

# Cookie clicker

Ou Clicker game

## Avant-propos

Votre objectif sera de faire le design et créer un *cookie clicker* ou *clicker game*, au sein d'une application web, ainsi que de mettre en ligne sur o2switch votre création.

Le projet est à effectuer en groupe de 3 personnes selon les groupes de TP. Personne ne doit être seul !

<https://tinyurl.com/MMI-S3-SAE-301-GROUPES>

Technologies attendues :

- HTML
- JavaScript (JS) ou TypeScript (TS)
- CSS

Librairies :

- JQuery autorisé
- Libraires CSS interdites (Bootstrap, Animate, Masonry...)
- Librairies JavaScript interdites

Navigateurs (au choix parmi la liste suivante) :

- Chrome
- Edge
- Firefox

Il faudra nous préciser le navigateur que vous aurez choisi.

### Attention !

- Si vous ne respectez pas ces règles, votre note sera de 0.
- Si vos fichiers ne fonctionnent pas, votre note sera de 0.  
Utilisez des liens relatifs plutôt qu'absolus.

Récapitulatif des dates :

- **Rendu de la maquette** : dimanche 04/12/2022 à 23h59
- **Rendu du projet entier** : lundi 02/01/2023 à 23h59
- **Présentation orale** : vendredi 06/01/2023

# Cahier des charges

## Pages à réaliser

Votre projet sera constitué de 3 pages :

- Accueil
- Page du jeu
- Page de classement

### Accueil

Pour pouvoir lancer le *cookie clicker* et inscrire son nom dans le classement, l'utilisateur doit s'enregistrer à l'aide d'un pseudonyme. Une fois le champ complété, le bouton pour lancer la partie devient disponible et il peut accéder à la page du jeu.

### Page du jeu

Cette page est le cœur de l'application web, contenant le *cookie clicker*, les *bonus*, les *succès*...

Elle doit contenir un lien vers la page de classement.

### Page de classement

Ouverte dans un nouvel onglet, la page de classement recense les scores de tous les utilisateurs de l'application web.

## Personnalisation

### Choix d'un thème

Votre *clicker* doit être original. Vous devez trouver un thème puis choisir vos couleurs, vos éléments, vos arrière-plans en fonction de ce thème.

### Changements visuels en cours de partie

Afin de rendre le jeu plus interactif, lorsque le joueur active pour la 1<sup>ère</sup> fois un *bonus*, le thème doit évoluer (décor, éléments animés, arrière-plan etc).

## Fonctionnalités

### Le *clicker*

L'élément central de votre jeu, un *cookie* cliquable qui génère des *points*.

Au clic sur cet élément le compteur de *points* doit être mis à jour, ainsi que la disponibilité des différents *bonus* achetables en échange de *points*.

**Cet élément doit être une image !**

### La génération de *points*

#### Les *points* au clic

Lorsque l'utilisateur clic sur le *cookie*, il gagne 1 *point*.

Les *bonus* qui modifient la génération de *points* au clic sont cumulatifs : si l'un de vos *bonus* augmente de 10 le gain de *points*, et que votre joueur en possède 4, alors le gain est de  $10 * 4 \Rightarrow 40$  *points*.

#### Les *points* automatiques

Par défaut, toutes les secondes, vous générez 1 *point*.

Les différents *bonus* qui modifient la génération automatique de *points* sont appliqués de la même façon que pour le clic : si l'un de vos *bonus* augmente de x2 *points* la génération automatique, et que votre joueur en possède 2, alors le *bonus* est cumulatif ( $2 * 2 \Rightarrow 4$ ).

### Les *bonus*

À mesure que l'utilisateur joue, il accumule des *points*. Afin d'offrir quelques options de gameplay supplémentaires, nous souhaitons mettre à sa disposition des *bonus*. Ils sont obtenus en échange de *points*.

Les *bonus* sont présentés sous la forme d'éléments cliquables. Chacun de ces *bonus* est lié à un compteur qui lui est propre, afin que l'utilisateur puisse savoir combien de ces *bonus* il a en sa possession.

Lorsque l'utilisateur survole un *bonus*, il pourra voir le détail de ce *bonus* (nom du *bonus*, gain de *points*, son coût etc). À vous de trouver comment présenter, pensez aux infobulles ou encore aux menus déroulants.

Au clic sur l'un de ces *bonus*, une animation doit se déclencher, afin de fournir à l'utilisateur un retour qui valide la prise en compte de son achat de *bonus*.

Il ne doit pas être possible d'acheter un *bonus* si le nombre de *points* est insuffisant. Deux solutions s'offrent à vous : soit vous bloquez le clic sur le bouton d'achat du *bonus*, soit vous cachez les *bonus* jusqu'à ce que le joueur arrive à un certain nombre de *points*. Vous pouvez aussi choisir d'utiliser ces deux solutions de façon complémentaire.

Quand l'utilisateur achète un *bonus*, cela lui permet d'augmenter le nombre de *points* générés à chaque clic, ou le nombre de *points* générés automatiquement toutes les secondes, ou les deux.

Remarque : si le *bonus A* rapporte +10 *points* à chaque clic, et si l'utilisateur possède 3 fois le *bonus A*, il gagne  $10 * 3 \Rightarrow 30$  *points* de plus à chaque clic.

Liste des *bonus* à mettre en place :

	Gain de points	Coût du bonus
Sur les <i>points</i> au clic	+10 <i>points</i>	100 <i>points</i> de base puis x1,5 à chaque achat
Sur les <i>points</i> automatiques	+10 <i>points</i>	100 <i>points</i> de base puis x1,5 à chaque achat
Sur les <i>points</i> au clic et automatiques	+1 000 <i>points</i>	10 000 <i>points</i> de base puis x3 à chaque achat

Exemple avec le *bonus* sur les *points* au clic :

- Le joueur achète le *bonus* pour la 1<sup>ère</sup> fois, il devra dépenser 100 *points*.
- Si il souhaite l'acheter une 2<sup>ème</sup> fois, il devra dépenser  $100 \times 1,5 \Rightarrow 150$  *points*.
- Si il souhaite l'acheter une 3<sup>ème</sup> fois, il devra dépenser  $150 \times 1,5 \Rightarrow 225$  *points*.
- Etc

## Les succès

Liste des *succès* à mettre en place :

- Palier des 10 *points* atteint ;
- Palier des 1 000 *points* atteint ;
- Palier des 100 000 *points* atteint ;
- 1 *succès* lorsque l'on achète un *bonus* pour la 1<sup>ère</sup> fois.

Les trophées des *succès* doivent apparaître sur l'interface du jeu.

Afin de détecter le déblocage d'un *succès* ou non, le plus simple est de faire cette détection lorsque vous faites gagner des *points* à votre utilisateur, ou lors de l'achat d'un *bonus*.

## Enregistrement du score

Afin de sauvegarder vos scores de façon centralisé et pouvoir comparer votre score avec celui de vos camarades, une API REST est mise à votre disposition, sur l'URL suivante :

<https://sae-301.azurewebsites.net>

Cette API ne requiert pas d'authentification pour être utilisée.

3 requêtes seront disponibles sur cette API afin de sauvegarder vos scores. À l'aide de celles-ci, vous devrez sauvegarder le score de votre utilisateur toutes les minutes. Pour ce faire vous devrez au préalable sauvegarder dans votre sessionStorage le username saisie par l'utilisateur sur la page d'accueil et ainsi, vous en servir durant la session de jeu de votre utilisateur pour envoyer son score à l'API toutes les minutes.

**Attention :** Les données de l'API seront remises à zéro le mardi 03/01/2023, afin que vous puissiez repartir d'une base saine pour préparer votre présentation orale.

### Requête POST (sauvegarde du score)

URL : <https://sae-301.azurewebsites.net/save-score.php>

#### Paramètres :

- username (doit contenir plus de 5 caractères, string)
- score (doit être supérieur à 0, number)
- force (non obligatoire, boolean)

Format de la requête : JSON

Format de la réponse : JSON

Informations complémentaires : Si vous avez déjà sauvegardé un score pour un username donné, l'API refusera de mettre à jour le score de l'utilisateur s'il est inférieur au score précédent. Toutefois vous pouvez forcer la sauvegarde en ajoutant le paramètre "force" avec une valeur "true" dans votre body.

### Requête GET (Récupération du score d'un utilisateur)

URL : <https://sae-301.azurewebsites.net/get-user-score.php>

#### Paramètres :

- username (en queryParams)

Format de la réponse : JSON

Information complémentaire : Si l'utilisateur est introuvable, l'API vous renverra une erreur 404.

## Requête GET (Récupération des 300 meilleurs scores)

URL : <https://sae-301.azurewebsites.net/get-leaderboard.php>

Paramètres : Aucun

Format de la réponse : JSON

Information complémentaire : Vous recevrez un tableau contenant les scores. Ces derniers, seront ordonnés par ordre décroissant.

## Responsive design

Votre page devra être responsive et devra supporter deux affichages :

- Un affichage standard pour une utilisation PC ;  
(largeur minimale : 720 px)
- Un affichage mobile.  
(largeur maximale : 576 px)

Comprenez que l'affichage entre 720 px et 576 px ne sera pas noté, c'est-à-dire que l'affichage entre ces deux valeurs peut être complètement cassé, ce n'est pas grave !

## Apparence alternative du *clicker*

À l'aide d'un élément cliquable (checkbox, bouton radio, bouton, select ou autre), au clic sur cet élément, le *cookie* doit changer d'apparence. Voyez cela comme un *skin*.

Vous devez en proposer 1 au minimum.

# Notation

Seront pris en compte :

- L'implémentation des fonctionnalités ;
- Le respect du cahier des charges ;
- Le respect des technologies utilisées ;
- La qualité du code ;
- L'architecture de la solution (séparation des fichiers dans des dossiers différents) ;
- L'ergonomie ;
- La prise en compte de l'expérience utilisateur ;
- La qualité des animations et leurs cohérences ;
- La qualité et la cohérence du thème de l'interface ;
- L'interactivité de votre maquette ;
- La mise en ligne sur o2switch.

# Livrables

Les livrables suivants sont à déposer sur eLearning.

Les archives doivent avoir le nom suivant :

[MMI-S3] Projet Nom1-Nom2-Nom3 .(zip/rar/tar/tar.gz)

Vous pouvez rendre les livrables plus tôt. Tant que la date de rendu n'est pas passée, vous pourrez mettre à jour votre rendu.

**Retard** : Chaque jour de retard commencé sera pénalisé par un retrait de 3 points.

Tout rendu après le jeudi 05/01/2023 sera ignoré.

Les livrables :

## 1) Archive contenant la maquette du projet

- Le fichier de la maquette
  - Adobe XD : au format Adobe XD (.xd)
  - Figma : au format FIG (.fig)
- Le PDF de la maquette (.pdf)

**Date de rendu** : dimanche 04/12/2022 à 23h59

## 2) Archive contenant votre *cookie clicker*

L'ensemble de votre projet est à déposer sous forme d'archive (ZIP, RAR, 7Zn TAR, TAR.GZ).

Il devra contenir :

- Dans un dossier “\_maquette” : la version finale de la maquette
  - Au format .xd ou .fig
  - Et au format .pdf
- Les fichiers HTML
- Les fichiers CSS
- Le(s) fichier(s) JS
- Vos ressources (images)
- JQuery (si utilisé)
- Un fichier README.txt qui contiendra une petite description de votre projet, ce que vous auriez développé en plus, les manipulations qui pourraient être cachées dans votre application, le navigateur que l'on doit utiliser pour ouvrir les pages, et toutes informations complémentaires que vous souhaiteriez porter à notre connaissance.

**Date de rendu** : lundi 02/01/2023 à 23h59



## Présentation orale

Vous devrez nous présenter votre projet et sa maquette, nous expliquer vos choix, les éléments que vous aimeriez améliorer etc. Durant cette présentation, vous devrez réaliser une démonstration de votre *cookie clicker*.

**Date de l'oral** : vendredi 06/01/2023

**Temps de présentation** : 15 minutes + 5 minutes de questions.

## Bonus

Voici une liste non exhaustive des fonctionnalités **BONUS** que vous pourriez implémenter :

### Malus périodiques (exemple)

Des malus qui s'appliquent à la génération de *points* (automatique ou au clic). Ces malus peuvent apparaître de façon aléatoire.

### Sauvegarde et Chargement de la partie (exemple)

Votre jeu sauvegarde automatiquement l'état de la partie de l'utilisateur, son nombre de *bonus*, de *points*, les *succès* débloqués dans le local storage.

Lorsque votre utilisateur retourne sur le jeu, sa partie est rechargée automatiquement et le jeu reprend.

### Nouvelle partie (exemple)

Un bouton permettant de tout réinitialiser et recommencer une partie de zéro.

### Rendre certaines apparences déblocables contre des *points* (exemple)

Certaines apparences du *clicher* ou de l'interface pourraient être déblocables contre un certain nombre de *points*.

### D'autres *succès* / « Haut-Faits » / « Trophées » (exemple)

Via certaines actions, l'utilisateur débloque d'autres *succès* (certain nombre de *points*, nombre de *bonus* achetés, temps passé sur le jeu, nombre de *points* perdus, etc).

Voyez ces propositions comme des petits plus, vous n'êtes pas obligés de les implémenter. Si vous les implémentez, n'oubliez pas de les signaler dans votre **README.txt**, pour qu'ils soient pris en compte dans la notation.