

# Rapport de Synthèse du Projet - Hangman Web

Date : Novembre 2024

---

Équipe de Développement :

- Massinissa Ahfir
  - Anthony FONTAINE
- 

Encadré par : Cyril Rodrigues

---

Description :

Projet réalisé dans le cadre du module de développement web en Golang, visant à créer une interface utilisateur pour le jeu du pendu (Hangman).

---

## 1. Introduction :

Le projet **Hangman Web** a pour objectif de rendre jouable, via une interface graphique web, le jeu du pendu initialement développé en ligne de commande. En utilisant le langage **Golang**, nous avons réutilisé notre module CLI existant en y ajoutant une interface utilisateur web moderne.

---

## 2. Décomposition du Projet :

### 2.1 Phases Clés

#### Phase 1 : Conception et Spécifications (2 jours)

- Identification des fonctionnalités à réutiliser depuis la version CLI.
- Définition des nouvelles fonctionnalités à implémenter pour la version web (pages HTML, gestion des sessions, stockage des scores).
- Création d'un diagramme de flux pour les interactions utilisateur.
- Choix du thème et style qui serviront pour le design du style
- Création du repository Git et structure initiale du projet

#### Phase 2 : Développement (10 jours)

- **Backend :**
  - Développement du serveur HTTP en Golang pour gérer les requêtes, les réponses, et les templates HTML.
  - Migration du code Hangman CLI.
  - Création des templates de base.
  - Mise en place de la logique de jeu.
- **Frontend :**
  - Développement des templates HTML.
  - Création du design CSS.
  - Implémentation des fonctionnalités interactives.
  - Tests utilisateur et ajustements
- **Intégration :** Connexion des routes HTTP aux méthodes du module CLI existant.

### **Phase 3 : Fonctionnalités Avancées (7 jours)**

- Système de scores
- Persistance des données
- Fonctionnalité bonus

### **Phase 4 : Tests & Débogage (2 jours)**

- Tests unitaires pour valider les fonctionnalités du backend.
- Tests fonctionnels pour vérifier le bon affichage des pages et le flux de jeu.
- Débogage des problèmes liés aux sessions et à la persistance des scores.

### **Phase 5 : Finalisation & Déploiement (2 jours)**

- Documentation
- Tests finaux
- Préparation de la présentation
- Revue de code finale
- Instructions d'installation documentées dans le README.md.

### **2.2 Jalons critiques :**

- ✓ J+2 : Architecture validée
  - ✓ J+12 : Core fonctionnel
  - ✓ J+19 : Interface complète
  - ✓ J+21 : Tests terminés
  - ✓ J+23 : Documentation finalisée
-

### 3. Organisation du travail :

#### 3.1 Répartition des tâches :

Membre de l'Équipe	Rôle	Tâches
Massinissa AHFIR	Full Stack	Migration du code Hangman , Implémentation serveur http, Logique de jeu, Tests unitaires, Système de scores, Persistance des données, Documentation , Support général.
Antony FONTAINE	Frontend	Templates HTML, Design CSS, Interface utilisateur, Tests d'intégration.

#### 3.2 Stratégie d'organisation :

##### Méthodologie Agile Simplifiée

- Sprints de 2-3 jours
- Daily standup de 15 minutes
- Revue de code par les pairs
- Tableau Kanban digital pour le suivi

##### Outils Utilisés

- GitHub : Gestion de versions et collaboration
  - Discord : Communication d'équipe
  - Trello : Suivi des tâches
  - Google Docs : Documentation partagée
-

## **4. Gestion du Temps :**

### **4.1 Priorisation :**

#### **1. Must Have (Priorité 1)**

- Core du jeu
- Interface basique
- Gestion des parties

#### **2. Should Have (Priorité 2)**

- Système de scores
- Design avancé
- Messages dynamiques

#### **3. Nice to Have (Priorité 3)**

- Fonctionnalités bonus
- Animations
- Optimisations

### **4.2 Gestion des Risques**

- Buffer de sécurité de 3 jours
- Liste des solutions de contournement
- Documentation des problèmes rencontrés

### **4.3 Planning quotidien**

- **8h30 : Daily standup**
  - **8h45-12h30 : Développement**
  - **14h00-16h30 : Développement**
  - **16h30-17h00 : Review et planification**
-

## **5. Stratégie de Documentation**

### **5.1 Documentation Technique**

- Documentation Go avec godoc
- Commentaires de code inline
- README détaillé sur GitHub
- Diagrammes d'architecture

### **5.2 Sources d'Information**

- Documentation officielle Go
- Stack Overflow pour les problèmes spécifiques
- Retours d'expérience d'autres projets similaires
- Documentation

### **5.3 Organisation de la Documentation**

- Templates standardisés
- Revue de documentation hebdomadaire
- Mise à jour continue

## **6. Conclusion**

Le projet Hangman Web a permis à notre équipe d'améliorer nos compétences en Golang et en développement web. Nous avons réussi à réutiliser notre projet existant tout en l'enrichissant avec une interface utilisateur web.