

Cahier des charges

Description du projet :

Nous étudions le problème de la gratification différée dans le domaine de l'informatique. Nous voulons trouver pourquoi les utilisateurs ne font pas l'effort d'apprendre les raccourcis ou des logiciels plus performant pour interagir avec un système interactif. Nous voulons savoir pourquoi ils ne prennent pas le temps d'apprendre pour être plus efficace. Pour cela nous allons mener une expérience et créer une interface web qui simule le fait d'apprendre, puis nous le ferons tester à l'utilisateur pour obtenir des résultats que nous analyserons.

Description de l'expérience:

Dans l'expérience, il y a des formes géométriques, le but, à chaque niveau, est de cliquer le plus rapidement possible sur toutes les formes attendues. Pour simuler l'apprentissage, nous permettons à l'utilisateur de libérer des verrous pour chaque forme. Dès que tous les verrous de la forme sont déverrouillés, l'utilisateur n'aura besoin que d'un seul clic sur la forme pour tous les sélectionner. Pour simuler le temps et l'effort d'apprentissage, nous mettons en place une barre qu'il faudrait agiter pour déverrouiller un seul verrou.

Définition des mots scientifiques dans la description

La gratification différée est une notion d'économie, c'est la capacité de résister à une tentation d'obtenir une récompense immédiate au profit d'une récompense plus tard . Par exemple, quelqu'un aurait tendance à préférer 10 euros aujourd'hui plutôt que 15 euros demain.

Un système interactif est une application informatique qui prend en compte, au cours de son exécution, des informations communiquées par le ou les utilisateurs du système, et qui produit, au cours de son exécution, une représentation perceptible de son état interne.

Historique du projet:

Nous avons repris le code, de nos camarades d'il y a 3 ans, qui est une application hébergée sur Glitch codée en langage JavaScript et Node Js.

Objectif envisagé:

Le but principal est de créer une plateforme simple et malléable pour pouvoir effectuer les expériences, ensuite nous ferons passer des utilisateurs pour visualiser et analyser leurs comportements.

Architecture générale:

- une API Web codé sous NodeJs
- avec pour entrée un fichier de configuration (taux d'apparence des formes, nombres de verrous, temps d'apprentissage, nombre de niveau)
- une interface pour les utilisateurs pour l'expérience
- une interface de tutoriel pour l'utilisateur
- une interface de configuration manuelle
- en sortie une page où l'on peut télécharger un fichier log des résultats

Tâches et organisation des tâches

Tâches principales

- Formater le code (fini d'ici fin février au plus tard)
 - Front end (html, css, design) : Kate
 - Back end (JavaScript, Node Js, fonctionnalités principales) : Eric
- Ajout de fonctionnalités (dans la foulée du codage)
 - L'utilisateur peut ou non avoir un pre-aperçu des formes qu'il va avoir à sélectionner (Savoir si cette fonctionnalité sera utilisée ou pas)
 - Une possibilité d'échec dans l'ouverture du cadenas (simule le fait de ne pas avoir compris la fonctionnalité)
 - Mettre une probabilité qu'un cadenas soit de nouveau verrouillé (oubli des raccourcis)
- Un cycle de l'expérience (si possible le faire 3 à 4 fois une seule fois prend 2 à 3 semaines maximum)
 - Définir des paramètres à évaluer
 - Faire tester aux utilisateurs dès que nous le pouvons avec si possible plusieurs groupes d'utilisateurs. (1 ou 2 semaine)
 - Visualisation et analyse des résultats (1 semaine)
 1. Traiter les données

2. Visualiser les données
 - En fonction des résultats, modifier les différents paramètres de l'application (valeur du temps d'apprentissage, variantes)

Tâches Optionnelle

faire des variantes qui dépendent des résultats

générer un rapport pour les participants

ajouter questionnaires et âges

Diagramme de Gantt (temporaire)

