

Deuxième année licence mention informatique

PROJET

SIMULATION DU JEU MONOPOLY

Equipe :

- Anis BOUDIAF
- Ismaïne MESSAOUD
- Lamine CHELAGHA
- Kamelia TERKMANI

Encadré par :

M. Flavien BREUVART

ANNEE UNIVERSITAIRE : 2020/2021

Remerciements

Nous tenons à remercier tout d'abord notre cher encadreur M. **Flavien BREUVART** pour l'aide qu'il nous a apporté tout au long du déroulement du projet en nous orientant et en nous assistant ainsi la bonne tenue de ce dernier.

Nous tenons également à remercier toute personne ayant contribué, de près ou de loin, à la réalisation du projet.

Introduction

Le **Monopoly** est un jeu de société américain édité par Hasbro. Le but du jeu consiste à ruiner ses concurrents par des opérations immobilières. Il symbolise les aspects apparents et spectaculaires du capitalisme, les fortunes se faisant et se défaisant au fil des coups de dés. Ce jeu de société est mondialement connu, et il en existe de multiples versions. Dans ce contexte, nous sommes appelés à concevoir, développer et mettre en place une application web qui code un jeu de société, inspiré du Monopoly.

C'est un jeu de plateau, au tour par tour, conçu pour huit joueurs. Le plateau est rectangulaire, et des cases sont présentes sur toute la bordure du plateau. Chaque case correspond à une propriété, que les joueurs peuvent acheter ou vendre sous certaines conditions (sauf une : la case "départ"). Chaque propriété a un prix d'achat, une couleur, et un loyer.

Les joueurs ont chacun un pion, qu'ils déplacent sur les cases du plateau et qui commence la case "départ", et ont chacun une certaine somme d'argent, non nulle au début de la partie de telle manière à permettre aux joueurs d'acheter rapidement quelques propriétés.

Lors de son tour, un joueur lance deux dés et déplace son pion d'un nombre de cases égal à la somme des dés sur le plateau dans le sens horaire.

Lorsqu'un pion s'arrête sur une case qui correspond à une propriété qui n'est pas encore achetée, le joueur qui possède le pion peut choisir de l'acheter, s'il a assez d'argent. De plus, si le pion s'arrête sur une case qui correspond à une propriété possédée par un autre joueur, le joueur qui possède le pion doit lui payer le loyer de la propriété et peut également demander au propriétaire de la lui vendre. Si un pion passe par la case "départ", le joueur qui possède le pion reçoit une certaine somme d'argent. Un joueur est éliminé de la partie dès qu'il est en faillite, et lorsque tous les joueurs ont fait faillite sauf un, le joueur restant a gagné.

Planification

Introduction

Comme chaque projet informatique, notre projet nécessite une planification bien précise, dans laquelle nous allons définir les besoins des utilisateurs et des clients, et en déduire donc les principaux objectifs de notre application.

Ces questions sont, à ce niveau, implicitement posées : Quelle méthodologie faut-il suivre pour concrétiser notre travail ? Comment pouvons-nous gérer notre projet ?

Dans ce chapitre, on s'intéresse à apporter des réponses précises avec des détails aux questions déjà posées en clarifiant leurs aspects techniques.

D'abord nous allons expliquer notre méthodologie de travail. Ensuite nous passerons aux outils nécessaires pour la réalisation de notre projet. Nous allons également préciser les utilisateurs auxquels notre application sera destinée et les cas de son utilisation.

Méthodologie de travail

Un travail planifié présente toujours des avantages par rapport à un travail anarchique. Cette planification permet d'avoir un travail organisé ce qui nous fait gagner beaucoup de temps parce que chaque tâche trouve un espace limité et nous évite de nous étaler dans des détails inutiles. D'autre part ceci nous évite de perdre du temps à rêver avant de décider de ce que l'on va entamer. Enfin un travail planifié paraît plus attractif et facilite les différentes étapes du projet. A cet effet nous avons décidé de choisir l'adaptation de trois étapes essentielles qui permettront de structurer la mise en œuvre de notre application, comme l'indique le schéma suivant :

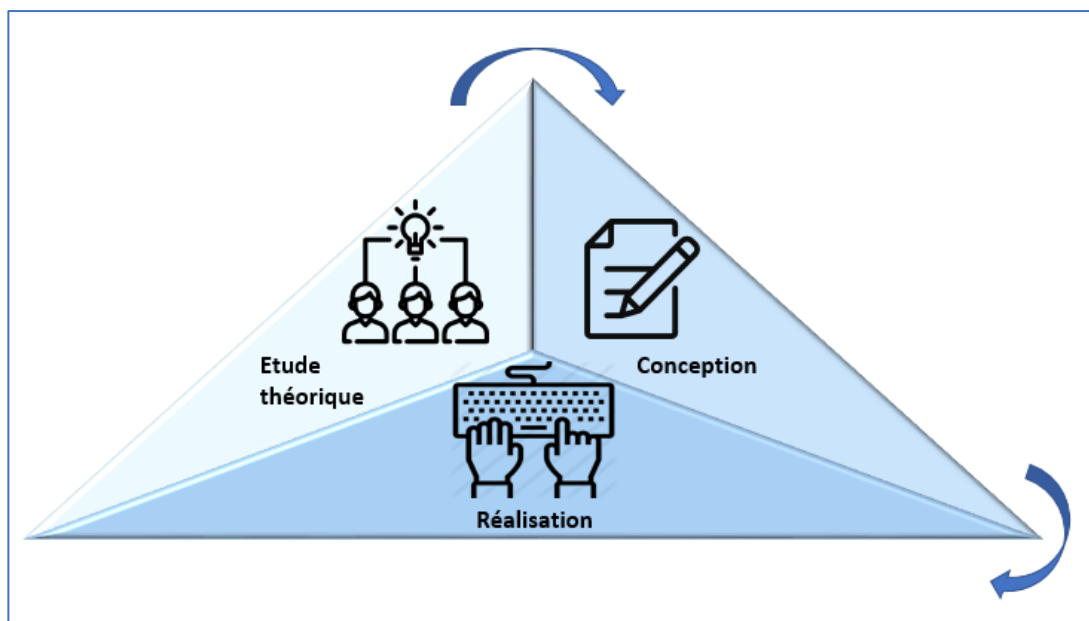


Figure 1: Etapes de progression

Notre plan s'étalera sur trois phases essentielles :

Première phase (Recherche et documentation)

Cette première phase a consisté en la prise de connaissance du sujet en consultant le cahier de charge, en nous approfondissant dans la lecture de son contenu et la compréhension de ses différents

articles. En outre nous nous sommes profondément documentés sur les différents outils utilisés (langages de programmation, logiciels ...etc.).

Deuxième phase (Conception)

Après avoir pris connaissance de la documentation liée au sujet, langage de programmation et logiciels utilisés, et après avoir fini l'étude théorique nous sommes passé à la conception, qui a consisté en la définition des différentes classes nécessaires, les méthodes à développer et le prototype de l'interface graphique.

Troisième phase (Réalisation)

Sur ce point nous avons commencé à concrétiser ce qui a précédé. Nous avons donc mis en pratique ce qui a été défini lors de la phase de conception, en commençant par la programmation, la création des classes de bases, leurs implémentations et enfin la mise en œuvre de l'interface graphique avec les changements adéquats selon les cas.

Outil de gestion de projet

Pour mieux nous organiser et gérer les modifications et mises à jour du code source nous avons opté pour le service web **GitHub** qui donne la possibilité aux différents membres de voir les changements faits dans le code par les autres et permet de récupérer les versions précédentes en cas d'erreur.

Les cas d'utilisation

Notre application sera destinée à tout type de personne.

Répartition des tâches

Afin de mieux s'organiser on a réparti notre projet sur plusieurs tâches :

- 1- Conception et implémentation des modules réalisés par Kamelia TERKMANI.
- 2- Réalisation du front (design et interface) par Ismaïne MESSAOUD.
- 3- Conception et implémentation des modèles de BDD et du Backend par Anisdhiyaeddine Boudiaf.
- 4- Intégration des différentes fonctionnalités en front par Lamine CHELAGHA.

Conclusion

Dans ce chapitre, nous avons présenté le plan général du travail, les outils de gestion du projet, et enfin les cas d'utilisation auxquels nous avons conçu notre application. Nous pouvons ainsi entamer le chapitre suivant.

Conception

Introduction

Après avoir déterminé notre plan de travail, la méthodologie à suivre ainsi que les cas pour lesquels notre application sera dédiée, nous pouvons passer à la conception détaillée de l'application qui comprend les différents diagrammes des classes.

Diagramme des classes

Pour notre application, nous aurons les classes :

- Case
- Départ
- Prison
- Cartes
- Propriété
- Couleur
- Joueur
- Pion
- Banque
- Monopoly

Définition des classes

1- Classe Case :

Classe	Rôle
Public class Case	
Attributs	Rôle
Private int nom	Le nom de la case
Méthodes	Rôle
Public void equals (Case)	Redéfinition de equals

2- Classe Départ :

Classe	Rôle
Public class Départ extends Case	
Attributs	Rôle
Private Nom	
Méthodes	Rôle
Public equals (depart)	

3- Classe Prison

Classe	Rôle
Public class Prison extends Case	
Attributs	Rôle
Private Nom	
Méthodes	Rôle
Public equals (prison)	

4- Classe propriété :

Classe	Rôle
Public class Propriété extends Case	
Attributs	Rôle
Private nom	Nom de la case
Private prixBase	Prix de la propriété
Méthodes	Rôle
Public equals ()	Egalité de deux cases

5- Classe Terrain :

Classe	Rôle
Public class Terrain extends Propriete	
Attributs	Rôle
Private nomcase	Nom de la case
Private prixBase	Prix de la propriété
Private loyerNu	Loyer sans maisons ni hôtels
Private loyer1M	Loyer avec une maison
Private loyer2M	Loyer avec 2 maisons
Private loyer3M	Loyer avec 3 maisons
Private loyer4M	Loyer avec 4 maisons
Private loyerH	Loyer avec un hôtel
Méthodes	Rôle
Public equals ()	Egalité de deux terrains

6- Classe compagnie :

Classe	Rôle
Public class Compagnie extends propriete	
Attributs	Rôle
Private nom	Nom de la case
Private prixBase	Prix de la compagnie
Méthodes	Rôle
Public equals ()	Egalité de deux compagnies

7- Classe gare :

Classe	Rôle
Public class Gare extends propriete	
Attributs	Rôle
Private nom	Nom de la case
Private prixBase	Prix de la gare
Private loyerG	Loyer gare
Méthodes	Rôle
Public equals ()	Egalité de deux gare

8- Classe Carte

Classe	
Public class Carte extends Case	
Attributs	Rôle

Private nom	
Private texte	Enoncé de la carte
Private type	Type de la carte
Méthodes	Rôle

9- Classe Banque :

Classe	Rôle
Public class Banque	
Attributs	Rôle
Private nbmaisons	Nombre de maisons
Private nbhotels	Nombre d'hôtels
Méthodes	Rôle
Public acheterM ()	Achat d'une maison
Public acheterH ()	Achat d'un hotel

10- Classe Couleur

Classe	
Public class Couleur	
Attributs	Rôle
Private couleur	Texte de la couleur
Private prixmaison	Prix d'une maison
Private prixhotel	Prix d'un hotel
Méthodes	Rôle

11- Classe Joueur

Classe	Rôle
Public class Joueur	
Attributs	Rôle
Private nom	Nom du joueur
Private couleur	Couleur du joueur
Private enPrison	Booléen vrai si le joueur est en Prison
Private pion	Pion du joueur
Private nbDouble	Nombre de doubles
Private bloque	Le joueur est bloqué, il doit se débloquer pour que le jeu continue
Private nbToursP	Nombre de tours en prison
Private defaite	Défaite du joueur
Private cartesSortiesPrison []	Cartes de sorties de prison que le joueur possède
Private montant	Solde du joueur
Proprietes[]	Propriétés du joueur
Méthodes	Rôle
Public choisir_pion (pion)	Choisir un pion au début du jeu
Public ajouter_carte_sortiePrison (carte)	Ajouter une carte de sortie de prison
Public utiliser_carte_sortiePrison ()	Utiliser une carte de sortie de prison
Public acheter (prop)	Acheter une propriété
Public acheter_maison ()	Acheter une maison
Public acheter_hotel ()	Acheter un hôtel
Public vendre_Propriete (prop,prix)	Vendre une propriété

Public nb_doubles ()	Vérifie si nombre des doubles est égal à 3 aller en prison
Public Payer(montant)	Payer un montant
Public Aller_prison()	Aller en prison
Public Sortir_prison (carte, payer)	Sortir de prison
Public Gagner(montant)	Gagner un montant
Public Perdre ()	Perdre
Public Modifier_nom(nom)	Modifier son nom
Public Modifier_couleur(couleur)	Modifier sa couleur
Public Payer_taxe()	Payer ses taxes

12- Classe Pion

Classe	
Public class Pion	
Attributs	Rôle
Private case_actuelle	Case ou se trouve le pion
Méthodes	Rôle

13- Classe Monopoly

Classe	Rôle
Public class Monopoly	
Attributs	Rôle
Liste_cases []	Liste des cases du plateau
Liste_joueurs []	Liste des joueurs de la partie
Listecartes_caissecommunauté []	Liste des cartes caisse communauté
Listecartes_chance []	Liste des cartes chance
Délançé	Vrai si les dés ont été lancés
Dé1	Nombre affiché après avoir lancé le Dé1
Dé2	Nombre affiché après avoir lancé le Dé2
Méthodes	Rôle
Public lancer_Dé ()	Lancer les dés
Public reinitialiser_Dé ()	Réinitialiser les dés
Public bouger_pion (joueur)	Bouger un pion
Public bonus_malus_case (joueur)	Traitement des cases
Public traitement_propriete (joueur, prop)	Traitement d'une propriété
Public Insertion_carte_chance ()	Insertion des cartes chance
Public Insertion_carte_caisse_communauté ()	Insertion des cartes caisse communauté
Public Bonus_malus_carteChance (joueur, carte)	Traite la carte chance
Public Bonus_malus_carteCC (joueur, carte)	Traite la carte caisse communauté
Public Tirer_carte ()	Permet de tirer une carte
Public calculer_loyerC ()	Calcule le loyer d'une compagnie
Public calculer_loyerT ()	Calcule le loyer d'un terrain
Public calculer_loyerG ()	Calcule le loyer d'une gare
Public construire_plateau ()	Construction du plateau avec initialisation de toutes les cases
Public jouer ()	Permet de jouer

Outils et langages utilisés

Le langage Orienté Objet JAVASCRIPT : Ses caractéristiques :

- ⇒ JavaScript est un langage “basé objets”:
- le programmeur peut “utiliser” des objets “JavaScript” pré-définis
 - il peut définir des objets simples (sans héritage)
 - tout élément conceptuel utilisé, affiché etc. à l’intérieur d’une page est représenté par un objet qui s’insère dans une structure de type DOM
 - d’autres objets existent.

Visual Studio & Blend for Visual Studio:

Visual Studio est un ensemble riche de logiciel de développement conçue par Microsoft qui permet de générer des applications web desktop et mobiles.

Visual Studio nous a permis d’implémenter le modèle et le Blend Visual Studio nous a permis de réa liser des interfaces graphiques bien animées.

Git -Outil de versionning :

Étant donné que nous sommes une équipe de 4 personnes à travailler sur le projet, la gestion du versionning devient une tâche de plus en plus pénible avec l’avancement du projet. De plus, il serait difficile de garder des versions de secours sans confusion entre elles. C’est pourquoi nous avons opté pour l’utilisation de GIT, en nous créant un dépôt sur GITHUB afin de gérer les versions de notre projet

Conclusion générale

Le Monopoly est un jeu de société sur parcours dont le but est, à travers l'achat et la vente de propriétés, de ruiner ses adversaires et ainsi parvenir au monopole. Dans ce cadre, s’installe notre projet qui porte sur la simulation de ce jeu de plateau.

Le point de départ de la réalisation de ce projet était une récolte des informations nécessaires ainsi que le choix de la méthodologie du travail. Par la suite, nous nous sommes intéressés à l’analyse de la problématique et la réalisation d’une solution globale (diagramme des classes). Le dernier volet de notre projet était la partie « étude technique et implémentation » qui a été consacrée à la présentation des outils du travail, les modules réalisés et les interfaces les plus significatives de notre application.

L’apport de ce travail a été d’une importance très considérable, en effet, il nous a permis d’approfondir nos connaissances dans le monde de développement. La réalisation d’un tel projet, nous a permis d’apprendre et de toucher du doigt une partie des divers aspects du métier de développeur et de celui du concepteur.

Bibliographie et références

- Apprendre à développer en Javascript « Livre du site OpenClassroom ».
- Base du langage MongoDB et des bases de données

Webographie

- www.openclassrooms.com
- www.docs.microsoft.com/fr-fr/
- www.stackoverflow.com
- www.developpez.net
- fr.wikipedia.org