players to alternate giving commands.
- The first player to connect four pieces in a diagonal, horizontal, or vertical line, wins.
- Print a message when one of the players wins or when there is a tie.
- The program should not throw any unhandled exceptions, have strange behavior or crash!
- Gravity works!

Chapter VI

Bonus Part

In no particular order, choose any/none/all:

- Allow resizing of the board and the streak goal so that we can play Connect 11 on a 42 x 42 board, or Connect 3 on a 4 x 5.
- Code a computer player which contains similar functions as your human players, but makes its own decisions of how to play.
- Coordinate with a friend so that your computer players can each interact with the same board and try to beat each other.
- Can you make the board change dynamically in place in the terminal, instead of each new version printing below the last?

`Google ANSI escape codes.`

- Can you even animate the dropping of the pieces using that technique?!?
- Code another game - try Tic Tac Toe, or Checkers.