

Cahier des charges

Rémi Simonetto / Walid Hajjout

v1.6 11/12/24

Sommaire

Table des matières

Contexte	3
Description Fonctionnelle	4
Liste Fonctions	5
Générer un nombre aléatoire entre 1 et 100	5
Avoir un formulaire	5
Saisir une valeur	5
Vérifier nombre	5
Vérifier écart	5
Vérifier résultat	5
Afficher le résultat du test	5
Afficher bouton « rejouer »	5
Mise en mémoire historique	5
Afficher l'historique	5
Recette	6
Objectif de la recette	6
Tests à effectuer	6
Génération de nombre aléatoire	6
Réponse aux tentatives	6
Historique des tentatives	6
Bouton « rejouer »	6
Affichage des messages	6
Scénario de test	6

Contexte

Ce projet vise la création d'une application en ligne ludique dont l'objectif principal est de concevoir un jeu interactif de devinette de nombres. Ce jeu repose sur un principe simple : l'utilisateur doit deviner un nombre choisi aléatoirement par l'application dans une plage définie (par exemple, entre 1 et 100). Au fur et à mesure des tentatives, des indices sont donnés pour guider l'utilisateur, tels que "plus grand" ou "plus petit".

Description Fonctionnelle

1. Fonctionnalité principale : Devinette d'un nombre

- **Description** : L'application génère un nombre aléatoire dans une plage définie (entre 1 et 100). L'utilisateur doit deviner ce nombre en entrant des propositions.
- **Interactions utilisateur** :
 - L'utilisateur saisit un nombre via un champ de saisie.
 - L'application répond en indiquant si le nombre à deviner est "Trop bas", "Trop haut" ou "correct".
- **Règles du jeu** :
 - Un compteur de tentatives est affiché lorsqu'on trouve le bon nombre
 - L'utilisateur gagne lorsqu'il trouve le bon nombre, et un message de succès est affiché.
 - Une fois le jeu terminé, l'utilisateur peut recommencer une nouvelle partie.

2. Gestion des plages de nombres

- **Paramètre de base** :
 - Plage minimale et maximale (1-100).

3. Feedback utilisateur

- **Messages** :
 - Indices : "Trop haut !" ou "Trop bas !".
 - Confirmation : "Bravo, vous avez trouvé en X tentatives !"
 - Erreur : "Veuillez entrer un nombre valide, entre 1 et 100 compris."

Aide :

- Touche F1 qui ouvre la documentation utilisateur
-

4. Interface utilisateur

- **Design simple et intuitif** :
 - Un écran de jeu principal avec un champ de saisie et un bouton de validation

5. Fonctionnalités techniques

- **Côté serveur** :
 - Génération du nombre aléatoire et validation des entrées.
 - Entrée en mémoire de l'historique
 - Vérification du nombre

Liste Fonctions

Générer un nombre aléatoire entre 1 et 100

Le programme doit générer un nombre aléatoire choisi entre 1 et 100 compris.

Avoir un formulaire

Le site doit disposer d'un formulaire dans lequel l'utilisateur peut renseigner sa réponse afin de l'envoyer au programme.

Saisir une valeur

L'utilisateur doit pouvoir saisir une valeur dans le formulaire.

Vérifier le nombre

Le programme doit vérifier que l'utilisateur a bel et bien saisi un nombre dans le formulaire, tout autres caractères seront refusés

Vérifier l'écart

Le programme doit vérifier si le nombre saisi par l'utilisateur est plus grand ou plus petit que le nombre choisi aléatoirement.

Vérifier le résultat

Le programme doit vérifier si le nombre saisi par l'utilisateur est égal au nombre choisi aléatoirement.

Afficher le résultat du test

Le site doit afficher le résultat du texte, que ce soit « Trop haut ! », « Trop bas ! » ou « Bravo, vous avez trouvé en X tentatives ! ».

Afficher le bouton « Rejouer »

Si l'utilisateur a saisi le nombre choisi aléatoirement, afficher un bouton « Rejouer » qui recharge la page.

Mise en mémoire de l'historique

Mise en mémoire des saisies de l'utilisateur ainsi que de l'affichage du résultat qui correspond.

Afficher l'historique

Afficher l'historique mis en mémoire.

Recette

Objectif de la recette

L'objectif de cette recette est de vérifier que le jeu en ligne développé fonctionne correctement, conformément aux spécifications et aux exigences définies dans le cahier des charges. Cette phase de tests a pour but de garantir que toutes les fonctionnalités du jeu sont opérationnelles, que l'interface utilisateur est fluide, et que le serveur répond correctement aux actions de l'utilisateur.

Tests à effectuer

Génération de nombre aléatoire

Vérifier que le nombre généré est bien compris entre 1 et 100 (bornes comprises).

Réponse aux tentatives

Vérifier que le jeu affiche bien « Trop haut ! », « Trop bas ! » ou « Bravo, vous avez trouvé en X tentatives! » selon la tentative de l'utilisateur.

Historique des tentatives

Vérifier que l'historique des tentatives est correctement affiché et mis à jour.

Bouton « rejouer »

Vérifier que le bouton « rejouer » apparaît lorsque l'utilisateur trouve la bonne réponse et que la partie réinitialise correctement en rechargeant la page.

Affichage des messages

Vérifier la bonne communication des messages (victoire, erreurs, etc.).

Scénario de test

- Un joueur commence une partie, fait plusieurs tentatives, puis trouve la bonne réponse.
- Un joueur entre un nombre trop élevé ou trop bas plusieurs fois avant de trouver la bonne réponse
- Un joueur clique sur « rejouer » après avoir trouvé le nombre sélectionné aléatoirement