



Ruby on Rails Training - Rush 00

Moviemon

Stéphane Ballet balletstephane.pro@gmail.com

Summary: This is the first complex project (everything is relative) you have to do.

Contents

I	instructions	2
II	Specific Rules of the day	4
III	Preamble	5
Part IV mandatory		6
IV.1	Introduction - FAQ.	6
IV.2 instructions6	
IV.2.1 Rulez.		6
IV.2.2 game data.		8
IV.2.3 Data Management.		8
IV.2.4 aesthetics of the game.		9
IV.2.5 pages.		10
V	party Bonus	13
VI	Rendering and peer-assessment	14
VII	Example Rendering	15

Chapter I

Instructions

- Only this page serve as a reference: Do not fi rm the hallway noise.
- The subject can change up to an hour before rendering.
- If no contrary information is explicitly present, you must assume the following language versions:
 - Ruby 2.3.0
 - for d09 Rails> 5
 - but for all the other days rails 4.2.6
 - HTML5
 - CSS 3
- We prohibit EXPRESSLY using keywords while, for, redo, break, retry and until in Ruby source code that you will make. Any use of these keywords is considered cheating (and / or unfit), giving you the score of -42.
- The exercises are precisely ordered from simple to more complex. In any case, we do not bear attention or take into account a complex exercise if a simpler exercise is not very successful.
- Attention to the rights of your fi les and your directories.
- You must follow the rendering process for all your exercise: only the present work on your GIT repository will be evaluated in defense.
- Your exercises will be evaluated by your pool mates.
- You should leave in your repertoire no other fi le than those explicitly speci fi ed by the forward drills.
- You have a question ? Ask your neighbor to the right. Otherwise, try your left neighbor.
- Your reference manual called Google / man / Internet /
- Think discuss on the forum of your pool Intra!
- Read the examples carefully. They might require things that

not otherwise specified in the subject ...

- Please, by Thor and Odin! Re fl échissez dammit!

chapter II

Specific Rules of the day

- You must make an application stateless
- You should not use database
- You should not use ActiveRecord
- This means:
 - Your class in / model (s) will inherit from nothing
 - Your / your controller (s) inherit (as by default) of ApplicationController- I
 - You are entitled (the app is not intended to be multi-user) to 4 global variables: \$ view, \$ selected, \$ \$ and game player
- The backups will be stored valid JSON.

Chapter III

Preamble

Intro scene from the movie [La Classe Américaine](#)

V12 - V12 called Captain George Abitbol, V12 called Captain George Abitbol. Someone asks you on the bridge.

George - Who ?

V12 - A man named Jose.

George - OK, I'm coming, V12.

José - Ah, that finally the king of the class!

José - The man too well-dressed, Abitbol!

José - So, you've been elected the most world class man! Let me laugh !

José - Style big Playboy Seabed kind that thrills household. Except me I fuck me, housewives, right? It's not true ?

George - Listen to me, my little Jose. You fuck housewives, well, you should have your ass shining. But that's not what we call the class. I'm telling you this as a more world class man.

José - Well, I'll stop now. The class is to be smart in the way he ha-biller.

José - Nothing better than going to Azzedine Alaia. or even to buy pullovers at Yohji Yamamoto!

George - Sorry to tell you this, my poor Jose, but you confuse a little while.

George - You make an amalgam between coquetry and class. You are crazy. You spend all your money on clothes and. . . Fashion Accessories. . . but you're ridiculous. Finally if you like it. . . It is you who the doors. But me, if you want my opinion, it's a little. . . has been.

José - The cow ! Me, I look has-been? I have for over a clothes bar on me. So fuck you put it!

George - You're really not very nice. But the train of your insults rolling on the rail of my indifference. I prefer to leave rather than to hear that rather than being deaf.

José - Well ! Considers that one is no longer friends, Abitbol!

chapter IV

mandatory part

IV.1 Introduction - FAQ

This rush to aim to make you encode a small single-player game with a web interface.

What is the purpose of this game?

This game is called Poke .. Heuu wait no, this game is called MovieMon and its goal is to capture all ...

-Movie-Mons- hiding on a game grid by using

-Movie-balls-.

What is Movieimon?

A Movieimon is a film available on IMDB. Ideally a film Monster.

How do you get a Movieimon?

By lowering its hit points to 0 before you. The IMDB rating of Movieimon

will be its strength. A high rating makes Movieimon more difficult to catch a low rating. The strength of the player, which is the number of Movieimons in his possession, increases the chances of catching.

How is a typical new part?

When a player starts a new game, the game application on IMDB all films required before sending the 'Worldmap', the main page of the game. On this page, the player moves freely and cheerfully from box to box on a grid of fixed size. Random boxes on which he walks, he uncovers a Movieimon. Two choices S'off then rent to him if it feels that its chances, the player tries to capture it. Otherwise, he fled cowardly.

If its energy falls below 0 before movieimon, this one escapes and leaves the game. If, instead, the movieimon is 0

before the player, then the movieimon is captured! The player can then consult its fi proudly Moviedex which lists all Movieimons

captured, before returning to hunt for all catch them! !

IV.2 instructions

IV.2.1 Rulez

Your game must obey some rules:

- the design of the front must be that of a gray laptop video game station has green screen (see screenshots).
- buttons should be clickable areas of the main screen layout [map image](#)-
- The size of the game grid must be a minimum of ten squares tall and ten squares wide.
- the game must start with a title screen
- the game is set on the grid screen (map) which should figure a map image and a player image (defining its position).
- on the game screen, the button 'start' via the "Movie_dex" and returns to the game screen via the 'start' button.
- if functionality via the buttons are present, a legend should indicate (eg "Press [A] to skip").
- click another button as defined by the legend does nothing (not even an error).
- the "movies_mons" must be loaded at the beginning of the game.
- the "movie_dex" is a list of "movie_mons" captured, the right and left buttons will to cycle through the pages of this circular fashion list: go left from the first item in the list should take you to the end of that list.



The API IMDB is rather obscure, you can use [OMDB](#) Which is an unofficial API, or even: [Random movie](#) .

IV.2.2 game data

You will need to retain data between different pages at a science game. A typical website would use cookies, or a system of server side sessions. But here there is no question of typical website.

You will need to store data in global variables. You have four at your disposal 'view', 'game', 'player' and menus 'selected'

So you must also create in your project, the logic needed to update and use of this file that should contain the following information:

- The player's position on the map.
- **The names (or identifiers) of all Moviemons of Moviedex.**
- **Full details of all Moviemons of the game, as obtained on IMDB.**

IV.2.3 Data Management

You must also create one (or more) class (es) whose mission is to manage these data set This (These) class (es) do (s) may contain at minimum the following methods.:

- 'Initialize'
- 'Save': Writes the data set to a valid JSON.
- 'Load': Plays the game data from a file and uses it to assign these data to the variables of the game.
- 'Get_movie': Returns an array or hash containing all the details of name **Moviemon Spent parameters necessary to page Detail.**

You can add to your class (es) all methods and attributes as necessary.

On the title screen, the button 'select' affiche célèbres the 3 'save slots' and load a backup. On the game screen (the map), the button 'select' affiche three 'save slots' and it can save the current game. On both screens to 'save slots', navigating with the fl up and down arrows.

IV.2.4 Aesthetics game

The aesthetic game will naturally run on your browser via the HTML and CSS. You do not have permission to use Javascript.

The aesthetic game is split into two parts to be visually distinguishable perfectly:

- **Screen:** Aesthetic what happens in the game. No interaction is possible at this point that should never contain any link or any form.
- **Controls :** Located below or on either side of the screen, they can interact with the game and are contextual, ie they change component according to the state of the game. This also means that they are not necessarily all new assets systematically. However, even inactive, they must be visible and keep the same place permanently.

There must be new 'buttons':

- Four directions, such as one might find on a directional pad Joystick: Up, Right, Down, Left
- A button select.
- A button start.
- A button Power (small red LED)
- A button AT
- A button B

You do not have permission to add / remove 'buttons' nor aesthetic information, apart from the name of 'buttons' in this area.

Beyond this minimum distinction, aesthetics does not matter in the mandatory part of the subject.

The behavior of buttons for each view is described in the next section.

IV.2.5 Pages

You must create pages / views / behaviors listed if after. A 'button' mentioned not a page is a 'button' inactive. Moreover, if the destination is not specified for a check is that it sent back the same page, potentially amended.

TitleScreen

- Description: Welcome Screen
- Display: Must a ffi expensive the name of the game and 'Start - New Game' and 'Select - Load'.
- controls:
 - **Start access Thu**
 - **Select access Save slots**
 - **Power " reboot "the game, reset global variables and returns to Title screen. worldmap**
- Description: card game, where the character moves, and flushes of Moviemons.
- Screen: A grid whose size is that defined in the settings. On the box corresponding to the current position of the player must find a representation (image, character, etc ...) clearly identifiable character.
- controls:
 - **directions:** Each department must move the character one square in the same direction. The player must not get out of the map. Each movement has a chance to flush a Moviemon
 - **start: access Moviedex.**
 - **select: access Save slots.**



Refresh this page does not change the position of the character on the card.

Battle

- **Screen:** A ffi che post the name Moviemon, the director's name, and your energy

If captured, you must also expensive ffi a sentence like "You caught it" to mark the event.

If a leak, you must also expensive ffi a sentence like "You coward!" .

- **controls:**

- **AT : hit moviemon**

Here the player launches a hit and subtracts its hit_point to the energy of moviemon, who also hit the turn with his hit_point to him who his' imdbRating. If successful (the energy of moviemon reaches 0 before you), the Moviemon is captured and stored in the MovieDex. You ffi Cherez has a right message, and return the energy of the player to the max, but also increase their hit points (and yes, experience it pays).

If unsuccessful, a ffi you Cherez an adequate message and you delete the mo- viemon the list available and you would redirect the player on the map.

- **B: flight to worldmap**

In this case the moviemon is released, player's energy is restored and back on the map.

Moviedex

- **Screen:** A Moviemon captured with information: Year, Genre, Director, imdbRating, Synopsis and Poster

- **controls:**

- **Right and left :** Directions to select a film diff erent. You must use at least two directions: left and right.

- **select:** Back to page worldmap



Default arriving on the page, the first film in the list is selected.

Save slot

- Description: Allows you to make backups of the game in the file 'save.json' or from the title screen to 'load' backup.
- Screen: A ffield three slots to save / load
- controls:
 - Up and down : Selects the slot to handle
 - Select: return to the screen precedent
 - the rest : AD LIB

Chapter V Part

Bonus

Once your required part perfectly realized, you can implement additional functionality in order to earn bonus points.

For your checker sees these features as successful, you will have to convince their good achievement.

An unhandled error invalidate the functionality in question.

Only under the bonus, the use of Javascript is tolerated as long as the latter does not affect the operation of the compulsory part (which must work without JavaScript). The AJAX or Websockets are not allowed.

Here are some bonus ideas you could implement:

- Match the controls to the keyboard in order to avoid having to use the mouse to play.
- In order to vary the game, made that each new part load a selection di ff E- monsters pension.
- Add variety to the worldmap with impassable items.

chapter VI

Rendering and peer-assessment

You must make a project tracks perfectly configured.

Apart from what is imposed on you in the subject, you are free to organize the project as you wish.

No server error will be tolerated. Test out the behavior of your views. You must provide a file requirement.txt containing all the necessary libraries to run your project.



Your models should not inherit anything, you may have migration generated by rails but nothing in the db

Veri fi cation with the following easy commnade executed in the console:

```
ActiveRecord :: Base.subclasses.map {| forbid | forbid.name}
```

chapter VII

Example Rendering



figure VII.1 - Your titlescreen Could look like that



figure VII.2 - This game Moviemon Appears to happen in A familiar map

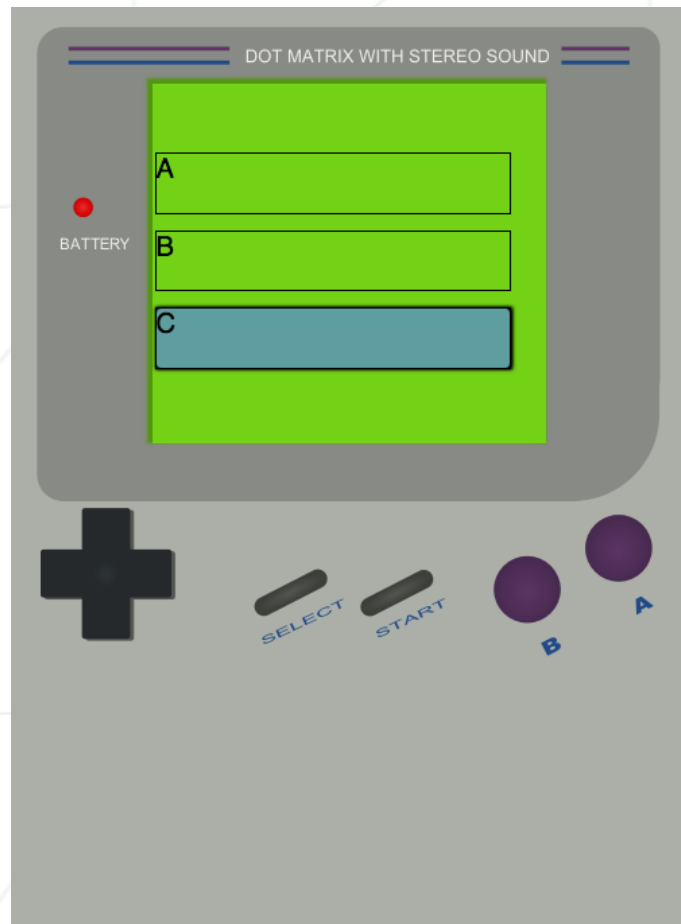


figure VII.3 - Save slot with 'c'selected

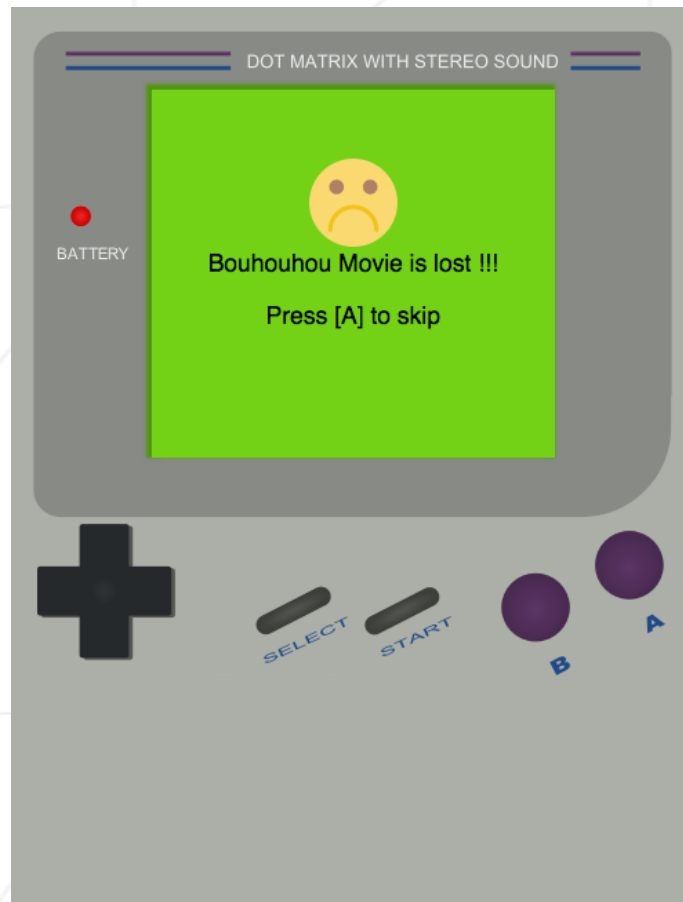


figure VII.4 - A battle lost messages

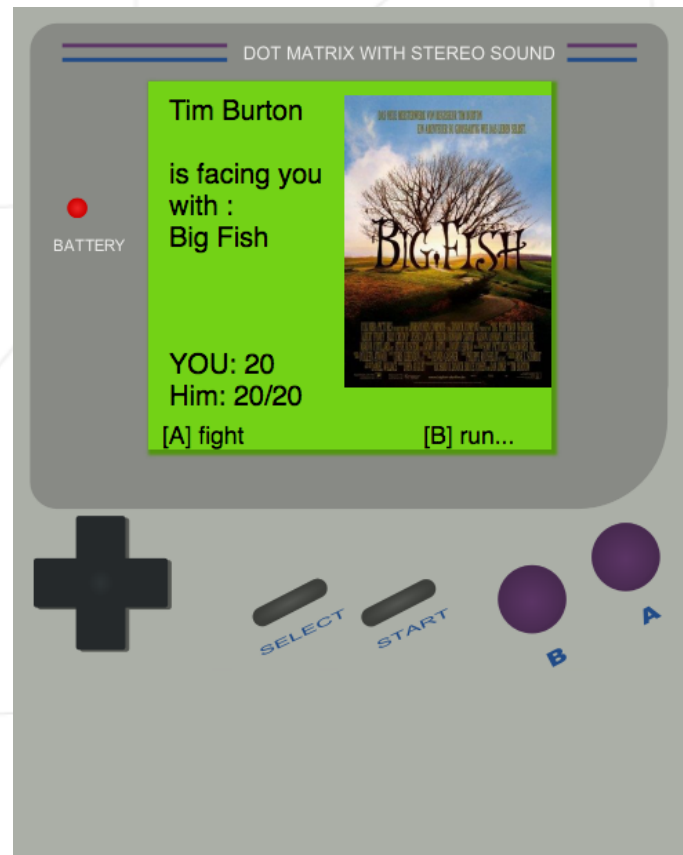


figure VII.5 - A fi ght screen



figure VII.6 - Your moviedex Could look like that