



PROJET REALISATION DE JEU

Devine le nombre de 1 à 100 pour gagner

Version n° 1.5
21/11/2024

Ayyoub BELHASSEN & Yazid BRAHMI

Table des matières

Ayyoub BELHASSEN & Yazid BRAHMI.....	0
Table des matières	1
1. Contexte :	2
1-1. Contexte du projet : Devine le nombre	2
2. Liste des fonctions :	2
3-1. Devine le nombre	2
3. Description fonctionnelle :	3
4. Recette	4
5. Annexe	6
5-1. Fiche des répartitions des taches.....	6
5-2. Maquette du logiciel	7
5-3. Réalisation technique du logiciel	8
5-3-1.HTML De l'application.....	8
5-3-2.CSS de l'application	9
5-3-3. Javascript de l'application.....	10
5-3-4.HTML de l'aide F1.....	11
5-3-5.CSS de l'aide F1.....	12

1. Contexte :

1-1. Contexte du projet : Devine le nombre

Dans le cadre de notre apprentissage en développement web et en réseau, nous avons décidé de créer un jeu simple et amusant : Devine le nombre. Accessible via un navigateur web, ce jeu se joue seul et propose un défi logique qui met à l'épreuve la réflexion.

Le principe est simple : l'ordinateur choisit un nombre aléatoire entre 1 et 100, et le joueur doit le deviner. Après chaque essai, un message indique s'il doit tenter « Trop haut », « Trop bas » ou s'il a trouvé avec un « Bravo ! ».

Pour que le jeu fonctionne, il est hébergé sur une machine virtuelle (Debian), ce qui permet à tous les ordinateurs connectés au réseau local d'y accéder facilement. De plus, le jeu utilise plusieurs fonctions essentielles, qui seront décrites en détail dans le cahier des charges.

Ce projet a été conçu pour offrir une expérience ludique et accessible, tout en utilisant des concepts techniques avancés que nous détaillerons plus tard. Une belle façon de joindre l'utile à l'agréable !

2. Liste des fonctions :

3-1. Devine le nombre

Voici les différentes fonctionnalités du jeu Devine le nombre :

1. Plage de nombres définie : Le jeu te demande de deviner un nombre compris entre 1 et 100, ni plus, ni moins.
2. Bouton Envoyer : Une fois que tu as entré un nombre, il te suffit de cliquer sur le bouton Envoyer pour voir si tu es sur la bonne piste.
3. Bouton Rejouer : Si tu veux recommencer, le bouton Rejouer te permet de lancer une nouvelle partie avec un tout nouveau nombre à deviner. L'historique et le compteur de tentatives sont remis à zéro pour repartir de zéro.
4. Génération d'un chiffre aléatoire : Au début de chaque partie, le jeu choisit un nombre au hasard entre 1 et 100. À toi de le découvrir !
5. Historique des tentatives : Le jeu garde une trace de tous les nombres que tu as testés. Pratique pour éviter de refaire les mêmes erreurs !

6. Compteur de tentatives : Tu peux voir combien de fois tu as essayé de trouver le bon nombre. C'est une bonne façon de te challenger pour t'améliorer.
 7. Indications pour te guider : Si le nombre que tu proposes est trop petit ou trop grand, le jeu te le dira immédiatement pour te mettre sur la bonne voie.
 8. Affichage des résultats : Quand tu trouves le bon nombre, le jeu te félicite avec un message comme "Bravo !" ou "Félicitations !" pour marquer ta victoire.
- ### **3. Description fonctionnelle :**
- 1) Plage de nombres définie : Cette fonctionnalité permet de définir un nombre aléatoire entre 1 et 100. Chaque tirage le nombre aléatoire change.
 - 2) Bouton Envoyer : Le bouton Envoyer qui permet au joueur de soumettre sa tentative. Il sert de validation pour vérifier si le nombre choisi correspond au nombre aléatoire.
 - 3) Bouton Rejouer : Le bouton Rejouer permet le renouvellement du jeu. Une fois la partie terminée, le bouton "Rejouer" réinitialise complètement le jeu : génération d'un nouveau nombre aléatoire, effacement de l'historique des tentatives et remise à zéro du compteur d'essais.
 - 4) Génération d'un chiffre aléatoire : le programme génère automatiquement un nombre entre 1 et 100 de manière aléatoire. Cette fonction garantit l'imprévisibilité et l'équité du jeu, assurant que chaque partie sera unique.
 - 5) Historique des tentatives : Cette fonction permet au joueur de visualiser les nombres déjà testés. Cette fonctionnalité aide à éviter les répétitions et permet au joueur d'avoir certains indices sur le bon nombres.
 - 6) Compteur de tentatives : Cette fonction permet de suivre et afficher le nombre d'essais réalisés par le joueur. Il ajoute un élément de défi en permettant au joueur de mesurer sa performance et de chercher à deviner le nombre dans un minimum d'essais.
 - 7) Indications pour te guider : Après chaque tentative, le jeu fournit un retour immédiat guidant le joueur. Les messages "Trop petit" ou "Trop grand" offrent des indices précieux qui aident le joueur à trouver le bon résultat.
 - 8) Affichage des résultats : Cette fonction fournit des messages clairs et motivants, comme les indications de progression ("Trop petit", "Trop grand") et les messages de félicitations lors de la découverte du bon nombre.

4. Recette

1. Plage de nombres définie :

Ce qu'on a testé : On a vérifié que le jeu accepte uniquement les nombres entre 1 et 100 et on a aussi essayé d'entrer des valeurs hors plage (comme 0, 101) ou des caractères non valides (texte, symboles).

Ce qu'on attendait : Les nombres entre 1 et 100 doivent être acceptés. Si on entre une valeur incorrecte, un message d'erreur doit s'afficher, du type : « Erreur : entrez un nombre entre 1 et 100 ».

Résultat obtenu : Tout fonctionne comme prévu. Les saisies hors plage ou invalides sont bien rejetées avec un message clair.

2. Bouton Envoyer :

Ce qu'on a testé : Nous avons cliqué sur Envoyer après avoir saisi un nombre pour voir si le jeu réagit bien en affichant un message : “Trop grand”, “Trop petit” ou “Bravo”.

Ce qu'on attendait : Le message doit s'afficher correctement en fonction de la valeur saisie.

Résultat obtenu : Le bouton fonctionne parfaitement et les messages s'affichent à chaque tentative.

3. Bouton Rejouer :

Ce qu'on a testé : Après avoir terminé une partie, on a cliqué sur Rejouer pour voir si tout repart de zéro (nouveau nombre généré, historique et compteur réinitialisés).

Ce qu'on attendait : Le jeu doit redémarrer complètement avec un nouveau nombre à deviner et tout le reste remis à zéro.

Résultat obtenu : Le bouton “Rejouer” fait exactement ce qu'il doit faire : une nouvelle partie commence proprement.

4. Génération d'un chiffre aléatoire :

Ce qu'on a testé : On a vérifié que le nombre généré est bien aléatoire et reste constant pendant toute une partie.

Ce qu'on attendait : Le nombre doit changer à chaque nouvelle partie et rester identique tout au long de la partie.

Résultat obtenu : Les chiffres générés sont bien aléatoires, et il n'y a pas de changements imprévus pendant une partie.

5. Historique des tentatives :

Ce qu'on a testé : Nous avons saisi plusieurs nombres pour vérifier que chacun est bien enregistré dans l'historique et on a aussi vérifié que les valeurs déjà essayées restent visibles pour éviter de les répéter.

Ce qu'on attendait : L'historique doit afficher toutes les tentatives, avec précision et sans duplication.

Résultat obtenu : L'historique fonctionne comme prévu : toutes les tentatives apparaissent clairement à l'écran.

6. Compteur de tentatives :

Ce qu'on a testé : On a compté combien de fois on a essayé pour voir si le compteur s'incrémentecorrectement à chaque tentative.

Ce qu'on attendait : Le compteur doit afficher le bon nombre d'essais et se réinitialiser en cliquant sur Rejouer.

Résultat obtenu: Le compteur est fiable et remet tout à zéro à chaque nouvelle partie.

7. Indications “Trop petit” ou “Trop grand”

Ce qu'on a testé : Nous avons saisi des nombres pour vérifier si les messages “Trop petit” ou “Trop grand” sont bien affichés en fonction de la saisie.

Ce qu'on attendait : Les messages doivent guider le joueur clairement vers le bon nombre.

Résultat obtenu : Les indications sont précises et aident vraiment à trouver le nombre.

8. Affichage des résultats

Ce qu'on a testé : Nous avons trouvé le bon nombre pour vérifier si le jeu affiche un message de félicitations comme “Bravo !”.

Ce qu'on attendait : Le jeu doit afficher un message clair quand on gagne.

Résultat obtenu : Le message de victoire s'affiche correctement à chaque fois que le nombre est deviné.

Test de réseau (accessibilité sur le réseau local) :

Ce qu'on a testé : Nous avons installé le jeu sur une machine virtuelle sous Debian, avec le serveur Apache2 pour héberger l'application et nous avons vérifié que le jeu est accessible

depuis différents ordinateurs du réseau local, en ouvrant simplement le navigateur et en entrant l'adresse IP de la machine virtuelle sur laquelle le jeu est hébergé.

Ce qu'on attendait : Le jeu doit être accessible depuis tous les ordinateurs du réseau local, sans problème de connexion.

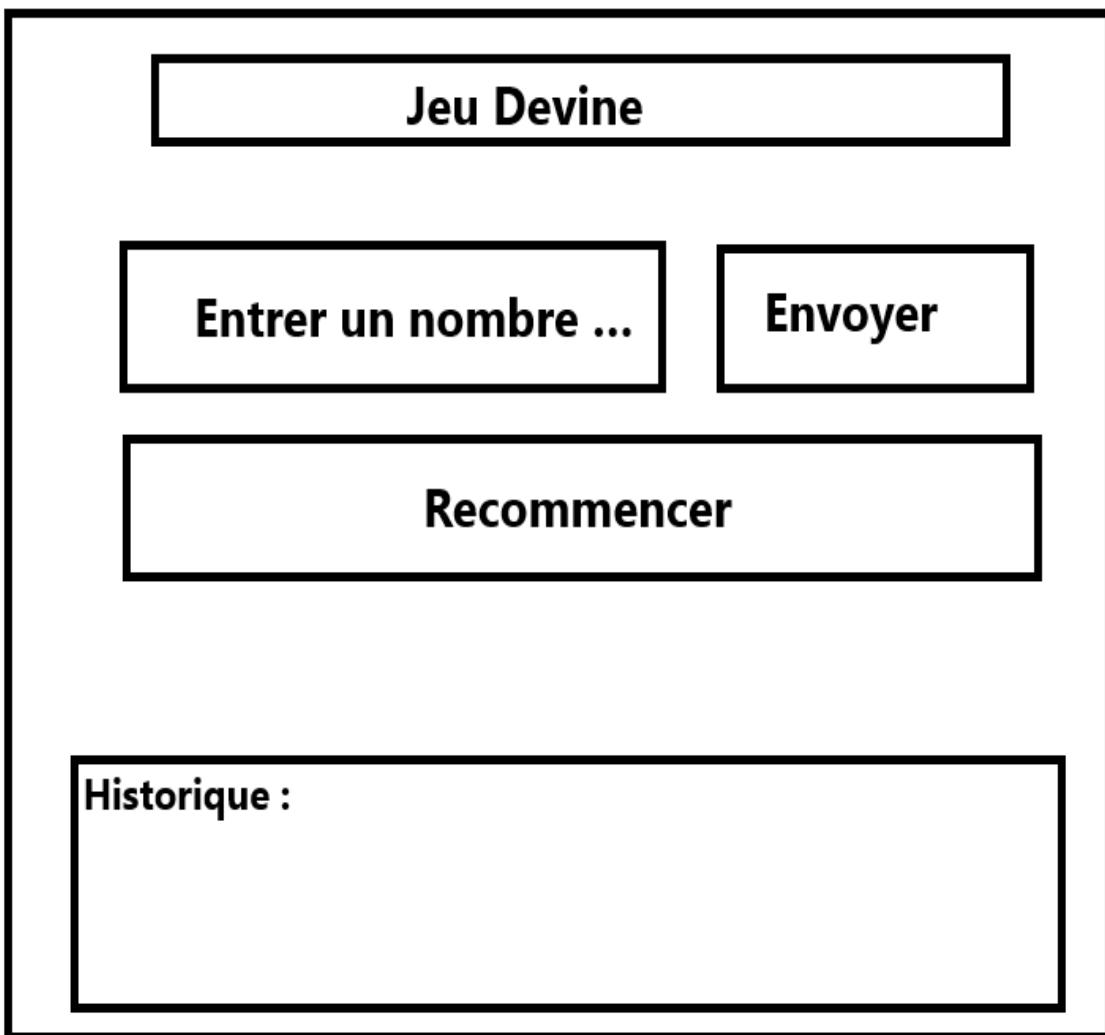
Résultat obtenu : Le jeu est correctement accessible depuis n'importe quel ordinateur du réseau local en utilisant l'adresse IP de la machine virtuelle. Aucun problème de connexion n'a été rencontré pendant les tests.

5. Annexe

5-1. Fiche des répartitions des tâches

Yazid Brahmi	Ayyoub Belhassen
-Jeu accessible par réseau - Cahier des charges - Documentation F1 - Fiche des répartitions des tâches	-Jeu accessible par réseau - Réalisation technique du logiciel - Documentation F1 - Maquette logiciel

5-2. Maquette du logiciel



5-3. Réalisation technique du logiciel

5-3-1.HTML De l'application

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Jeu Devine</title>
5          <link rel="stylesheet" type="text/css" href="cssjeudevine.css" />
6          <link rel="stylesheet" type="text/css" href="documentationF1.html" />
7          <script src="script.js" defer></script>
8      </head>
9      <body>
10         <div class="container">
11             <h1>Jeu Devine</h1>
12             <p>Devinez le numéro entre 1 et 100</p>
13             <input type="number" id="devine" min="1" max="100" />
14             <button id="envoyer">Envoyer</button>
15             <button id="rejouer">Rejouer</button>
16             <p id="message"></p>
17             <p id="tantatives"></p>
18
19             <div id="historique">
20                 <strong>Historique :</strong>
21             </div>
22         </div>
23     </body>
24 </html>
25
```

5-3-2.CSS de l'application

```
1  body {
2      font-family: Arial, sans-serif;
3      display: flex;
4      justify-content: center;
5      align-items: center;
6      height: 100vh;
7      background-color: #f4f4f9;
8      margin: 0;
9  }
10 .container {
11     text-align: center;
12     padding: 30px;
13     background-color: #fff;
14     border-radius: 10px;
15     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
16     width: 300px;
17 }
18 h1 {
19     margin-bottom: 20px;
20 }
21 .input-container {
22     display: flex;
23     justify-content: space-between;
24     align-items: center;
25     margin-bottom: 20px;
26 }
27 input[type="number"] {
28     padding: 10px;
29     font-size: 16px;
30     width: 70%;
31     text-align: center;
32 }
33 button {
34     padding: 10px 20px;
35     font-size: 16px;
36     margin: 10px;
37     cursor: pointer;
38     border: none;
39     border-radius: 5px;
40 }
41 .reset {
42     width: 100%;
43 }
44 #message {
45     margin-top: 10px;
46 }
47 #history {
48     margin-top: 20px;
49     text-align: left;
50     max-height: 150px;
51     overflow-y: auto;
52     padding: 10px;
53     background-color: #f7f7f7;
54     border-radius: 5px;
55 }
56 .error {
57     color: red;
58 }
59 .success {
60     color: green;
61     font-weight: bold
62 }
```

5-3-3. Javascript de l'application

```
45 // Mise à jour des tentatives
46 affichageTentatives.textContent = `Tentatives : ${tentativesCount}`;
47
48 // Mise à jour de l'historique
49 const nouvelEssai = document.createElement("p");
50 nouvelEssai.textContent = `Tentative ${tentativesCount}: ${nombre} (${resultat})`;
51 historique.appendChild(nouvelEssai);
52 }
53
54
55 // Fonction pour réinitialiser le jeu
56 function rejouer() {
57
58     nombreCible = Math.floor(Math.random() * 100) + 1;
59     tentativesCount = 0;
60     messageIndication.textContent = "";
61     affichageTentatives.textContent = "";
62     champDevine.value = "";
63     champDevine.disabled = false; // Réactive l'input
64     boutonEnvoyer.disabled = false; // Réactive le bouton
65     historique.innerHTML = "<strong>Historique :</strong>";
66 }
67
68 // Code pour ouvrir le fichier documentationF1.html quand F1 est pressé
69 document.addEventListener('keydown', function(event) {
70     // Vérifie si la touche F1 a été pressée
71     if (event.key === 'F1') {
72         event.preventDefault(); // Empêche l'action par défaut de F1 (qui ouvre l'aide du navigateur)
73         window.open("documentationF1.html", "_blank"); // Ouvre la documentation dans un nouvel onglet
74     }
75 });
76
77 // Ajout des événements
78 boutonEnvoyer.addEventListener("click", verifier);
79 boutonRejouer.addEventListener("click", rejouer);
80
81
82 // Variables pour le jeu
83 let nombreCible = Math.floor(Math.random() * 100) + 1;
84 let tentativesCount = 0;
85
86 // Récupération des éléments HTML
87 const champDevine = document.getElementById("devine");
88 const boutonEnvoyer = document.getElementById("envoyer");
89 const boutonRejouer = document.getElementById("rejouer");
90 const messageIndication = document.getElementById("message");
91 const affichageTentatives = document.getElementById("tentatives");
92 const historique = document.getElementById("historique");
93
94
95 // Fonction pour vérifier la tentative
96 function verifier() {
97
98     const valeur = champDevine.value;
99
100    // Validation de l'entrée
101    if (!valeur || isNaN(valeur) || valeur % 1 !== 0 || valeur < 1 || valeur > 100) {
102        messageIndication.textContent = "Entrée invalide. Veuillez entrer un nombre entier entre 1 et 100.";
103        messageIndication.style.color = "orange";
104        return; // Arrêter la fonction ici, aucune tentative ne sera comptée
105    }
106
107    const nombre = Number(valeur);
108    tentativesCount++;
109
110    let résultat = "";
111
112    if (nombre === nombreCible) {
113        messageIndication.textContent = "Bravo ! Vous avez réussi à gagner !";
114        messageIndication.className = "success";
115        champDevine.disabled = true; // Désactive l'input
116        boutonEnvoyer.disabled = true; // Désactive le bouton
117        résultat = "Correct !";
118    } else if (nombre < nombreCible) {
119        messageIndication.textContent = "Trop petit. Essayez encore.";
120        messageIndication.className = "error";
121        résultat = "Trop petit";
122    } else {
123        messageIndication.textContent = "Trop grand. Essayez encore.";
124        messageIndication.className = "error";
125        résultat = "Trop grand";
126    }
127 }
```

5-3-4.HTML de l'aide F1

```
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   <title>Documentation F1 - Devine le Nombre</title>
7   <link href="documentationf1.css" type="text/css" rel="stylesheet" media="all">
8 </head>
9 <body>
10
11 <h1>Page d'aide - Application "Devine le Nombre"</h1>
12
13 <div class="section">
14   <h2>1. Objectif</h2>
15   <p>
16     L'application permet à l'utilisateur de deviner un nombre secret entre 1 et 100. À chaque essai, l'utilisateur reçoit un indice si le nombre est trop petit ou trop grand. Le jeu se termine lorsqu'
17   </p>
18 </div>
19
20 <div class="section">
21   <h2>2. Fonctionnalités</h2>
22   <ul>
23     <li><strong>Entrée d'un nombre :</strong> L'utilisateur entre un nombre entier entre 1 et 100.</li>
24     <li><strong>Indices :</strong> Le programme indique si le nombre proposé est trop petit ou trop grand.<br></li>
25     <li><strong>Message de succès :</strong> Lorsque l'utilisateur trouve le bon nombre, un message de félicitations s'affiche.</li>
26     <li><strong>Historique des tentatives :</strong> Toutes les tentatives sont enregistrées et affichées.</li>
27     <li><strong>Réinitialisation :</strong> Le jeu peut être réinitialisé pour recommencer une nouvelle partie.</li>
28   </ul>
29 </div>
30
31 <div class="section">
32   <h2>3. Comment ça marche ?</h2>
33   <p>
34     - L'utilisateur entre un nombre dans un champ prévu à cet effet.<br>
35     - Si le nombre est incorrect, un message indique si le nombre est trop grand ou trop petit.<br>
36     - Si l'utilisateur devine le bon nombre, un message de succès apparaît.<br>
37     - L'utilisateur peut recommencer à tout moment en cliquant sur "Rejouer".
38   </p>
39 </div>
40
41 <div class="section">
42   <h2>4. Interface Utilisateur</h2>
43   <ul>
44     <li><strong>Champ de saisie :</strong> Permet à l'utilisateur d'entrer un nombre.</li>
45     <li><strong>Button "Envoyer" :</strong> Soumet la tentative de l'utilisateur.</li>
46     <li><strong>Button "Recommencer" :</strong> Réinitialise le jeu pour une nouvelle partie.</li>
47     <li><strong>Zone de messages :</strong> Affiche les messages d'indication ou de succès.</li>
48     <li><strong>Historique :</strong> Affiche les tentatives précédentes de l'utilisateur.</li>
49   </ul>
50 </div>
51
52 <div class="section">
53   <h2>5. Limitations</h2>
54   <ul>
55     <li>Le nombre doit être un entier entre 1 et 100.</li>
56     <li>Les entrées non numériques ou en dehors de cette plage sont rejetées.</li>
57   </ul>
58 </div>
59
60 <div class="section">
61   <h2>6. Conclusion</h2>
62   <p>
63     "Devine le Nombre" est un jeu simple et amusant où l'utilisateur essaie de deviner un nombre généré aléatoirement. Il fournit des indices après chaque tentative et garde une trace des essais précédents.
64   </p>
65 </div>
66
67 </body>
68 </html>
```

5-3-5.CSS de l'aide F1

```
1      body {
2          font-family: Arial, sans-serif;
3          margin: 20px;
4          background-color: #f4f4f9;
5      }
6      h1 {
7          color: #333;
8      }
9      h2 {
10         color: #444;
11     }
12     p {
13         line-height: 1.6;
14     }
15     .section {
16         margin-bottom: 20px;
17         padding: 15px;
18         background-color: #fff;
19         border-radius: 8px;
20         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
21     }
22     .important {
23         color: #e74c3c;
24         font-weight: bold;
25     }
26
27
```