# Configuration

## How to start

#### Add necessary files

Include plugin’s stylesheet and script file in your project. You can do it either inside <head> tag or just before closing </body> tag.

(code)

#### HTML markup

Firstly you need to prepare your HTML markup. Nothing complicated here, the plugin only needs one empty div with an ID to hook on and it will do the rest.

(code)

\* If no div is specified, plugin is going to append the game to document’s body.

#### Initialize the script

You can run the plugin with default options, or customize it to your needs (available options are explained below). To run the plugin you need to add the following code in your script file or an inline script tag.

(code)

The most basic version, without any settings specified should look something like this:

(code)

If you added some custom settings it probably looks a bit more like this:

(code)

## Customization

#### How to specify settings

When you initialize the script you create a new object using a constructor, which accepts one argument – yes, you probably guessed it – this argument is our settings. You can store object with settings in separate variable and use it as an argument, or just write the object straight inside the constructor.

(code) or (code)

#### Available options

Below you can find list of settings you can customize as you like. It doesn’t need to come in exact order as they’re listed.

##### wrapper

Probably the one you’d like to change straight away. Do you remember that empty div you’ve created in ‘How to start’ section? That’s the time to use it. The game is going to be created inside this div.

(code) and (img)

Type: string

Default: document’s body

\*if you skip this, the game is going to be appended to document’s body.

##### images

This should be an array of strings with absolute paths to images you’d like to use in your game. The plugin will automatically create pairs of images and shuffle them, so you don’t need to insert same image twice. You can skip domain name, as that’s what browser assumes on its own. The size of the images might be different, it will be scaled down and matched, however the ratio must be the same! Otherwise the layout will break. So let’s say you use 400x200px image and 200x100px image – that’s fine. But using 400x200px image and 200x400px image is going to break the layout.

Type: array of strings

Default: [‘def1.jpg’, ‘def2.jpg’, ‘def3.jpg’]

(code)

##### backImage

Very similar to option above, but you need only one single string with absolute path to image you want to use as a back for your cards. This image should also have the same ratio as images specified in images setting.

Type: string

Default: ‘back.jpg’

(code)

##### animation

There are three different animation effects, when you flip the card, you can choose from: ‘flip’, ‘spin’ and ‘open’. Experiment, and pick one that suits you.

Type: string

Default: ‘flip’

(code) and (gifs)

##### columns

You can specify in how many columns you’d like the cards to be distributed.

Type: integer

Default: 2

(code) and (imgs)

cardShape

There is a choice of ‘square’ and ‘circle’ shapes.

Type: string

Default: ‘circle’

(code) and (imgs)

##### time

Amount of time you allow to solve the puzzle. The number represents minutes so 0.5 = 30s; 2.25 = 2min 15sec etc.

Type: integer

Default: 0.2

(code) and (img)

##### message

This setting is an object containing messages shown when puzzle is solved, failed or when user tries to reset the game.

Type: object

Default: { success: 'Well done', fail: 'Try again', reset: 'Progress will be lost'}

(code) and (imgs)

##### deductPoints

If it’s ‘true’, points will be deducted whenever card is flipped.

Type: boolean

Default: true

(code) and (gif)

##### overlayColour

Colour of overlay that appears with popup message. You can choose between ‘dark’ and ‘light’

Type: string

Default: ‘dark’

(code) and (imgs)

##### popupColour

Colour of popup modal.

Type: string

Default: ‘light’

(code) and(imgs)

##### popupAnimation

Choose the direction you’d like the modal to slide. Available options: ‘right’, ‘left’, ‘up’ and ‘down’

Type: string;

Default: ‘right’

(code) and (gifs)

##### responsive

Yup. It is responsive. You can specify number of columns at certain breakpoints. It is mobile first. Number of columns at breakpoint 0 – (your first defined breakpoint) will be the number you typed in columns setting or two (default value).

Type: array of objects

Default: []

(code) and (gif)

#### Colour scheme

Changing colour scheme is extremely easy; all you need to do is change variable values in SASS file and recompile it. By default $primary is orange and $secondary is yellow, feel free to change it to whatever you like to match your website layout.