**Équipe 24 :**

*Cyrille ADAM*

*Vincent CARON*

*Louis-François GARNIER*

RAPPORT Jeu Domi’nations

**Client : ISEP**

*Pour le 13 janvier 2019*

Table des matières

1. Contextualisation

[1. Présentation du projet 2](#_Toc535160321)

[2. Notre rôle dans le projet 2](#_Toc535160322)

3. Le périmètre du projet 3

1. Architecture du projet

[1. Unified Modeling Language (UML) : 4](#_Toc535160324)

[2. Présentation des classes 6](#_Toc535160325)

[3. … 4](#_Toc535160324)

[4. structure du projet 6](#_Toc535160325)

[5. Autre: 4](#_Toc535160324)

1. interface graphique

conclusion

Annexes

1. Contextualisation

# Présentation du projet

Le but du jeu Domi’Nations est simple : le joueur doit constituer un royaume de 5x5 cases autour de son château tout en faisant en sorte de gagner le maximum de points. Toutes les tuiles (2 cases) n’ont pas la même valeur puisque celle-ci est indiquée par le nombre de couronnes présentes sur chacune.

Ce projet nous amène à créer une interface graphique en Java pour Domi’Nations permettant à un utilisateur de saisir un nouveau plateau de jeu et de gérer de manière optimale les déplacements de ces pions.

# Notre rôle dans le projet

Notre équipe, constituée de 3 élèves ingénieurs ISEPiens a travaillé afin de répondre aux attentes de notre client, l’ISEP dans le cadre du module « Algorithme et Programmation ».

Celle-ci a développé en langage de programmation JAVA le jeu de plateau Domi’Nations dont les règles ont été fixées dans un cahier des charges précis.

Afin de répondre à vos attentes nous avons dans le cadre du « Document de spécifications » réfléchi à **l’architecture technique** de la solution, à l’organisation des données …



*Figure 1 – Équipe*

# Le périmètre du projet

Dans cette partie nous définissons le périmètre, autrement dit, ce que notre équipe a réalisé dans le cadre du projet demandé.

Il sera donc question pour l’équipe de prendre en charge la réalisation :

* De l’interface graphique (Plateau de jeu, dominos, interface Homme/machine)
* De préparation du jeu (nombre de joueurs, affectation des rois, des châteaux, des dominos)
* Du premier tour et des mécanismes de pioche (pioche, disposition, gestion face cachée/visible des dominos)
* Des tours de jeu (récupérer le domino, placer le domino selon les règles, sélectionner le domino
* Fin du jeu (calcul des points, arrêt du jeu)

Ci-dessous un extrait du cahier des charges correspondant aux fonctionnalités attendues :

*- Développement de l’interface graphique rendant le jeu jouable pour 2 à 4 joueurs humains en tour par tour.*

*- Récupération des dominos à partir d’un fichier qui vous sera fourni.*

*- Implémentation des fonctionnalités de base : initialisation du jeu, tours de jeu, calcul des points, et identification du/des vainqueur(s).*

*- Implémentation d’une IA capable de jouer selon les règles du jeu.*

*- Rédaction d’un document technique reprenant la conception de votre programme, les différents modèles de données utilisés, et un manuel d’utilisation de votre projet.*

1. Architecture du projet

# Unified Modeling Language (UML) :

Dans le cadre du projet nous avons réfléchi à comment faire pour que le système proposé soit simple de compréhension.

Nous avons donc opté pour l’utilisation d’un langage de modélisation simplifié afin de modéliser rapidement la conception de notre système.

L’utilisation des diagrammes UML (Unified Modelling Language) a donc été utilisée.

Nous avons tout d’abord schématisé les cas d’utilisation faisant interagir les différents acteurs du système.

Ce diagramme a pour fonction de répondre à une question simple, « A qui et à quoi va servir le système ? »

#### UML acteurs :

#### 

*Figure 2 : UML acteurs*

L’UML ci-dessus présente le rôle de chaque acteur du jeu.

Explications :

On peut retrouver ici deux services différents, les acteurs ainsi que les cas d’utilisation.

Les acteurs correspondent à des rôles, et sont des entités extérieures au système qui interagissent directement avec lui.

Deux acteurs sont modélisés ici le joueur et la machine.

Principe de fonctionnement :

Un acteur joueur a comme possibilité de placer le roi sur le plateau et de placer son domino (carte) autour de son château.

Un acteur machine distribue aléatoirement les rois aux joueurs, définit l’ordre, pioche un domino (carte) et les distribue aléatoirement sur le plateau de jeu en les triant par numéro croissant, compte les points.

#### UML de spécification fonctionnelle :



Ce diagramme a donc pour fonction de montrer la progression du jeu et le rôle de chaque acteur.

# Présentation des classes

Les classes suivantes permettent de représenter le jeu Domi’Nations dans la console :

#### DemiDomino

#### La **classe DemiDomino** représente

#### Domino

#### La **classe Domino** représente un domino, caractérisé par une valeur, séparée en deux terrains (Nord & Sud) représentés eux-mêmes par deux valeurs. Afin d’identifier si un roi est ou n’est pas présent sur celui-ci on affecte un booléen à cette condition.

#### Jeu

#### La **classe Jeu** représente

#### Joueurs

La **classe Joueurs** représente chaque joueur caractérisé par 3 valeurs : ordre actuel & ordre futur de jeu et son prénom.

Un comparateur est également présent au sein de cette classe afin de de comparer l’ordre de jeu de chaque joueur.

#### ListDomino

La **classe ListDomino** permet de récupérer le fichier CSV, par la méthode ReadFile, comportant les informations sur chacun des dominos. Ensuite, un mélange aléatoire est effectué par la méthode shuffle. Enfin, elle récupère les 3 premières cartes de la pile.

#### Main

#### La **classe Main** représente

#### Pioche

La **classe Pioche** permet de récupérer tous les dominos et de les mélanger de manière aléatoire. Elle permettait également d’afficher les numéros de chacun ainsi que d’afficher les trois premières cartes de la pile avec des données rentrées « à la main ».

#### Plateau

La **classe Plateau** représente le plateau de jeu, caractérisé par deux valeurs : nombre de lignes et nombre de colonnes.

#### Royaume

La **classe Royaume** représente

#### Terrain

La **classe Terrain** initialise les différents terrains (« mer », « champ », « bois », « mine », « montagne », « prairie »). Elle permet ensuite d’afficher ces noms en les retournant.

#### …

Version console graphique et serveur web est ce que certaines classes ne changent pas

#### Structure du projet

#### Répertoires et fichiers

Doc

Fichier CSV

#### Autre

* Si vous pensez que l'implémentation d'une fonctionnalité particulière nécessite d'être mise en avant, vous pouvez tout à fait le faire ; (compter les points)
* Si vous avez utilisé des bibliothèques pour des fonctionnalités précises, précisez-le et expliquez l'intérêt de ces bibliothèques.

1. Interface graphique

# Présentation de l’interface graphique

Nous avons souhaité soigner l’interface graphique afin que l’utilisateur ait envie d’interagir avec le jeu.

Voici un petit aperçu des caractéristiques graphiques que nous avons utilisées :

DOMINO :

*Img*

Ainsi, chaque terrain est symbolisé par une image et chaque domino a été conçu de manière à qu’on l’ait en deux orientations différentes : 0°/90°. De plus, chaque terrain peut comporter 0 à 3 couronnes.

PLATEAU DE JEU :

*Img*

Le terrain de jeu a été réalisé de la manière la plus réaliste possible. Comme dans un jeu de société, le joueur doit pouvoir manipuler et placer ses dominos de manière simple et intuitive.

Conclusion

L’objectif de notre projet est de répondre au mieux à votre appel d’offre en vous proposant des solutions nouvelles avant de passer à la réalisation de nos missions. Pour cela notre équipe a travaillé sur diverse activités présentés :

Les UML qui permettent de situer simplement le fonctionnement de notre système ainsi que de visualiser rapidement l’ensemble des activités de cette application.

Le schéma navigationnel qui permet de comprendre le fonctionnement et l’organisation de l’application en détaillant les différentes parties de cette dernière comme l’information du compte, la gestion d’abonnement, le suivi consommation, les aides et contrats.

Et finalement tout cela nous a amené à la présentation visuelle de cette application avec toutes ses spécificités ergonomiques réfléchies, notamment avec la présentation du logo de l’application et la modélisation de celle-ci.

Annexes