

JESSICA CHAN ET SEBASTIAN PEREZ

420-C61-IN Projet Synthèse

Groupe 00001

Générateur d'art dynamique

Projet Synthèse en techniques de l'informatique

**RAPPORT MI-MANDAT**

Travail présenté à

M Jean-Christophe DEMERS

Cégep du Vieux Montréal

Mercredi, 23 novembre 2022

## Jessica Chan

**Description de la fonctionnalité réalisée :** J'ai réalisé la logique de connexion du côté frontend en faisant des appels au serveur pour connaître les états de connexion du membre ou du visiteur. Si un nouveau visiteur souhaite s'inscrire sur notre plateforme, il passe par la page d'inscription pour se créer un compte. Une inscription inscrit les informations de l'utilisateur à notre base de données hébergée avec AWS. Par la suite, une fois le compte créé, il s'authentifie à la page de connexion. Une authentification valide signifie que le compte a pu être recueilli de la BD à la suite d'une requête à la base de données envoyée par le UserDAO. Elle permet au membre d'accéder à deux nouvelles pages auparavant cachées pour le visiteur : la page du profil de l'utilisateur et la page des partages. Les liens vers ces deux pages apparaissent sur la barre de navigation à la suite de la connexion. Une personne non authentifiée ne pourra pas accéder à ces pages, même en entrant le lien dans la barre d'adresse URL. Toute tentative renverra le visiteur à la page d'accueil. Si l'authentification n'est pas valide, le visiteur est notifié par des alertes d'erreur apparaissant au-dessus des champs du formulaire, ainsi que par l'effet de secoue du bouton de connexion.

**Plus grand défi :** Le plus grand défi venant avec cette fonctionnalité est de se servir de React, un environnement nouveau pour l'équipe, pour mettre en place le patron de conception Template method, plus précisément le CommonAction de Frédéric Thériault. C'est avec ce design pattern que nous nous basions pour faire la logique de connexion. Cependant, à la suite d'un mauvais jugement sur le fonctionnement de React, il a été nécessaire pour l'équipe de faire une migration du projet vers une application React plutôt que de l'importer comme dépendance dans le HTML. Étant donné que React nécessite une séparation claire entre ce qui

est du côté serveur et ce qui est du côté client, on a dû transformer les pages php de notre frontend original en composantes de React, donc des pages javascript. Conséquemment, il a fallu une réévaluation du fonctionnement de la connexion étant donné que le frontend ne peut plus parler directement au backend, mais doit plutôt se servir des fonctions fetchs de javascript.

### **Emplacement de la fonctionnalité:**

*ProjetSynthese\_A2022/dev/projet-synthese/src/pages/components/Form/FormConnexion.js*

Lignes 16 à 41

*ProjetSynthese\_A2022/dev/projet-synthese/src/pages/components/Form/FormRegister.js*

Lignes 19 à 32

*ProjetSynthese\_A2022/dev/projet-synthese/src/pages/components/Navigation.js*

Lignes 5 à 22

**Développement important à venir :** La prochaine étape importante est de permettre à l'utilisateur, une fois authentifié, de pouvoir télécharger ou de partager les animations de page d'animation pour qu'elles soient sauvegardées localement ou partagées sur leur page de profil et la page de partage.

**Défi le plus significatif dudit développement:** Le défi consiste à comprendre le fonctionnement du téléchargement de GIFs à partir de la librairie P5.js afin de juger la faisabilité d'insérer ce dernier dans notre base de données Postgresql, ainsi que de comprendre comment gérer un fichier GIF. De plus, étant donné que les boutons de partage et de téléchargement se retrouvent dans une composante de React en dehors du code des animations dans lequel nous avons

accès à la librairie P5.js, nous devons trouver une manière qu'au déclenchement de ces boutons, on appelle la fonction de téléchargement de ce dernier.

**Estimation d'avancement total : 60%**

## Sebastian Perez

**Description de la fonctionnalité réalisée :** J'ai fait une animation qui consiste à trouver le milieu de l'écran et le définir comme point d'origine. Suite à cela, une boucle de 0 à  $2\pi$  par intervalles de  $\pi/10$  est créée dans laquelle les variables  $x$  et  $y$  sont définies respectivement comme étant le cos et le sin d'un angle de rotation. Ces variables sont ensuite multipliées par un rayon. Alors, j'additionne  $\pm x$  et  $y$  à l'origine afin de créer deux points symétriques à ce dernier. Ces deux points sont ensuite liés par une ligne tout en tournant autour de l'origine. Ce processus se répète à l'intérieur de la boucle et on se retrouve avec l'animation projetée à l'écran.

**Plus grand défi :** L'aspect mathématique de l'animation a été le plus grand défi. À certains moments, il m'était difficile de comprendre la logique de ce qui se passait à l'écran et il m'est même arrivé de simplement ne pas comprendre le comportement de mon code. Éventuellement, la logique de l'animation est devenue beaucoup plus claire et j'ai pu comprendre son fonctionnement.

### **Emplacement de la fonctionnalité:**

*ProjetSynthese\_A2022/dev/projet-synthese/src/pages/components/generativeAlgorithm/LineRotationAlgorithm.js*

Lignes : 18 à 130

**Développement important à venir :** Une des animations à venir consiste d'un triangle qui se déplace à l'écran pendant qu'un cercle se déplace sur les extrémités de l'écran à l'aide des courbes de Bézier. Lors de la collision de ces éléments, ils seront poussés dans des directions opposées.

**Défi le plus significatif dudit développement:** Coder et utiliser les structures de données (file et stack) utilisées par l'animation ainsi que la gestion de collision.

**Estimation d'avancement total :** 60%