



UE L220 JavaScript avancé
Rapport de mini-projet
Le pendu

Groupe 2

Julien Biechler - Bryan Gast - Damien Cuda

Table des matières

Le projet.....	3
Contraintes du projet	3
Système du jeu	3
Système de score.....	4
Gestion de projet.....	4
Aspects techniques.....	4
Technologies.....	4
Architecture du dossier	4
Développement des étapes de jeu.....	5
Login	5
Construction de la page de jeux et initialisation	5
Déroulement du jeu	7
Paramètres du jeu	8
Répartition des tâches.....	9

Le projet

Le projet est de reproduire en groupe un jeu de type pendu. Nous serons trois à travailler sur ce projet¹ : Bryan Gast, Julien Biechler et Damien Cuda.

Contraintes du projet

Trois fonctionnalités ont été imposées :

- Avant d'accéder au jeu, l'utilisateur devra entrer son pseudo
- Le jeu devra intégrer un système de score
- Le jeu devra permettre au joueur de choisir un niveau de difficulté

Système du jeu

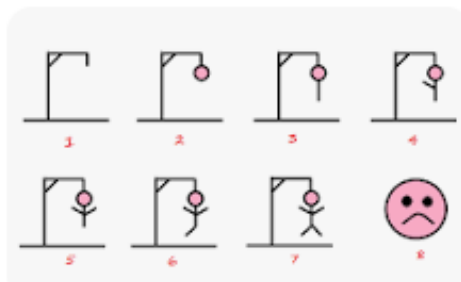
Le joueur **doit** entrer un pseudo et choisir entre 3 niveaux de difficulté :

- Facile : Mot de 5 lettre à trouver
- Normal : Mot de 7 lettre à trouver
- Difficile : Mot de 10 lettre à trouver

Le joueur, à l'aide du clavier virtuel ou physique, doit trouver le mot mystère en essayant de deviner les lettres qui le compose.

- 1) S'il la lettre choisies est dans le mot mystère, elle apparait.
- 2) Sinon une portion du pendu apparait.

Il a le droit à un maximum de 7 erreurs. Chaque erreur fait apparaitre une section du pendu selon le schéma suivant.



Le jeu s'arrête quand le joueur a découvert le mot ou épuisé toutes ses chances. (Voir [système de score](#)).

Une fois le mot trouvé ou le pendu complet le joueur peut essayer de trouver un nouveau mot ou quitter la partie.

¹ Initialement à quatre mais à part sa courte présence à la première réunion de groupe Josue Denis Caman n'a pas participé au projet.

Système de score

En cas de victoire :

Le joueur marque 5 points auxquels sont ajoutés 1 point par tronçon du pendu non découvert multiplié par un coefficient défini par le niveau de difficulté.

En cas de défaite :

Le joueur perd 5 points auxquels sont retirés 1 point par tronçon du pendu découvert multiplié par un coefficient défini par le niveau de difficulté.

Gestion de projet

Nous avons utilisé la solution de versioning [GitHub](#) pour le développement et la solution [Trello](#) pour le suivi des tâches.

Nous avons lors de la première réunion d'équipe défini les grandes lignes du projet ainsi que les technologies que nous utiliserions.

Lors de réunions synchrones sur Discord nous exposions nos avancés respectives ainsi que les éventuels problèmes rencontrés.

Nous faisons ensuite un point sur les étapes restantes et nous repartissions ainsi le travail à faire.

Aspects techniques

Technologies

Les pages seront développées en HTML et stylisées en CSS avec l'appui du Framework Bootstrap afin de s'appuyer sur les modules proposés par ce dernier.

La librairie JQuery sera utilisée pour la partie JavaScript afin d'accélérer le développement.

Les deux librairies seront ajoutées au projet afin de permettre une utilisation hors-ligne.

Architecture du dossier

- 2 pages HTML :
 - index.html (page de login)
 - game.html (page de jeu)
- Un dossier "css":
 - index.html & index.css (page de login)
 - game.html & game.css (page de jeu)
- Un dossier "js" avec :
 - index.js
 - games.js
 - hangman.js (contenant les fonctions de construction du canvas du pendu)
 - Un dossier json (contenant les tableaux de mots)
- Dossier "svg" (les SVG d'icônes)

- Dossier "sound" (son du jeu)
- Dossier app-assets (les librairies Bootstrap et JQuery)
- Dossier "font" (font d'icônes)
- Dossier "asset" (stock d'images)

Développement des étapes de jeu

Login

L'utilisateur entre son pseudo et choisit un niveau de difficulté

Au clique sur le bouton Start ou la touche entrée une fonction vérifie que le pseudo est :

- Non vide

```
if ($(pseudo_input).val() == ""){
  e.preventDefault();
  $(error_message).html('Vous devez entrer votre pseudo pour jouer 🚫') //Affichage message d'erreur si pas de pseudo choisit
}
```

- Valide (pas de caractères permettant l'injection de code)

```
}else{
  let regex = /^[a-z0-9]+$/gi;
  let pseudo = escape($(pseudo_input).val());
  if(regex.test(pseudo)){S
```

Si ces critères sont respectés les pseudo et niveau sont récupérés dans le sessionStorage.

Construction de la page de jeux et initialisation

Les pseudo, niveau et autres élément récupéré de la page de login sont injecté dans la page de jeu :

```
$("#current_difficulty").html(difficulty_level); //Injection du niveau de difficulté
$("#player_pseudo").html(player_pseudo); //Injection du pseudo
$("#player_score").html(player_score); //Injection du score
```

Le clavier virtuel est généré.



Une section d'affichage des lettres est générée avec des underscores pour commencer.

_ _ _ _ _

Le mot mystère est choisit de manière aléatoire grâce à l'appel AJAX sur le fichier JSON correspondant au niveau de difficulté et est retourné sous forme de promesse dans une variable:

```
//Fonction de choix du mot
function generateWord(){
  return new Promise((resolve, reject) => {

    let json; //La variable json accueillera le nom du fichier json en fonction du niveau choisit

    switch(coef){
      case "1.5":
        json = "wordeasy.json";
        break;
      case "2":
        json = "wordmedium.json";
        break;
      case "2.5":
        json = "wordhard.json";
        break;
    }

    $.ajax({
      url: "../js/json/" + json,
      type: "GET",
      data: {},
      dataType: "json",
      success: function(reponse) {
        let random;
        for(let i = 0; i < reponse.length; i++){
          random = Math.floor(Math.random() * reponse.length)
        }
        mystery_word = reponse[random];
        resolve(mystery_word);
      },
      error: function(error){
        reject(alert("Le mot n'a pu être généré"));
      }
    });
  });
}
```

Le jeu est ensuite initialisé et les fonctions de vérification sont déclenchées au clique sur les claviers.

```
var mystery_word = generateWord(); //Le mot mystère est généré et passé à la variable sous forme de promesses

mystery_word
  .then(value => {
    init(value); //Le résultat de la promesse sert à initialiser le jeu
  });
```

Le jeu est prêt !

Déroulement du jeu

Le joueur clique sur une lettre

-> Vérification de la présence de la lettre dans le mot mystère :

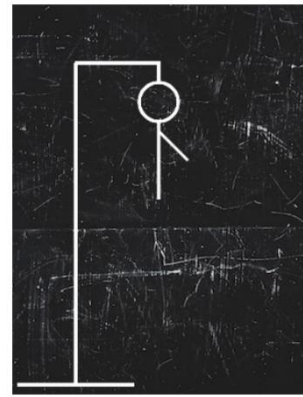
- Si présent (1)
 - > affichage de la lettre à la bonne place et update du score
 - > vérification si le mot est complètement trouvé ou non (2):
 - Si oui déclenchement la fonction de réussite est lancée et la modale correspondante apparait.
 - Si non le jeu continue
- Si erreur

-> le score est mis à jour

-> le compteur d'erreur est incrémenté et la section du pendu correspondant s'affiche dans le canvas (3)

-> vérification du compteur d'erreur :

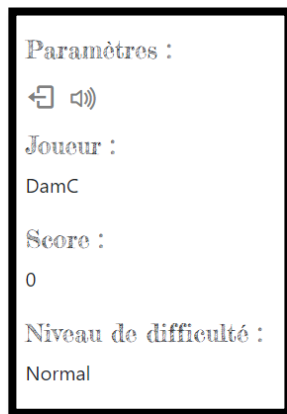
- Si compteur d'erreur est = à 7 (nombre max d'erreur) la fonction d'échec est lancée et la modale correspondante apparait.



Le joueur a ensuite la possibilité de continuer à jouer avec un nouveau mot ou de quitter la partie.

- (1) La présence de la lettre s'effectue grâce au passage du mot mystère en array et de la création d'un array temporaire de même longueur rempli de _. Ainsi il est possible de vérifier si la lettre est présente mais également de récupérer son index pour :
 - L'afficher au bon endroit pour le joueur
 - De remplacer le _ dans l'array temporaire
- (2) L'array du mot mystère et l'array temporaire doivent être identique pour définir si le mot a été entièrement trouvé.
- (3) Un switch est utilisé sur le compteur d'erreur pour appeler la bonne fonction de construction du canvas

Paramètres du jeu



Le joueur à la possibilité sur la page de jeu de couper les sons du jeu et de quitter la partie en cours.

Son pseudo est affiché ainsi que le score général de sa partie (le score de la manche étant affiché dans la modale de fin de manche)

Le niveau de difficulté choisi sur la page de login est également affiché ici.

Répartition des tâches

Bryan Gast	Création du canvas avec le pendu et des fonctions de hangman.js Complément start_game de index.js (regex, switch coeff et messages d'erreur) Fonctions de score Fonctions de lose et win Fonction init Fonction verif Fonction clear canvas Fonction restart Adaption responsive
Julien Biechler	Création des fichiers json pour les mots et proposition d'idées quand a certain concept du jeu et logique.
Damien Cuda	Création dépôt Git Création et rédaction Trello Création structure des dossiers Structure et code HTML des pages index.html et game.html Fonction start_game de index.js Création clavier virtuel et écoutes des claviers generate_keyboard Fonction hidden_word Fonction show_letter Fonction hangman_step Commentaires et organisation générale du code

Les attributions de créations de fonctions notées ci-dessus doivent toutefois prendre en compte que chaque membre de l'équipe a adapté, modifié et/ou amélioré le travail des autres si besoin était.

La fonction generateWord (AJAX et promesse) est une création commune.