YADUSTOCK

Projet Tutoré S3

Marion Tallandier

Youssef El Kadi

Loïs Pazola

Corentin Fautre

Samuel Lachaud

03 octobre 2020

# Sommaire

I - Introduction au sujet……………………………………...p ?

II - Etude des jeux existants…………………………………p ?

III - Les technologies envisagées

1. Présentation des technologies………………………...……p ?
2. Notre choix………………………………………...……p ?

IV - Conception

1. Diagramme de classe……………………………………...p ?
2. Diagramme de cas d’utilisation………..……………………p ?
3. Diagrammes de séquence……………………………….…p ?

V - Maquettes…………………………………………..……p ?

# Introduction au sujet

Le projet Yadustock porte sur la création d’un jeu de gestion d’une entreprise. Nous avons choisi que l’entreprise sera spécialisée dans la vente de jeux de cartes afin d’offrir au joueur un large éventail de produits pour lui permettre une certaine personnalisation.

Afin de profiter du jeu le joueur aura besoin d’une interface graphique lui permettant de gérant les parties les plus importantes d’une entreprise. L’interface doit être claire et intuitive pour que l’utilisateur puisse prendre en main facilement le mécanisme du jeu.

Nos principaux objectifs se basent sur la bonne expérience du joueur, nous allons donc nous focaliser sur la simplification des actions menées par celui-ci en offrant une interface claire et agréable à utiliser. Nos seconds objectifs seront d’offrir quelques possibilités pour pimenter l’expérience du joueur tel que des parts de marché partagées entre lui et différents concurrents ou encore un système de banque pour pouvoir emprunter.

# Etude des jeux existants

# Les technologies envisagées

## Présentation des technologies

|  |  |
| --- | --- |
| Java + JavaFX | |
| Points positifs | Points négatifs |
| * Accessible à l’entièreté de l’équipe * JavaFX est facile à prendre en main * Programmes multi-plateformes * JavaFX possède des fonctions graphiques | * Java est un langage très lourd |

Pour notre projet un large choix de technologies s’offrait à nous, nous avons donc fait un tableau comparatif pour nous aider à choisir la meilleure pour remplir nos principaux objectifs. Notre tableau nous a donc permis de tirer ces informations :

|  |  |
| --- | --- |
| Python + PyQT | |
| Points positifs | Points négatifs |
| * Facilement lisible * PyQT possède des fonctions graphiques * Rapide est léger * Possibilité de générer l’IHM depuis un logiciel tiers (QT) | * Langage non-maitrisé par l’équipe et nécessiterais donc un temps d’apprentissage. * Syntaxe difficile à utiliser * Non multi-plateforme * Difficile de générer l’application finie |

**Langages de programmation**

|  |  |
| --- | --- |
| C# + .NET | |
| Points positifs | Points négatifs |
| * Langage récent et grandement documenté * La création d’IHM est très intuitive et très complète * Gagne beaucoup de temps à la programmation | * Langage en apprentissage pour chacun des membres de l’équipe |

**IDE**

|  |  |
| --- | --- |
| Visual Studio | |
| Points positifs | Points négatifs |
| * IDE Très complet proposant beaucoup de fonctionnalités | * IDE très lourde à utiliser, très fournis. * Met du temps à compiler |

|  |  |
| --- | --- |
| Netbeans | |
| Points positifs | Points négatifs |
| * Implémente bien Java et JavaFX * Très facile de faire le produit final (.jar) | * Met beaucoup de temps à la compilation les gros projets. |

|  |  |
| --- | --- |
| QT Creator | |
| Points positifs | Points négatifs |
| * Générer facilement le code de l’IHM * Incorporation facile du CSS * Large gamme de paramètres pour configurer l’IHM | * Code généré en XML, transformation en format PyQT difficile. * Lourd à utiliser * Difficultés de travail dans différentes versions de QT |

**Edition d’image**

|  |  |
| --- | --- |
| Gimp | |
| Points positifs | Points négatifs |
| * Edition poussé possible | * Non intuitif |

|  |  |
| --- | --- |
| PhotoFiltre | |
| Points positifs | Points négatifs |
| * Intuitif, facile d'utilisation | * Ne permet pas d'édition poussé |

**Logiciels de partage de fichiers**

|  |  |
| --- | --- |
| SVN (TORTOISE) | |
| Points positifs | Points négatifs |
| * Intégration avec l’environnement Windows * Privé * Hébergé à l’IUT | * Peu pratique d’utilisation * Peu intuitif |

|  |  |
| --- | --- |
| Dépôt Git (GitHub) | |
| Points positifs | Points négatifs |
| * Très permissif * Bien organisé * Facile d’utilisation * Géré depuis un client Git et non depuis les dossiers Windows * Disponible depuis un navigateur web * Privé | * Stocké sur des serveurs non possédés par l’iut. * L’organisation des fichiers et les commit sont plus longs à effectuer |

## Nos choix

À la suite de l’analyse du tableau nous avons pris les décisions ci-dessous :

Nous utiliserons le langage C#/.NET pour la réalisation du produit final. Nos besoins étant énormément basé sur l’IHM ce langage est le plus approprié. De plus cela nous permettra d’approfondir notre apprentissage actuel.

Nous utiliserons Visual Studio pour réaliser le produit car il s’agît du meilleur IDE pour C#.

Nous utiliserons PhotoFiltre pour l’édition d’image car il s’agît du logiciel le plus simple d’utilisation. De plus certains membres de l’équipe connaissent déjà très bien celui-ci.

Nous utilisons un dépôt git qui a déjà été crée pour la réalisation de ce dossier et la suite du projet.

# Conception

## Diagramme de classe

## Diagramme de cas d’utilisation

## Diagrammes de séquence

# Maquettes