THE INTERNATIONAL INSTITUTE OF
SUPINFO
INFORMATION TECHNOLOGY

HTML, CSS & JavaScript

Academic Year – 2012 - 2013

Puzzler

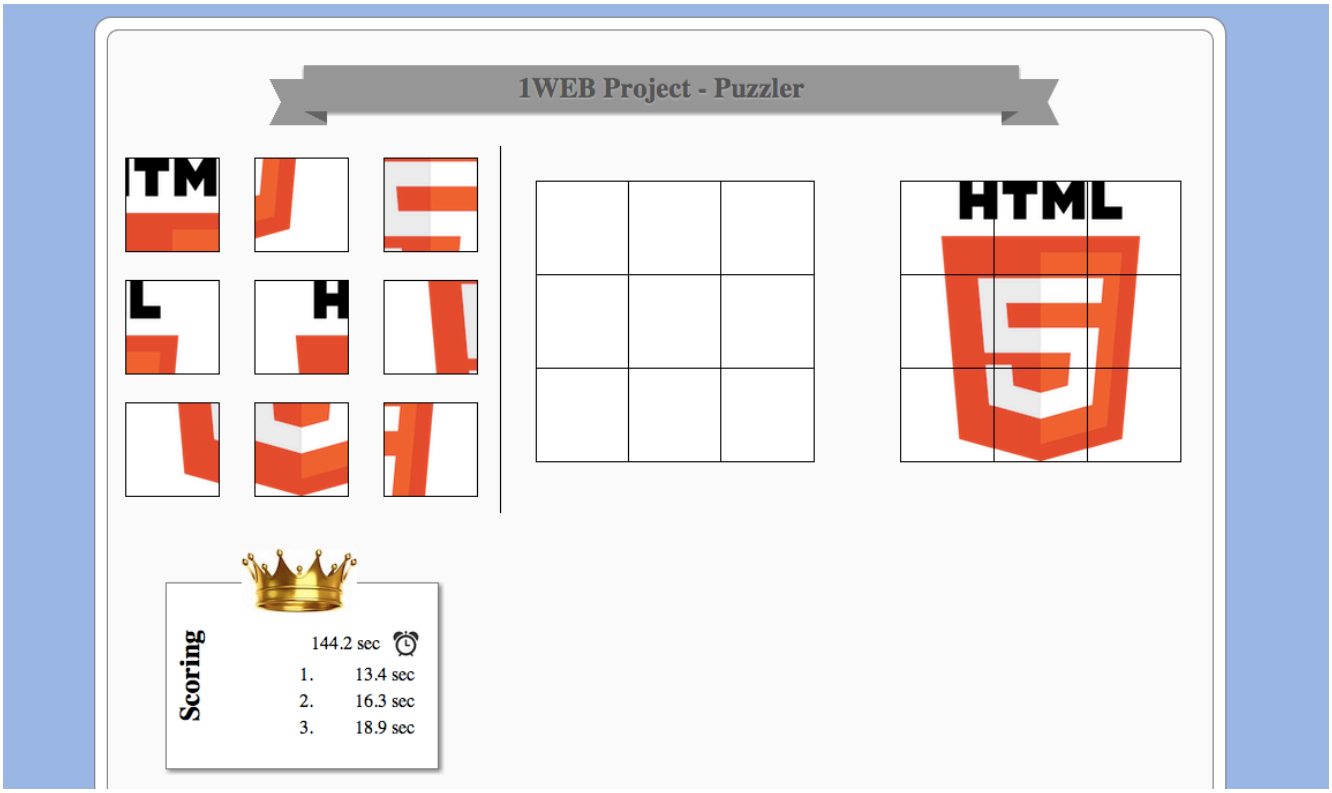
Graded Exercise

Context

An independent software development company called SupGaming, organize a contest. The purpose of this contest is to create a simple web game called “Puzzler”.

In two words, Puzzler shows a picture, and the user has to reproduce it. The game area consists of a 3x3 cells plate. Users have to drag and drop some pieces of picture inside it, in order to reproduce the full picture. The aim of the game leads in speed. When the pattern is reproduced, the time spent appears on the page, and another picture to reproduce is delivered to the user.

Here is an example of what SupGaming expects:



Specifications

The website will have the following functionalities:

- Represent the game area
- Make the pieces draggable and droppable
- Compare the user's input with the pattern
- Count the time spent on each pattern
- Display and sort scores

1. Game initialization

The game area is composed of:

- A « toolbox » zone with the picture pieces in disorder:
  - These must be draggable into the « game » table

- A « game » table with 3 rows and 3 columns:
  - Where you have to recreate the picture

- The complete « Picture » the user has to reproduce

- A « score » zone:
  - Used to display rankings in function of user time

The picture to reproduce must be chosen randomly in a set of three (at least).
To cut the full picture into pieces, **you must use only JS & CSS**.

2. Make the pieces draggable and droppable

As the user has to reproduce the pattern, he needs to drag and drop pieces from the "Toolbox" to the "game" table.

Be careful: pieces have to be dragged only into the "game" table, not elsewhere.

3. Compare the user's input with the pattern

Once all the pieces are placed in the "game" table, you have to compare what the user did with the complete picture. If there are the same, display a simple message like "You win" at the bottom of the page.

This message must disappear after one second.

4. Chronometer

At the right top corner, you have to display the time spent on the current picture. When the user finishes this pattern, take it and store it in the ranking area. Don't forget to set back the counter to 0 second after each game.

5. Display and sort scores

The "score" area is based on a ranking philosophy: best scores leads at top, while worst scores stays at bottom. You need to display only the three best scores, and sort them: the less time a user takes to finish its picture, the better he has to be placed.

6. Bonus

In order to be better placed in SupGaming contest, you can propose three difficulty levels.
Before beginning the game, a popup asks the user which difficulty level he wants:

- « Easy »: Displays a 3x3 cells grid
- « Medium »: Displays a 4x4 cells grid
- « Hard »: Displays a 5x5 cells grid

Notation

Functionalities	Points
Game Initialization	5
Draggable and droppable pieces	2
Win condition	4
Chronometer	2
Display and sort scores	3
Design	2
Code quality	2
Bonus – Difficulty levels	2
TOTAL	22

Appendix

You can use jQueryUI to develop easily some of these project functionalities: http://jqueryui.com/

For code quality, just remember this quote:

> *"Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live"* - ***Martin Golding***

Return

You will return your graded exercise inside a ZIP archive named: **Puzzler_Campus_IdBooster.zip**.
For example: Puzzler_Troyes_10000.zip
Not respecting this convention will be sanctioned by penalties points.
You will send the archive **to your STA SUPINFO email address only** and **before the 24th March at 11:59PM**.
After that delay, your project **will not be corrected and the mark 0 will be assigned to you**.