



# DOSSIER PROFESSIONNEL (DP)

*Nom de naissance* ▶ GIBEAUD

*Nom d'usage* ▶ GIBEAUD

*Prénom* ▶ Emilien

*Adresse* ▶ 2 route de Saint Sigismond  
85490 BENET

## Titre professionnel visé

Développeur Web et Web Mobile

### MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.

**Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

### Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation.
4. de l'entretien final (dans le cadre de la session titre).

*[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]*

### Ce dossier comporte :

- pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- une déclaration sur l'honneur à compléter et à signer ;
- des documents illustrant la pratique professionnelle du candidat (facultatif)
- des annexes, si nécessaire.

*Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.*



<http://travail-emploi.gouv.fr/titres-professionnels>

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

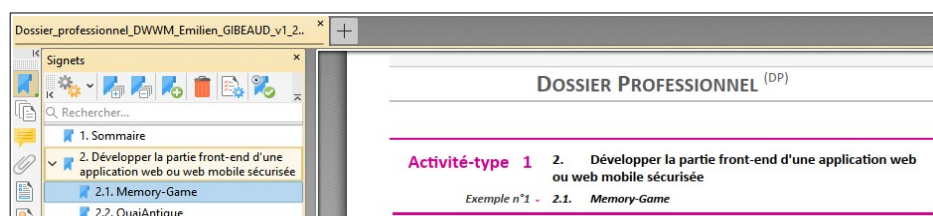
## Sommaire

### 1. Sommaire

Si vous visionnez ce document en **version pdf**, les numéros de **page ci-dessous** sont **cliquables** afin de naviguer dans le document.

Dans la suite du document vous rencontrerez d'**autres liens cliquables au format suivant** : Figure 1, 1.1 Titre 2, en page 1, ... (les liens cliquables sont systématiquement soulignés)

Aussi la version pdf possède une **table des matières** facilitant la **navigation dans le document** (Voir les réglages de votre lecteur PDF, exemple ci-dessous avec **PDF-Xchange**).



### Exemples de pratique professionnelle

<b>Développer la partie front-end d'une application web ou web mobile sécurisée</b>	<b>p.</b>	<b>5</b>
- Memory-Game	p.	5
- QuaiAntique	p.	10
- Au Bord de l'Eau	p.	13
<b>Développer la partie back-end d'une application web ou web mobile sécurisée</b>	<b>p.</b>	<b>15</b>
- Moviz	p.	15
- QuaiAntique	p.	18
- Au Bord de l'Eau	p.	20
<b>Titres, diplômes, CQP, attestations de formation (facultatif)</b>	<b>p.</b>	<b>22</b>
<b>Déclaration sur l'honneur</b>	<b>p.</b>	<b>23</b>
<b>Documents illustrant la pratique professionnelle (facultatif)</b>	<b>p.</b>	<b>24</b>
<b>Annexes</b>	<b>p.</b>	<b>25</b>

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

## Activité-type 1 2. Développer la partie front-end d'une application web ou web mobile sécurisée

### Exemple n°1 - 2.1. Memory-Game

#### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Ce projet est adapté d'un cours présent dans la médiathèque Studi ("Développer la partie front-end d'une application" → "Programmer avec Javascript" → "Projet : Memory game").

J'ai tout d'abord créé un **nouveau repository** sur mon compte **GitHub**, je l'ai mis en accès **public** et je l'ai nommé **Memory-Game**.

J'ai ensuite **cloné** le dépôt sur ma machine locale à l'aide de la commande :

```
git clone https://github.com/Emilien-GIBEAUD/Memory-Game.git
```

La procédure pour mettre en place le projet git est présentée en annexe [7.2 Projet Memory-Game](#) à la [Figure 1](#).

J'ai ensuite créé le **fichier HTML** qui est très simple :

Le **<body>** contient :

- une **<div>** contenant
  - les **<input>** de saisie de taille de grille (horizontale et verticale)
  - le **<button>** permettant de recharger la grille
- une **<div>** où seront injectées les cartes à l'aide de **JavaScript**

Le **fichier HTML** est présenté en annexe [7.2 Projet Memory-Game](#) à la [Figure 2](#).

Le **fichier CSS** est présenté en annexe [7.2 Projet Memory-Game](#) à la [Figure 3](#).

Le **fichier JS** est présenté en annexe [7.2 Projet Memory-Game](#) de la [Figure 4](#) à la [Figure 6](#).

- **Manipulation du DOM** (Document Object Model) :
  - Création dynamique des **éléments HTML** représentant les cartes (balises **<div>**, **<img>**, **<p>**) à l'aide de la **fonction create\_card(card\_address, english\_name, french\_name, card\_id)** définie en ligne 77.
  - Insertion et suppression dans la **<div id="game\_board">** à l'aide de la **fonction start()** définie en ligne 194.
  - Modification des **classes CSS** pour montrer ou masquer (ajout ou retrait de la **class="flip"**) et pour valider les cartes (ajout de la **class="matched"**) à l'aide de la **fonction card\_check(elem)** définie en ligne 157.

- Ajout des **propriétés CSS** du "**display: grid;**" permettant de définir la taille de la grille.
- **Gestion de la partie dynamique :**
  - Ajout d'un **écouteur d'événement** (au "click" sur le **<button>**) en ligne 60 pour recharger le jeu (à l'aide de **fonction start()**).
  - Ajout d'**écouteurs d'événements** (au "click" dans la **fonction create\_card(,,)** définis en ligne 97) pour interagir avec les cartes et déclencher les vérifications (à l'aide de la **fonction card\_check(elem)** pour rappel) et pour recharger le jeu (à l'aide de **fonction start()**) au "click" sur le **<button>** (cf. ligne 60).
  - Gestion de la logique d'ajout des cartes au chargement de la page (cf. ligne 56) ou au "click" sur le **<button>** à l'aide de la **fonction start()** définie en ligne 194.

Logique de la **fonction start()** :

- Mise à 0 du nombre de cartes découvertes
- Suppression des cartes (si présentes)
- Récupération des paramètres de l'utilisateur
- Ajout du style CSS définissant la taille de la grille
- Appel de la **fonction select\_n\_cards(array,n)** définie en ligne 139
- Appel de la **fonction duplicate\_array(array\_simple)** définie en ligne 111
- Appel de la **fonction shuffle\_array(array)** définie en ligne 128
- Mise à 0 du tableau de vérification de paire de cartes
- Ajout des cartes une à une dans la **<div id="game\_board">** à l'aide de la **fonction create\_card(,,)** définie en ligne 77

Logique de la **fonction select\_n\_cards(array,n)** :

- Déclaration de **array\_out**
- Passage de la **fonction shuffle\_array()** à **array**
- Écriture des **n** premières valeurs de **array** dans **array\_out** et renvoie de **array\_out**

Logique de la **fonction duplicate\_array(array\_simple)** :

- Copie profonde de **array\_simple** dans **array\_double**
- Copie profonde de **array\_simple** dans **array\_simple\_clone** et remplacement de la valeur "carte 1" par "carte 2" dans la quatrième valeur
- Ajout de **array\_simple\_clone** dans **array\_double** et renvoie de **array\_double**

Logique de la **fonction shuffle\_array(array)** :

- Déclaration de la constante **array\_shuffled** et enregistrement des valeurs de **array** après les avoir mélangées
- Renvoie de **array\_shuffled**

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

- Gestion de la logique de jeu pour vérifier une paire de carte à l'aide de la **fonction card\_check(elem)** définie en ligne 157 (déclenchée au "click" sur les cartes pour rappel).
  - Sortie de la fonction si le click est désactivé
  - Ajout de la **class="flip"** à la carte et ajout de la carte dans **cards\_to\_check**
  - La fonction est terminée sauf si **cards\_to\_check** contient deux cartes, dans ce cas :
    - **Désactivation du "click"** puis lancement d'un minuteur qui exécute les actions suivantes après la fin du délai indiqué :  
Suppression des **class="flip"** sur les cartes présentes dans **cards\_to\_check**, remise à 0 de **cards\_to\_check** et **réactivation du "click"**
    - La fonction est terminée sauf si les deux cartes présentes dans **cards\_to\_check** sont identiques, dans ce cas :
      - Arrêt immédiat du minuteur (les actions précédemment mentionnées ne sont donc pas réalisées)
      - Ajout de la **class="matched"** aux deux cartes et suppression des écouteurs
      - Incrémentation de **nb\_cards\_discovered**, remise à 0 de **cards\_to\_check** et **réactivation du "click"**
      - Si toutes les cartes sont découvertes, une alerte de réussite est envoyée

Le jeu est **déployé** sur une **GitHub Page** à l'aide d'une **GitHub Action** qui se déclenche lors de chaque commit poussé sur la branche "main" du repo GitHub.

On peut parler ici de **CD (Continuous Deployment)**.

Pour faire cela il faut aller sur la page **GitHub du projet** puis aller dans **Settings**→**Pages**→**GitHub Actions**→**Static HTML**→**Configure**.

Un fichier "static.yml" sera créé dans un dossier "./.github/workflow" et il faut ensuite "commiter" ce changement sur le dépôt distant.

On a maintenant notre dépôt distant qui est en "avance" sur notre dépôt local, pour mettre à jour ce dernier on va faire cette commande (sur le dépôt local) :

```
git pull
```

A chaque commit poussé sur la branche main, le déploiement sera mis à jour avec les dernières modifications :

```
git add .
git commit -m "Update ..."
git push
```

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

Cette procédure de mise en place du déploiement continu est présentée en annexe 7.2 Projet Memory-Game sur la Figure 7.

Le jeu en action est présenté en annexe 7.2 Projet Memory-Game de la Figure 8 à la Figure 10.

## 2. Précisez les moyens utilisés :

Pour ce projet j'ai utiliser l'IDE **Visual Studio Code**.

J'ai également utilisé **Git** et le projet est sur le repo **GitHub** suivant :

<https://github.com/Emilien-GIBEAUD/Memory-Game>

Frontend :

- **HTML5**
- **CSS3**
- **JavaScript**

Le projet est déployé sur une **GitHub Page** :

<https://emilien-gibeaud.github.io/Memory-Game/>

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.



# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cours Studi adapté

Chantier, atelier, service ▶ Formation

Période d'exercice ▶ Entre le 03/01 et le 10/01, le 06/06 et entre le 06/10 et le 10/10 (en 2025).  
(environ 10 jours de travail au total)

## 5. Informations complémentaires (facultatif)

Ce projet est adapté d'un cours présent dans la médiathèque Studi ("Développer la partie front-end d'une application" → "Programmer avec Javascript" → "Projet : Memory game").

Les améliorations suivantes ont été apportées :

- La carte à découvrir n'était qu'une simple image (le jeu présente un animal avec son nom en anglais et en français)
- Le clic est bloqué pendant la seconde attente (ce qui évite des bugs)
- En cas de découverte d'une paire, la seconde attente est supprimée
- Un bug qui permettait de voir l'image au "drag" est corrigé
- Il est possible de changer la taille de la grille
- Il est possible de relancer la partie

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Activité-type 1

Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°2 - 2.2. QuaiAntique

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

À venir

Projet en cours de réalisation, le repo GitHub est ici :

[https://github.com/Emilien-GIBAUD/Quai\\_Antique\\_Front](https://github.com/Emilien-GIBAUD/Quai_Antique_Front) (partie front uniquement)

Un aperçu rapide du projet est présenté en annexe 7.3 QuaiAntique (à venir) à la Figure 11.

Les travaux sur la partie back sont présentés au 3.2 QuaiAntique.

### 2. Précisez les moyens utilisés :

Pour ce projet j'ai utilisé l'IDE **Visual Studio Code**.

J'ai également utilisé **Git** et le projet est sur le repo **GitHub** suivant :

[https://github.com/Emilien-GIBAUD/Quai\\_Antique\\_Front](https://github.com/Emilien-GIBAUD/Quai_Antique_Front) (partie front uniquement)

Frontend :

- **HTML5**
- **CSS3** (généré à l'aide du langage SassScript et de sa syntaxe SCSS)
- **Bootstrap5**
- **JavaScript**

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

Tests :

- **Postman**

DevOps :

- **Docker**

Le projet est déployé sur :

[déploiement à venir](#)

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cours Studi

Chantier, atelier, service ▶ Formation

Période d'exercice ▶ Mi mars, fin mai et **décembre** (2025)  
(environ XX jours de travail au total)

## 5. Informations complémentaires (facultatif)

Ce projet a été développé à l'aide de cours présents dans la médiathèque Studi :

- "Développer la partie front-end d'une application" → "Projet fil rouge" (7 cours d'environ 2h)
- "Développer la partie back-end d'une application" → "Projet fil rouge" (9 cours d'environ 2h)

Il s'agit d'un ancien projet ECF proposé par Studi et qui couvre l'ensemble des compétences de l'ancien référentiel du Titre Professionnel (TP) Développeur Web et Web Mobile (DWWM).

Les cours couvrent l'entièreté de la structure du projet et des compétences de l'ancien référentiel du TP DWWM mais pas l'entièreté du développement des fonctionnalités.

Aussi, j'ai fait certain choix différents de ceux des cours :

- J'ai utilisé la **version 8.3 de PHP** (plutôt que la version 7.2)
- J'ai utilisé la **version 7.2 de Symfony** (plutôt que la version 5.4)
- J'ai commencé par utiliser (comme dans le cours) une **stack AMP** (Apache, MariaDB, PHP autrement dit **serveur Web Apache, base de données MariaDB, langage de programmation PHP**) portée par **WAMP** pour rapidement **basculer vers une stack Docker**.
- J'ai déployé sur un VPS Digital Ocean (grâce aux 200 \$ offerts dans le GitHub Student Developer Pack) plutôt que sur alwaysdata.com pour le front et sur platform.sh pour le back.
- J'ai adapté le router JavaScript pour factoriser le code (à voir si besoin aussi complexe que pour EcoRide)

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Activité-type 1

Développer la partie front-end d'une application web ou web mobile sécurisée

Exemple n°3 - 2.3. Au Bord de l'Eau

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

À venir

Projet en cours de réalisation, le repo GitHub est privé et la GitHub Page de la branch "main" est ici : <https://emilien-gibeaud.github.io/auborddeleau/>

Un aperçu rapide du projet est présenté en annexe 7.4 Au Bord de l'Eau (en cours) à la Figure 12.

Les travaux sur la partie back (concernant le déploiement uniquement) sont présentés au 3.3 Au Bord de l'Eau.

### 2. Précisez les moyens utilisés :

Pour ce projet j'ai utilisé l'IDE **Visual Studio Code** (avec l'extension "WSL").

J'ai également utilisé **Git** et le projet est sur le repo **GitHub** suivant : <https://github.com/Emilien-GIBAUD/auborddeleau> (ce repo est privé)

Frontend :

- **HTML5**
- **CSS3** (généré à l'aide du langage SassScript et de sa syntaxe SCSS)
- **JavaScript**

Le projet a été développé sur une distribution **Ubuntu** (via **WSL2**) et est servi par un **conteneur Docker** utilisant un **serveur Nginx**.

Le projet est déployé sur une **GitHub Page** et **utilisera** le nom de domaine suivant : <https://www.restaurantauborddeleau.com/> (ancienne version, nouvelle version en construction)

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé avec ma belle sœur (propriétaire du restaurant avec mon frère) pour le cadrage du projet et le maquettage.

Le maquettage a principalement été réalisé par ma belle sœur sous Figma mais avec mon aide pour la gestion de l'accessibilité, du responsive et de manière générale lui indiquer ce qu'il est possible de coder. Le code et le déploiement du site ont été réalisés seul.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Restaurant Au Bord de l'Eau

Chantier, atelier, service ▶ /

Période d'exercice ▶ Octobre à décembre 2025  
(environ XX jours de travail au total)

## 5. Informations complémentaires (facultatif)

L'ancienne version du site avait été réalisée en 2017, le site utilisait une base **Wordpress**, était hébergé sur un **serveur mutualisé OVH** (offre Pro à 6,59 € HT/mois) et utilisait un nom de domaine via OVH (à 13,49 € HT/mois).

Le serveur mutualisé a donc été remplacé par la **GitHub Page** et le **nom de domaine conservé**.

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Activité-type 2 3. Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 1 - 3.1. Moviz

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

À venir

Projet en cours de réalisation, le repo GitHub est ici :

<https://github.com/Emilien-GIBEAUD/Moviz>

Un aperçu rapide du projet est présenté en annexe 7.5 Moviz (à venir) à la Figure 14.

### 2. Précisez les moyens utilisés :

Pour ce projet j'ai utilisé l'IDE **Visual Studio Code**.

J'ai également utilisé **Git** et le projet est sur le repo **GitHub** suivant :

<https://github.com/Emilien-GIBEAUD/Moviz>

Frontend :

- **HTML5**
- **Bootstrap5**

Backend :

- **PHP 8.3**

Base de données :

- **MariaDB**

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

DevOps :

- **Docker**

Le projet est déployé sur :

[déploiement à venir](#)

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Travaux dirigés Studi (5 lives d'une heure)

Chantier, atelier, service ▶ Formation

Période d'exercice ▶ Fin avril à début mai et **décembre** (2025)

*(environ XX jours de travail au total)*



## 5. Informations complémentaires *(facultatif)*

Ce projet est adapté d'une série de 5 lives d'une heure intitulés "PHP : Développement d'un site de critique de film en PHP OBJET et MVC".

Le contenu des lives :

- Live 1 : Présentation du projet, conception de la base de données et mise en place Wamp
- Live 2 : Explications POO et MVC, mise en place de la structure MVC
- Live 3 : Création de la table User et mise en place de la création/connexion d'un compte utilisateur
- Live 4 : Création des autres tables et mise en place de la sélection d'un film (model et controller)
- Live 5 : Vue d'un film et affichage de tous les films (model vue et controller)

Les fonctionnalités suivantes ont été mises en place par moi même :

- Ajout d'un film (rôle admin)
- Ajout d'un commentaire (rôle user) et validation (rôle admin)
- **Ajout d'une fonctionnalité de recherche (avec filtres) en cours**

## Activité-type 2

Développer la partie back-end d'une application web ou web mobile sécurisée

Exemple n° 2 - 3.2. QuaiAntique

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

À venir

Projet en cours de réalisation, le repo GitHub est ici :

[https://github.com/Emilien-GIBEAUD/quai\\_antique](https://github.com/Emilien-GIBEAUD/quai_antique) (partie back uniquement)

Un aperçu rapide du projet est présenté en annexe 7.3 QuaiAntique (à venir) à la Figure 11.

Pour rappel, les travaux sur la partie front sont présentés au 2.2 QuaiAntique.

### 2. Précisez les moyens utilisés :

Pour ce projet j'ai utilisé l'IDE **Visual Studio Code**.

J'ai également utilisé **Git** et le projet est sur le repo **GitHub** suivant :

[https://github.com/Emilien-GIBEAUD/quai\\_antique](https://github.com/Emilien-GIBEAUD/quai_antique) (partie back uniquement)

Backend :

- **PHP 8.3**
- **Symfony 7.2 (API REST)**

Base de données :

- **MariaDB**

Tests :

- **Postman**

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

DevOps :

- **Docker**

Le projet est déployé sur :

[déploiement à venir](#)

## 3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet.

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Cours Studi

Chantier, atelier, service ▶ Formation

Période d'exercice ▶ Mi mai à mi juin et **décembre** (2025)  
(environ XX jours de travail au total)

## 5. Informations complémentaires (facultatif)

Voir le point 5 du [2.2 QuaiAntique](#).

## Activité-type 2

### Développer la partie back-end d'une application web ou web mobile sécurisée

#### Exemple n° 3 - 3.3. Au Bord de l'Eau

#### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour rappel, les travaux sur la partie front sont présentés au [2.3 Au Bord de l'Eau](#) en page [13](#).

Le projet est déployé sur une **GitHub Page** et utilise donc la même procédure de déploiement que celle utilisée par le projet "Memory-Game" (voir explications au [2.1 Memory-Game](#) en page [7](#) ainsi que la [Figure 7](#) à l'annexe [7.2 Projet Memory-Game](#)).

Il a ensuite fallu raccorder le **nom de domaine** à la **GitHub Page**, je me suis aidé pour cela de la documentation GitHub concernant les GitHub Page.

La documentation et la procédure suivie sont présentées en annexe [7.4 Au Bord de l'Eau \(en cours\)](#) à la [Figure 13](#).

#### 2. Précisez les moyens utilisés :

Voir le point 2 du [2.3 Au Bord de l'Eau](#).

#### 3. Avec qui avez-vous travaillé ?

Voir le point 3 du [2.3 Au Bord de l'Eau](#).

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## 4. Contexte

Nom de l'entreprise, organisme ou association ▶ Voir le point 4 du 2.3 Au Bord de l'Eau.

Chantier, atelier, service ▶ Voir le point 4 du 2.3 Au Bord de l'Eau.

Période d'exercice ▶ Voir le point 4 du 2.3 Au Bord de l'Eau.

## 5. Informations complémentaires *(facultatif)*

Voir le point 5 du 2.3 Au Bord de l'Eau.

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Titres, diplômes, CQP, attestations de formation

*(facultatif)*

### 4. Titres, diplômes, CQP, attestations de formation

Intitulé	Autorité ou organisme	Date
Master en Conception Mécanique	Université de Poitiers	2009
DUT Génie Mécanique et Productique	Université d'Angers	2006

## Déclaration sur l'honneur

### 5. Déclaration sur l'honneur

Je soussigné(e) Emilien GIBEAUD,

déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à Benet

le 01/12/2025

pour faire valoir ce que de droit.

Signature :



# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

## Documents illustrant la pratique professionnelle

*(facultatif)*

### 6. Documents illustrant la pratique professionnelle

Intitulé



## ANNEXES

### 7. Annexes

#### Sommaire des annexes

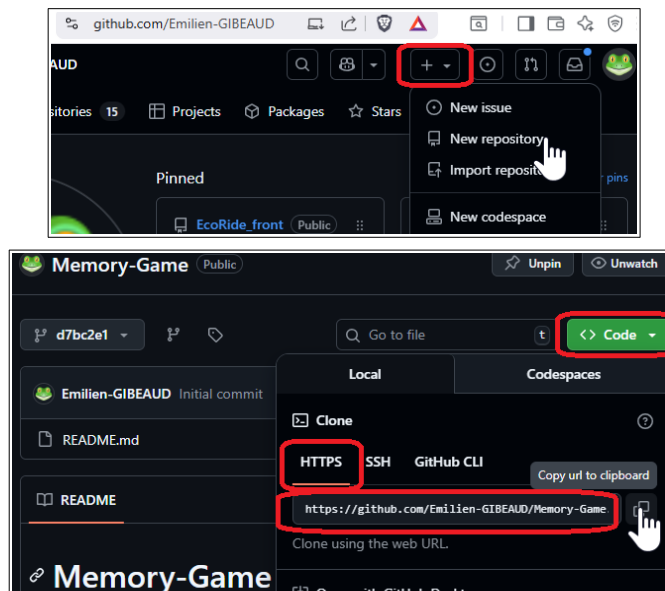
7. Annexes.....	25
7.1. Index des figures.....	25
7.2. Projet Memory-Game.....	26
7.3. QuaiAntique (à venir).....	36
7.4. Au Bord de l'Eau (en cours).....	37
7.5. Moviz (à venir).....	39

#### 7.1. Index des figures

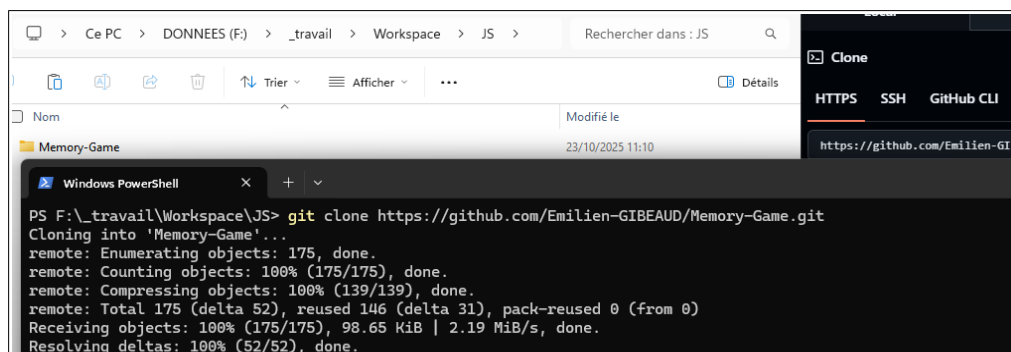
#### Index des figures

Figure 1: Memory-Game – Mise en place du projet Git/GitHub.....	26
Figure 2: Memory-Game - Fichier HTML.....	27
Figure 3: Memory-Game - Fichier CSS.....	28
Figure 4: Memory-Game - Fichier JS 1/3.....	29
Figure 5: Memory-Game - Fichier JS 2/3.....	30
Figure 6: Memory-Game - Fichier JS 3/3.....	31
Figure 7: Memory-Game – GitHub Action.....	32
Figure 8: Memory-Game – Exemple jeu 1/3.....	33
Figure 9: Memory-Game – Exemple jeu 2/3.....	34
Figure 10: Memory-Game – Exemple jeu 3/3.....	35
Figure 11: QuaiAntique – Plusieurs figures à venir.....	36
Figure 12: Au Bord de l'Eau – Plusieurs figures à venir.....	37
Figure 13: Au Bord de l'Eau – Nom de domaine et gestion DNS.....	38
Figure 14: Moviz – Plusieurs figures à venir.....	39

## 7.2. Projet Memory-Game

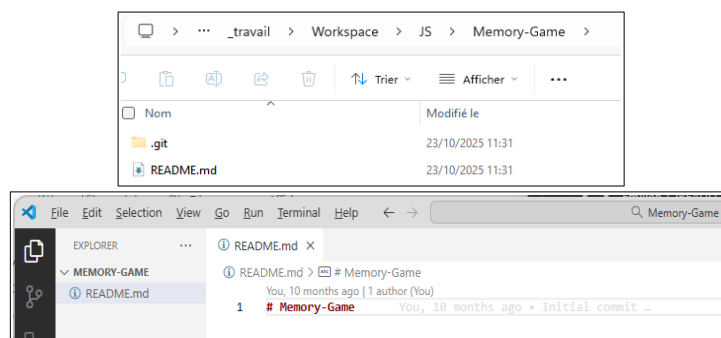


Création d'un nouveau repo GitHub et copie de l'adresse.



*(il ne s'agit pas de la vraie adresse du dépôt local)*

Clonage du projet dans le dossier souhaité.



Ouverture du projet dans Visual Studio Code.

**Figure 1:** Memory-Game – Mise en place du projet Git/GitHub

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta name="author" content="Emilien GIBEAUD">
7   <meta name="description" content="Jeu permettant de faire travailler sa mémoire.">
8   <link rel="shortcut icon" href="/Assets/icon/brain.svg" type="image/x-icon">
9   <link rel="stylesheet" href="/Assets/style.css">
10  <script src="/Assets/script.js" type="module"></script>
11  <title>Memory Game</title>
12 </head>
13 <body>
14   <div class="settings">
15     <div class="input">
16       <label for="horizontal">Cartes → (3≤n≤8)&nbsp;  </label>
17       <input type="number" id="horizontal" value="3" min="3" max="8" step="1">
18     </div>
19     <div class="input">
20       <label for="vertical">Cartes ↓ (3≤n≤6)&nbsp;  </label>
21       <input type="number" id="vertical" value="4" min="3" max="6" step="1">
22     </div>
23     <input type="button" id="start" class="btn" value="Recharger">
24   </div>
25   <div id="game_board">
26     <!-- Exemple de carte
27     <div class="card" data-value="/Assets/icon/grasshopper.svg" data-id="carte 2">
28       
29       <p class="name_en">grasshopper</p>
30       <p class="name_fr">(sauterelle)</p>
31     </div>
32     -->
33   </div>
34 </body>
35 </html>
```

**Figure 2: Memory-Game - Fichier HTML**

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

```
1  * { /* RESET */
2    margin: 0;
3    padding: 0;
4    box-sizing: border-box;
5    text-decoration: none;
6    list-style: none;
7  }
8  body {
9    text-align: center;
10   font-family: Arial, sans-serif;
11 }
12 .settings {
13   padding: 5px;
14   display: flex;
15   flex-wrap: wrap;
16   justify-content: space-evenly;
17   align-items: center;
18   gap: 10px;
19 }
20 .input {
21   display: flex;
22   align-items: center;
23   gap: 5px;
24 }
25 @media (max-width: 480px) {
26   label {
27     width: 70px;
28   }
29 }
30 input[type="number"] {
31   text-align: center;
32   width: 45px;
33   padding: 3px;
34 }
35 .btn {
36   width: auto;
37   height: auto;
38   padding: 5px 8px;
39
40   padding: 5px 8px;
41   border-width: thin;
42   border-radius: 10px;
43 }
44 #game_board {
45   display: grid;
46   /* grid-template-columns: repeat(3, 100px);
47   grid-template-rows: repeat(6, 100px);
48   Mis par JS en fonction de la valeurs des inputs*/
49   gap: 5px;
50   justify-content: center;
51   margin: 20px auto;
52 }
53 .card {
54   display: flex;
55   flex-direction: column;
56   align-items: center;
57   background-color: whitesmoke;
58   border-radius: 20px;
59   cursor: pointer; /* affiche le doigt au survol de l'élément */
60 }
61 .card_content {
62   height: 65px;
63   padding: 2px 15px;
64 }
65 .card.flip .card_content, .card.flip .name_en, .card.flip .name_fr {
66   opacity: 1;
67 }
68 .card .card_content, .card .name_en, .card .name_fr {
69   opacity: 0;
70 }
71 .card img { /* Pour éviter que l'image s'affiche lors d'un drag */
72   -webkit-user-drag: none;
73 }
74 .card.matched {
75   border: solid medium green;
76   cursor: default;
77 }
78
79 .name_fr, .name_en {
80   width: 100%;
81 }
82 .name_fr {
83   font-style: italic;
84   font-size: smaller;
85 }
```

**Figure 3: Memory-Game - Fichier CSS**

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

```
You, 2 minutes ago | 1 author (You)
1 // --- MISE EN PLACE DU JEU -----
2 const game_board = document.getElementById("game_board");
3 const horizontal = document.getElementById("horizontal");
4 const vertical = document.getElementById("vertical");
5
6 // Déclaration des paramètres de jeu
7 const timeout = 1000; // durée d'affichage des cartes non identiques (ms)
8 const nb_x_min = 3; // nombre de cartes minimum à l'horizontale
9 const nb_x_max = 8; // nombre de cartes maximum à l'horizontale
10 const nb_y_min = 3; // nombre de cartes minimum à la verticale
11 const nb_y_max = 6; // nombre de cartes maximum à la verticale
12 let nb_x; // nombre de cartes à l'horizontale (input utilisateur)
13 let nb_y; // nombre de cartes à la verticale (input utilisateur)
14 let nb_cards; // nombre de cartes à trouver (calculé avec nb_x et nb_y)
15 let nb_cards_discovered; // nombre de cartes trouvées
16 let selected_cards; // cartes sélectionnées parmi les cartes disponibles
17 let all_cards; // cartes sélectionnées avec les doublons
18 let cards_to_check; // paire de cartes en sélection pour vérification
19 let click_disabled = false; // Le click est activé au démarrage
20
21 // déclaration des cartes disponibles
22 const available_cards = [
23   ["/Assets/icon/ant.svg", "ant", "fourmi", "carte 1"],
24   ["/Assets/icon/bat.svg", "bat", "chauve-souris", "carte 1"],
25
26   ["/Assets/icon/spider.svg", "spider", "araignée", "carte 1"],
27   ["/Assets/icon/whale.svg", "whale", "baleine", "carte 1"]
28 ];
29
30 // Lancement du jeu au chargement de la page (ou au rafraîchissement)
31 start();
32
33 // Bouton pour recharger le jeu
34 const start_button = document.getElementById("start");
35 start_button.addEventListener("click", start);
36
37 // --- DÉBUT DES DECLARATIONS DE FONCTIONS -----
38 /* Carte d'exemple : (cf. index.html) ...
39
40 /**
41 * Crée une carte avec une image, un nom en anglais, un nom en français et le numéro de carte ("carte 1").
42 * @param {string} card_address - L'URL de l'image de la carte.
43 * @param {string} english_name - Le nom de l'animal en anglais.
44 * @param {string} french_name - Le nom de l'animal en français.
45 * @param {string} card_id - "carte 1".
46 */
47 function create_card(card_address, english_name, french_name, card_id) {
48   const card = document.createElement("div"); // crée un <div>-----
49   card.classList.add("card"); // avec la class="card"
50   card.dataset.value = card_address; // avec la data-value="card_address"
51   card.dataset.id = card_id; // avec la data-id="card_id"
52   const card_content = document.createElement("img"); // crée un <img>-----
53   card_content.classList.add("card_content"); // avec la class="card_content"
54   card_content.src = `${card_address}`; // avec la src="card_address"
55   card.appendChild(card_content); // ajoute l'enfant "card_content" à "card"
56   const name_en = document.createElement("p"); // crée un <p>-----
57   name_en.classList.add("name_en"); // avec la class="name_en"
58   const txt_en = document.createTextNode(english_name); // nom anglais présent dans la balise <p>
59   name_en.appendChild(txt_en); // ajoute l'enfant "txt_en" à "name_en"
60   card.appendChild(name_en); // ajoute l'enfant "name_en" à "card"
61   const name_fr = document.createElement("p"); // crée un <p>-----
62   name_fr.classList.add("name_fr"); // avec la class="name_fr"
63   const txt_fr = document.createTextNode(`${french_name}`); // nom français
64   name_fr.appendChild(txt_fr); // ajoute l'enfant "txt_fr" à "name_fr"
65   card.appendChild(name_fr); // ajoute l'enfant "name_fr" à "card"
66   card.addEventListener("click", card_check); // ajoute un écouteur d'événement au click
67   return card;
68 }
```

**Figure 4: Memory-Game - Fichier JS 1/3**

```
99  };
100 > /* //Code pour tester la fonction...
105
106 /**
107  * Duplique un tableau.
108  * Le tableau se retrouve avec 2 fois chaque élément ("carte 1" et "carte 2").
109  * @param {array} array_simple - L'URL de l'image de la carte.
110  */
111 function duplicate_array(array_simple) {
112     let array_double = [];
113     array_double.push(...JSON.parse(JSON.stringify(array_simple))); // Copie profonde
114     // ... : ajoute 1 à 1 les cases du tableau, ajoute le tableau complet dans la 1ère case sinon
115
116     let array_simple_clone = JSON.parse(JSON.stringify(array_simple)); // Copie profonde
117     for (let i = 0; i < array_simple_clone.length; i++){
118         array_simple_clone[i][3] = "carte 2"; // pour différencier les 2 cartes
119     }
120     array_double.push(...array_simple_clone); // ajoute le clone avec les "carte 2"
121     return array_double;
122 };
123
124 /**
125  * Mélange un tableau.
126  * @param {array} array - Le tableau à mélanger.
127  */
128 function shuffle_array (array) {
129     const array_shuffled = array.sort(() => 0.5 - Math.random());
130     return array_shuffled
131 };
132
133 /**
134  * Retourne un tableau avec les n premières valeurs d'un autre tableau.
135  * @param {array} array - Le tableau d'où extraire les valeurs.
136
137  * @param {integer} n - Le nombre de valeurs à extraire.
138  * @returns {array} Le tableau avec les n premières valeurs de array.
139  */
140 function select_n_cards (array,n) {
141     let array_out = [];
142     array = shuffle_array(array);
143     for(let i = 0; i < n; i++) {
144         array_out.push(array[i]);
145     }
146     return array_out;
147 }
148
149 /**
150  * Vérifie une paire de carte.
151  * Si une carte est cliquée, elle est rendue visible.
152  * Si une deuxième carte est cliquée, elle est également rendue visible.
153  * Si elles sont identiques, elles sont encadrées en vert immédiatement et conservées visibles.
154  * Si elles sont différentes, une tempo de 2s s'enclenche pour mémorisation puis les cartes sont masquées.
155  * Si toutes les paires sont trouvées, une alerte de félicitations s'affiche.
156  * @param {HTMLElement} elem - L'élément HTML à rendre visible.
157  */
158 function card_check(elem){
159     if(click_disabled){
160         return;
161     }
162     const card = elem.target.parentElement; // cible le parent
163     card.classList.add("flip"); // ajoute la class="flip" au parent
164     cards_to_check.push(card); // ajoute une carte à vérifier
165
166     if(cards_to_check.length === 2) {
167         click_disabled = true; // click désactivé si 2 cartes sont en sélection
168         let timer = window.setTimeout(() => { // démarre un timer de 2 secondes
169             cards_to_check[0].classList.remove("flip"); // supprime les class="flip"
170             cards_to_check[1].classList.remove("flip");
```

**Figure 5: Memory-Game - Fichier JS 2/3**

# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

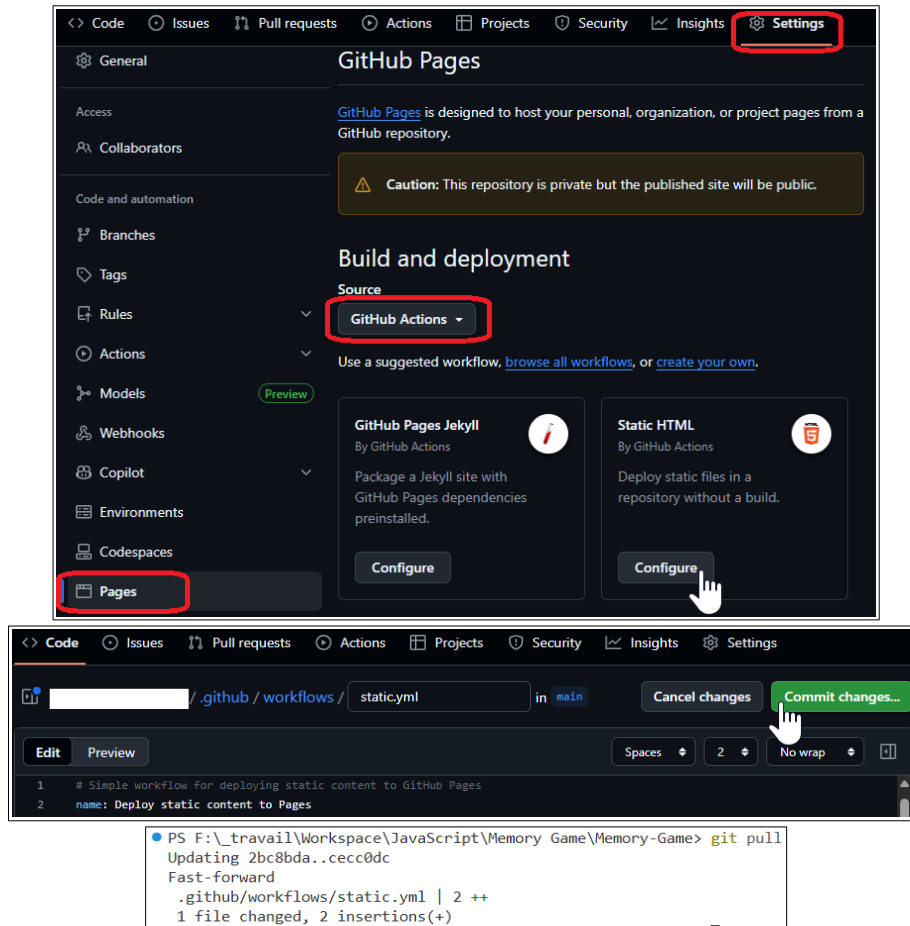
```
170 cards_to_check = []; // suppression des cartes à vérifier
171 click_disabled = false; // click réactivé
172 }, timeout);
173 if( (cards_to_check[0].dataset.value === cards_to_check[1].dataset.value) &&
174 (cards_to_check[0].dataset.id !== cards_to_check[1].dataset.id) ) { // paire trouvée

175 window.clearTimeout(timer);
176 // ajoute les class="matched" et supprime les écouteurs d'événement
177 cards_to_check[0].classList.add("matched");
178 cards_to_check[1].classList.add("matched");
179 cards_to_check[0].removeEventListener("click", card_check);
180 cards_to_check[1].removeEventListener("click", card_check);
181 nb_cards_discovered++;
182 cards_to_check = []; // suppression des cartes à vérifier
183 click_disabled = false; // click réactivé
184 if(nb_cards_discovered === nb_cards) {
185 window.alert("Bravo, vous avez gagné !!!");
186 }
187 }
188 }
189 };
190
191 /**
192 * Relance le jeu (en utilisant les paramètres saisis).
193 */
194 function start() {
195 nb_cards_discovered = 0;
196 // Suppression des cartes
197 const div_card = document.querySelectorAll(".card");
198 div_card.forEach(div => {
199 div.remove();
200 });
201 // Récupération et validation des paramètres
202 nb_x = parseInt(escapeHTML(horizontal.value));
203 nb_x = nb_x < nb_x_min ? (nb_x_min , horizontal.value = nb_x_min) : nb_x;
204 nb_x = nb_x > nb_x_max ? (nb_x_max , horizontal.value = nb_x_max) : nb_x;
205 nb_y = parseInt(escapeHTML(vertical.value));
206 nb_y = nb_y < nb_y_min ? (nb_y_min , vertical.value = nb_y_min) : nb_y;
207 nb_y = nb_y > nb_y_max ? (nb_y_max , vertical.value = nb_y_max) : nb_y;
208 nb_cards = ((nb_x * nb_y) / 2) - ((nb_x * nb_y) % 2);
209 // Mise en place de la grille
210 game_board.style.gridTemplateColumns = `repeat(${nb_x}, 100px)`;
211 game_board.style.gridTemplateRows = `repeat(${nb_y}, 100px)`;
212 selected_cards = select_n_cards(available_cards,nb_cards);

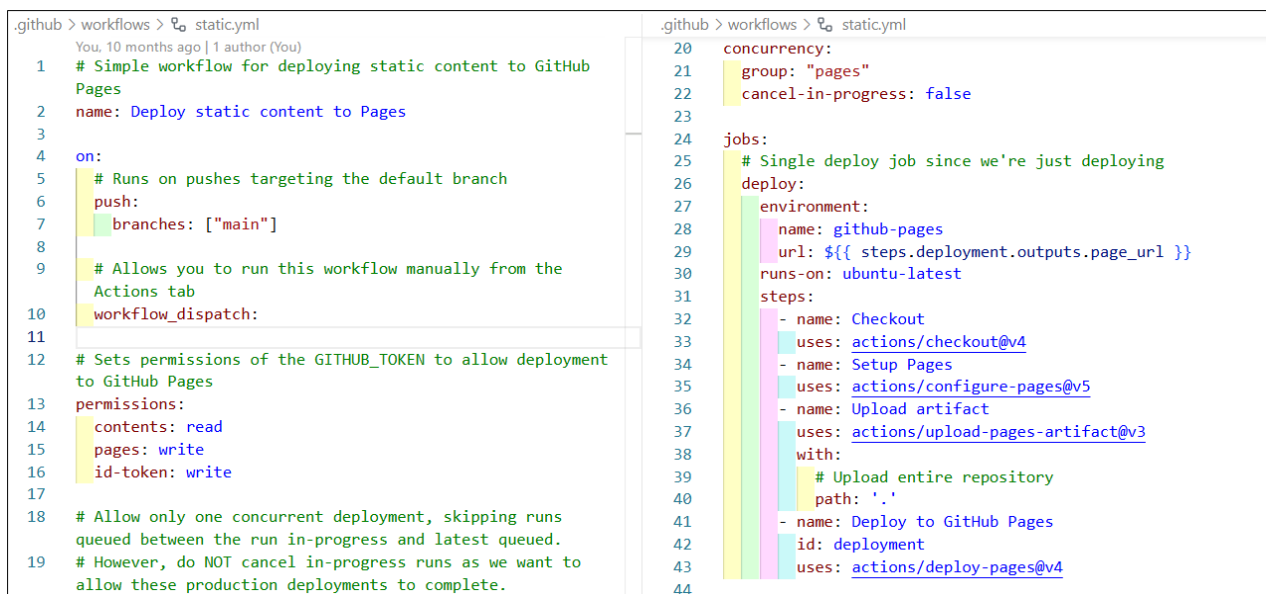
213 all_cards = duplicate_array(selected_cards);
214 all_cards = shuffle_array(all_cards);
215 cards_to_check = [];
216 // Affichage des cartes
217 all_cards.forEach(card_to_create => {
218 const card_html = create_card(card_to_create[0],card_to_create[1],card_to_create[2],card_to_create[3]);
219 game_board.appendChild(card_html);
220 });
221 }
222
223 /**
224 * Échappe le HTML d'un input pour éviter les attaques XSS.
225 * Convertit les caractères spéciaux (<, >, &, etc.) en entités HTML.
226 * @param {string|number} input - L'input à échapper.
227 * @returns {string} L'input avec les caractères HTML échappés.
228 */
229 function escapeHTML(text){
230 const tempHtml = document.createElement("div");
231 tempHtml.textContent = text;
232 return tempHtml.innerHTML;
233 }
234
```

**Figure 6: Memory-Game - Fichier JS 3/3**

# DOSSIER PROFESSIONNEL (DP)



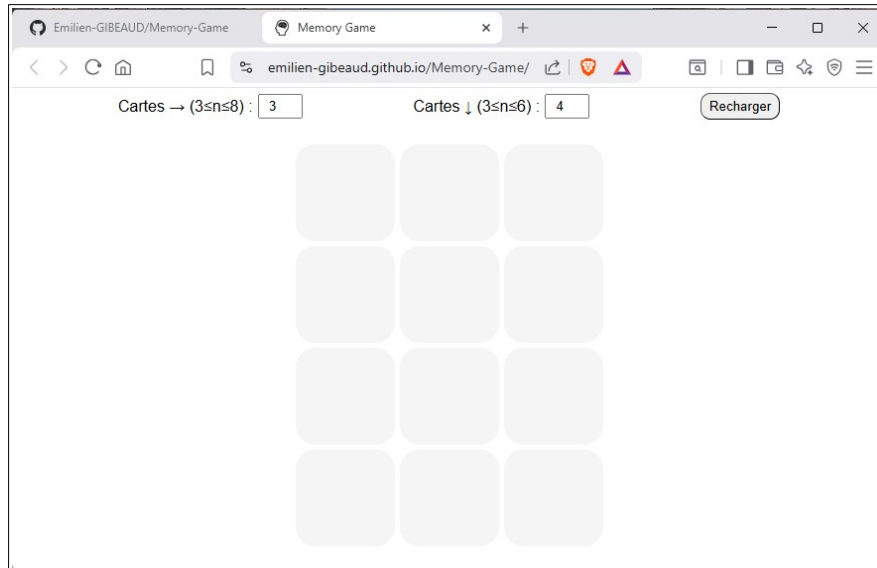
*(il ne s'agit pas du git pull ayant récupéré le commit mettant en place la GitHub Action)*



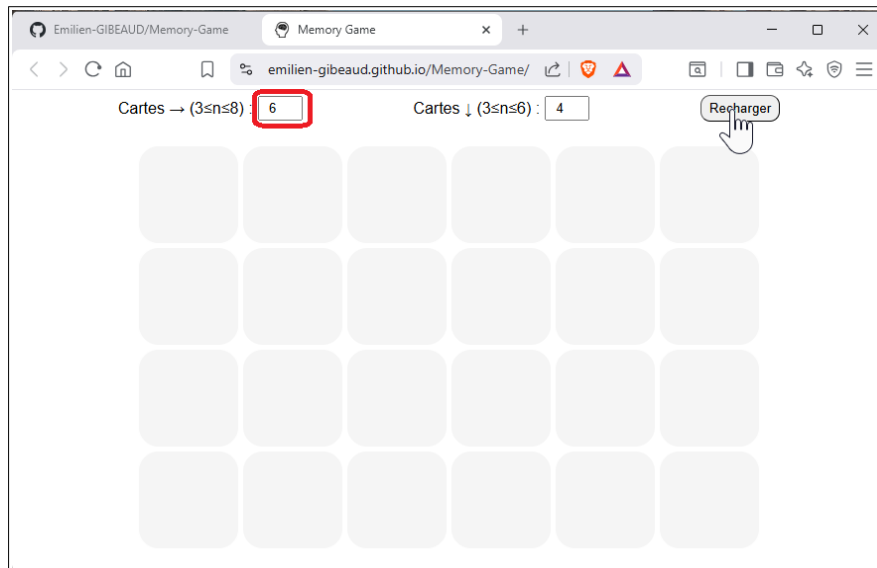
**Figure 7: Memory-Game – GitHub Action**



# DOSSIER PROFESSIONNEL <sup>(DP)</sup>

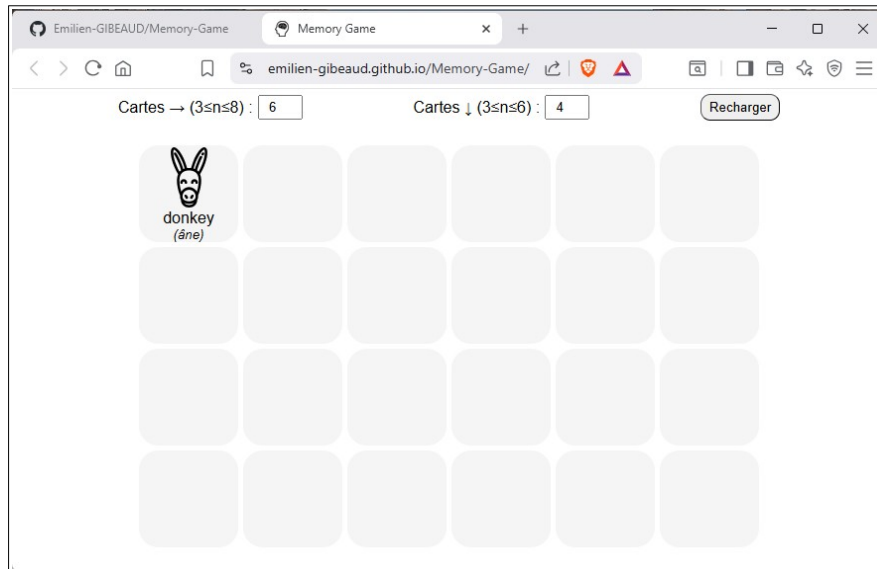


Le jeu se charge avec une grille de 3 x 4 (taille correspondant au mobile les plus petits).  
Il est ensuite possible de choisir sa taille de grille et de recharger la grille à tout moment (même sur un jeu en cours, on pourrait améliorer le jeu en bloquant cette possibilité).



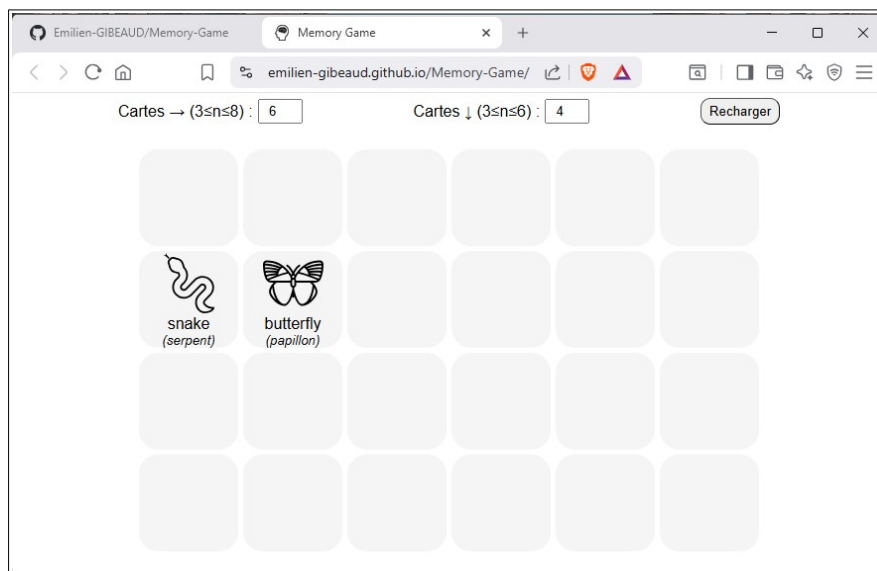
**Figure 8:** Memory-Game – Exemple jeu 1/3

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>



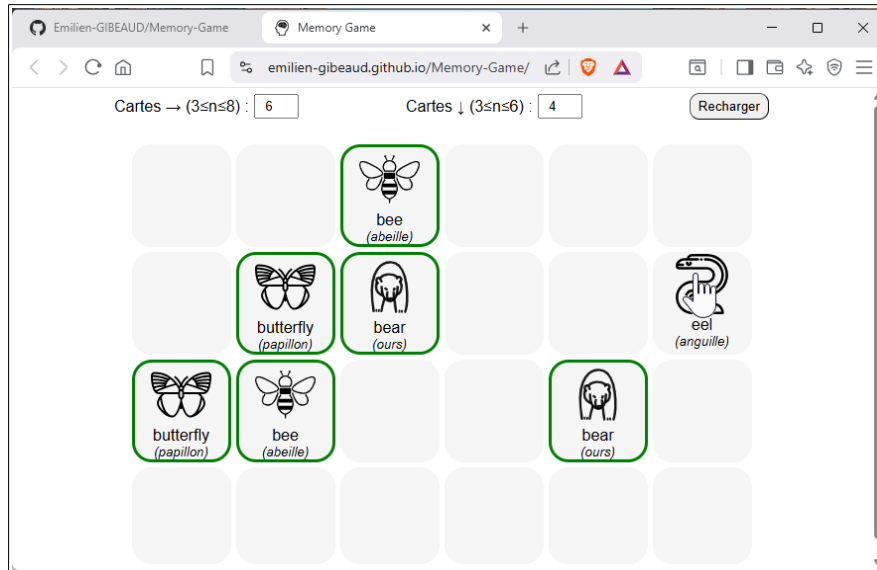
Sur l'image du haut, le joueur a sélectionné une carte, elle reste affichée tant qu'une seconde carte n'est pas sélectionnée.

Sur l'image du bas, le joueur a sélectionné deux cartes. Les cartes ne sont pas identiques, elles restent affichées durant une petite temporisation (afin de les mémoriser).



**Figure 9: Memory-Game – Exemple jeu 2/3**

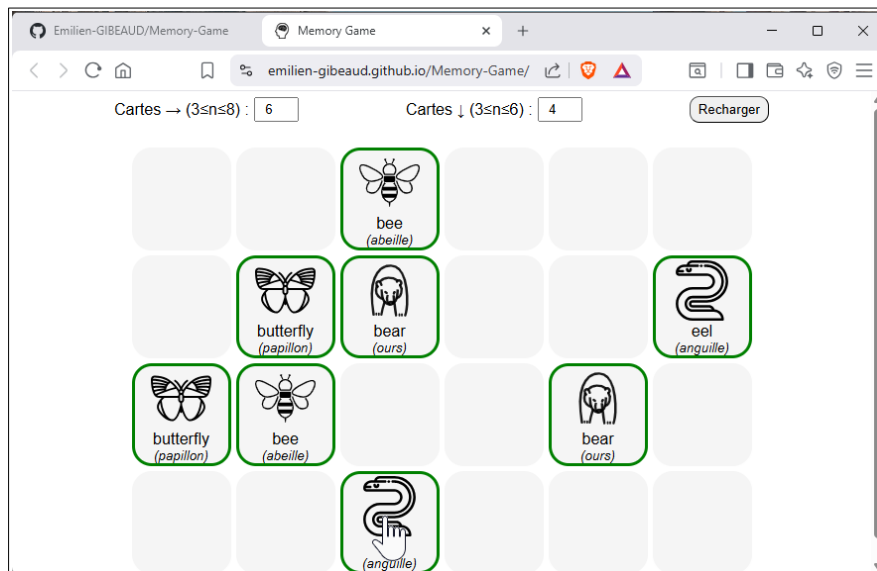
## DOSSIER PROFESSIONNEL <sup>(DP)</sup>



Une bordure verte est ajoutée autour des cartes quand une paire est trouvée.

Sur l'image du haut, le joueur a sélectionné une première carte (*et des paires ont été trouvées depuis les images de la page précédente*).

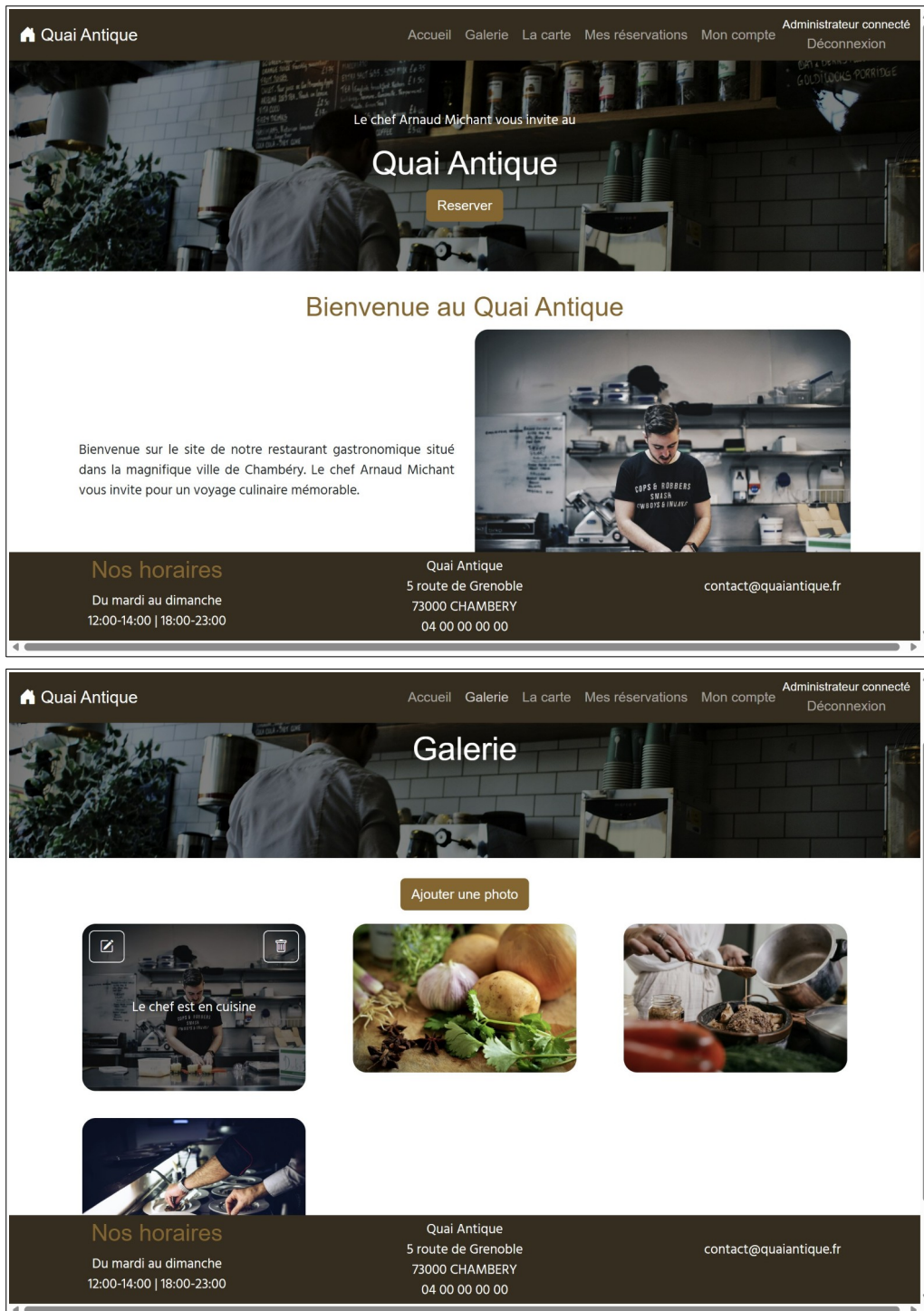
Sur l'image du bas, le joueur a sélectionné une deuxième carte. Les cartes étaient identiques, elles ont directement été validées avec une bordure verte (sans temporisation).



**Figure 10:** Memory-Game – Exemple jeu 3/3

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

### 7.3. QuaiAntique (à venir)



**Figure 11:** QuaiAntique – Plusieurs figures à venir



## 7.4. Au Bord de l'Eau (en cours)



Figure 12: Au Bord de l'Eau – Plusieurs figures à venir

# DOSSIER PROFESSIONNEL (DP)

GitHub Pages / Configure a custom domain / Manage a custom domain

## Configuring an apex domain

To set up an apex domain, such as `example.com`, you must configure a custom domain in your repository settings and at least one `ALIAS`, `ANAME`, or `A` record with your DNS provider.

- 1 On GitHub, navigate to your site's repository.
- 2 Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the ... dropdown menu, then click **Settings**.

octo-org / octo-repo (Private)

Code Issues 16 Pull requests 19 Discussions Actions Projects 8 Security 3 Insights Settings

- 3 In the "Code and automation" section of the sidebar, click **Pages**.
- 4 Under "Custom domain", type your custom domain, then click **Save**. If you are publishing your site from a branch, this will create a commit that adds a `CNAME` file directly to the root of your source branch. If you are publishing from a custom GitHub Actions workflow, no `CNAME` file is created, and any existing `CNAME` file is ignored and is not required. For more information about your publishing source, see [Configuring a publishing source for your GitHub Pages site](#).
- 5 Navigate to your DNS provider and create either an `ALIAS`, `ANAME`, or `A` record. You can also create `AAAA` records for IPv6 support. If you're implementing IPv6 support, we highly recommend using an `A` record in addition to your `AAAA` record, due to slow adoption of IPv6 globally. For more information about how to create the correct record, see your DNS provider's documentation.

- To create an `ALIAS` or `ANAME` record, point your apex domain to the default domain for your site. For more information about the default domain for your site, see [What is GitHub Pages?](#).
- To create `A` records, point your apex domain to the IP addresses for GitHub Pages.

185.199.108.153

185.199.109.153

185.199.110.153

185.199.111.153

Exemple ".tech domains" utilisé pour un portfolio personnel. Sera mis conforme lorsque le nécessaire sera fait sur OVH.

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://emilien-gibaud.tech/>  
Last deployed by [Emilien-GIBAUD](#) 2 weeks ago

[Visit site](#) [Unpublish site](#)

## Build and deployment

Source

GitHub Actions

Your site was last deployed to the `github-pages` environment by the `Deploy static content to Pages` workflow.  
[Learn more about deploying to GitHub Pages using custom workflows](#)

## Custom domain

Custom domains allow you to serve your site from a domain other than `emilien-gibaud.github.io`. [Learn more about configuring custom domains](#).

[Save](#) [Remove](#)

DNS Check in Progress

Admin Area for **emilien-gibaud.tech**

Manage Records for **emilien-gibaud.tech**

[A Records](#) [AAAA Records](#) [MX Records](#) [CNAME Records](#)  
[NS Records](#) [TXT Records](#) [SRV Records](#) [SOA Parameters](#)

Below is the list of Address (A) Records. Click the 'Add Address A Record' button to add more 'A' records. Alternatively you may click on any row to manage the corresponding 'A' record

### List of Address A Records

[Add A Record](#) [1](#) pg no [Jump To](#)


Sr No	Record Id	Name	Destination IP Address	Status
1	154531276	<a href="#">emilien-gibaud.tech</a>	185.199.108.153	Active
2	154531306	<a href="#">emilien-gibaud.tech</a>	185.199.109.153	Active
3	154531309	<a href="#">emilien-gibaud.tech</a>	185.199.110.153	Active
4	154531321	<a href="#">emilien-gibaud.tech</a>	185.199.111.153	Active

Exemple ".tech domains" utilisé pour un portfolio personnel. Sera mis conforme lorsque le nécessaire sera fait sur OVH.

Figure 13: Au Bord de l'Eau – Nom de domaine et gestion DNS

## DOSSIER PROFESSIONNEL <sup>(DP)</sup>

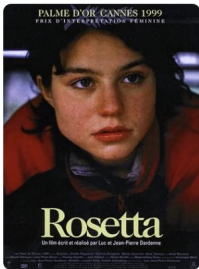
### 7.5. Moviz (à venir)

 **Moviz**

Accueil **Films**

**Admin :** [Ajouter film](#)

administrateur est connecté [Déconnexion](#)



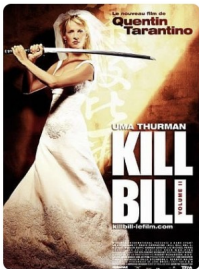
**Rosetta**

**Date de sortie :** 1999

**Réalisateur :** Jean-Pierre Dardenne, Luc Dardenne.

**Genre :** Drame, Social.

[Voir la fiche](#)




**Kill Bill: Volume 2**

**Date de sortie :** 2004

**Réalisateur :** Quentin Tarantino.

**Genre :** Action, Thriller, Drame, Crime.

[Voir la fiche](#)



**Kill Bill: Volume 1**

**Date de sortie :** 2003

**Réalisateur :** Quentin Tarantino.

**Genre :** Action, Thriller, Crime.

[Voir la fiche](#)

 **Moviz**

Accueil **Films**

**Admin :** [Ajouter film](#)

administrateur est connecté [Déconnexion](#)

## Ajouter un film

Titre :

Synopsis :

Image :

Choisir un fichier

Aucun fichier choisi

Format jpg, jpeg, png ou webp de 4 MB au maximum.

Année de sortie :

**Figure 14:** Moviz – Plusieurs figures à venir