

Création du jeu de société « UNO® » en C#

Rebaptisé en « NUO »



SOVILLA Flavio
Flavio.sovilla@cpnv.ch

COSTA CABRAL Mauricio
Mauricio.costa-cabral@cpnv.ch



SI-CA2a

07 novembre 2019

Table des matières

1	Introduction	3
1.1	Cadre, description et motivation	3
1.2	Organisation	3
1.3	Objectifs.....	4
1.4	Planification initiale	4
2	Analyse.....	5
2.1	Maquette du jeu	5
2.2	Cas d'utilisation et scénarios	9
2.2.1	Jouer au NUO.....	10
2.2.2	Consulter les scores	11
2.2.3	Inscrire un pseudo	12
2.2.4	Consulter les règles du jeu	12
2.3	Modèle Conceptuel de Données.....	12
2.4	Stratégie de tests.....	13
2.5	Budget	13
3	Implémentation	13
3.1	Vue d'ensemble	13
3.2	Choix techniques	13
3.3	Points techniques spécifiques	14
3.3.1	Diagramme d'états – Jouer au NUO.....	14
3.3.2	Description de la distribution aléatoire des cartes.....	15
3.4	Livraisons	16
4	Tests.....	17
4.1	Tests effectués	17
4.2	Erreurs restantes	17
5	Conclusion.....	18
6	Annexes.....	21
6.1	Sources – Bibliographie.....	21
6.2	Journal de bord du projet.....	21

1 Introduction

1.1 Cadre, description et motivation

Dans le cadre du projet en binôme, qui se déroule lors du 1^{er} trimestre de la 2^{ème} année FPA d'informaticien, nous avons pour objectif de réaliser un projet concret de notre choix. Ce module a pour but de développer notre esprit d'équipe et de collaboration avec autrui ainsi que de mettre en pratique nos connaissances assimilées et/ou d'en acquérir davantage.

Comme le titre de cette documentation l'indique, nous avons décidé de reproduire le célèbre jeu de société « UNO® ». Ce choix est justifié par le fait que nous souhaitons avoir un minimum de challenge afin de mettre à profit nos connaissances et d'en développer de nouvelles.

Malheureusement, notre projet n'a pu être complètement achevé. Réaliser une application de cette envergure demande beaucoup de connaissances que nous n'avons pas forcément. Nous avons effectué énormément de recherches qui nous ont permis d'avancer passablement dans l'application. C'est pourquoi, nous sommes persuadés que nous aurions réussi, si le temps ne nous avait pas manqué.

Notre programme étant basé principalement sur le jeu original UNO®, nous avons renommé notre version du jeu en « NUO » (*droit d'auteur*). Cependant, pour une question de compréhension de la documentation, nous nous permettons d'utiliser le terme original.

Comme mentionné ci-dessus, nous n'avons pas pu terminer notre programme, de ce fait, les images qui sont dans le jeu sont celles du UNO®, nous n'avons pas eu le temps de faire les images nous-mêmes et de les remplacer.

1.2 Organisation

Chef de projet : VIRET Loic, loic.viret@cpnv.ch

Cliente : ANDOLFATTO Frédérique, frédérique.andolfatto@cpnv.ch

	Elève	Elève
Nom, Prénom	SOVILLA Flavio	COSTA-CABRAL Mauricio
Email	Flavio.sovilla@cpnv.ch	Mauricio.costa-cabral@cpnv.ch
Téléphone	+ 41 79 296 43 39	+41 76 449 18 31
Partie administration	X	X

Partie
Développement

X

X

1.3 Objectifs

Nous souhaitons créer une version du célèbre jeu de société « UNO® », prévue pour un joueur contre des intelligences artificielles (IA), en langage de programmation orientée objet C#, en mode graphique, d'ici la semaine d'examen (COM) du CPNV qui aura lieu du 11 au 15 novembre 2019

1.4 Planification initiale

La planification initiale de notre projet emploie la méthode « Agile », qui permet d'impliquer davantage la cliente dans le développement du projet. De cette manière, nous serons plus réactifs dans les besoins de cette dernière.

<div> <div>6 Open</div> <div>0 Closed</div> </div> <div>Sort</div>	
Planning général Updated 43 minutes ago	Sprint 1 - Finir le cahier des charges, use cases et scénarios Sprint 2 - Coder les menus et le plateau du jeu Sprint 3 - Créer la base de données et les IA Sprint 4 - Finaliser les tests et créer les règles optionnelles
Sprint 1 (28.08.19) Updated 32 minutes ago	<ul style="list-style-type: none"> Créer les cahier des charges (parties 3, 4 et 6) Créer les use cases et scénarios Planifier le tout sur Github (Issues et Projects avec basic kanban)
Sprint 2 (12.09.19) Updated 1 minute ago	<ul style="list-style-type: none"> Créer une page avec les règles du jeu Créer le menu avec les différentes options Créer le menu d'entrée graphique Créer un MCD Mettre à jour la documentation
Sprint 3 (26.09.19) Updated now	<ul style="list-style-type: none"> Créer la plateforme du jeu Créer le choix du lancement du jeu avec 1, 2 ou 3 IA Créer un tableau avec les 10 meilleures scores Mettre à jour la documentation
Sprint 4 (10.10.19) Updated 6 minutes ago	<ul style="list-style-type: none"> Créer une base de données pour les scores Créer une base de données pour les cartes Créer l'IA Mettre à jour la documentation
Sprint 5 (07.11.19) Updated 6 minutes ago	<ul style="list-style-type: none"> Créer, si le temps le permet, des règles optionnelles Tester le jeu final Mettre à jour la documentation

2 Analyse

Cette partie est consacrée à l'élaboration de la maquette de notre jeu ainsi que de nos cas d'utilisation et scénarios. Ces derniers ont été validé par notre chef de projet ainsi que par la cliente.

2.1 Maquette du jeu

Page d'accueil avec demande de pseudo (Lancement de l'application)

NUO - Version 1.0

NUO

LES CREATEURS

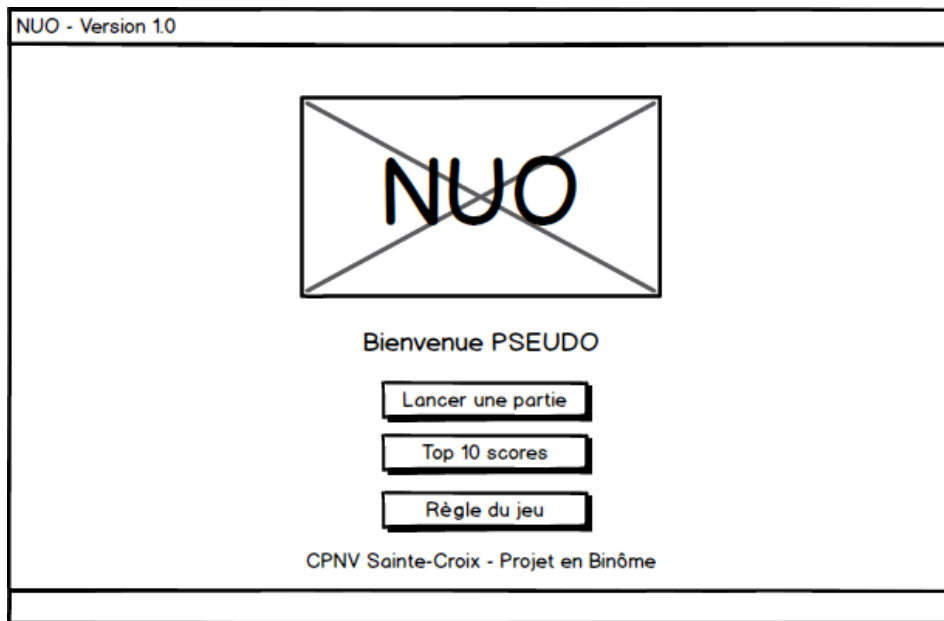
Flavio SOVILLA
Mauricio COSTA CABRAL

CPNV

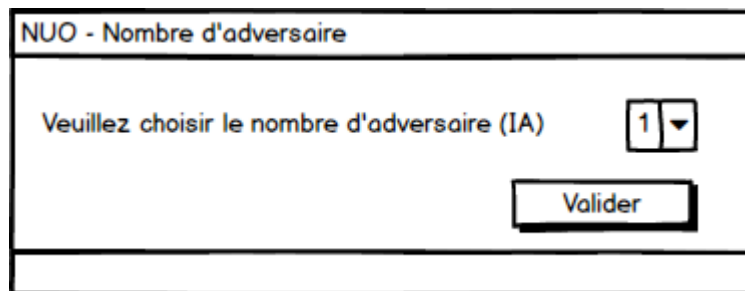
Veuillez insérer un pseudo

Entrer

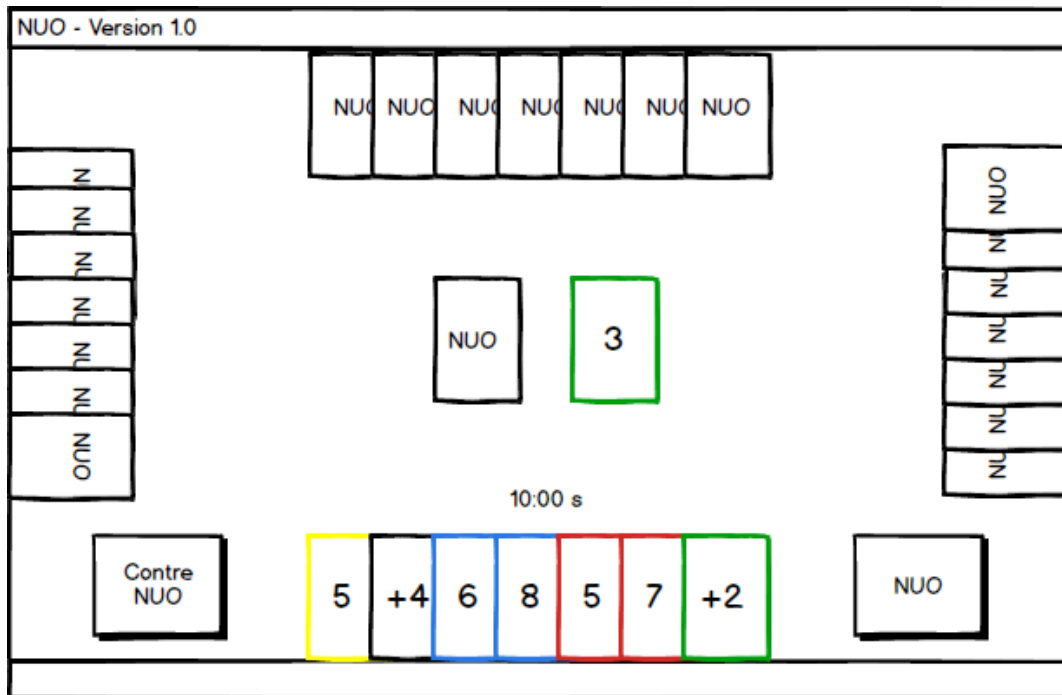
Menu principal avec liste des options (Suite à la validation du pseudo)



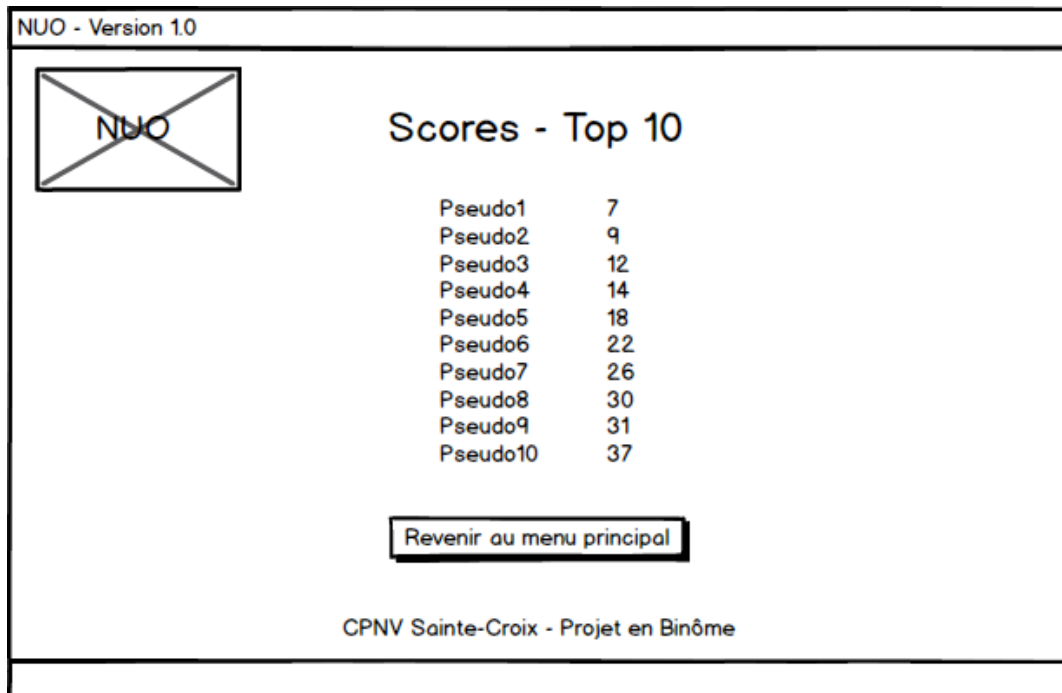
Fenêtre pop-up demandant le nombre d'adversaires (Suite au bouton « Lancer une partie »)



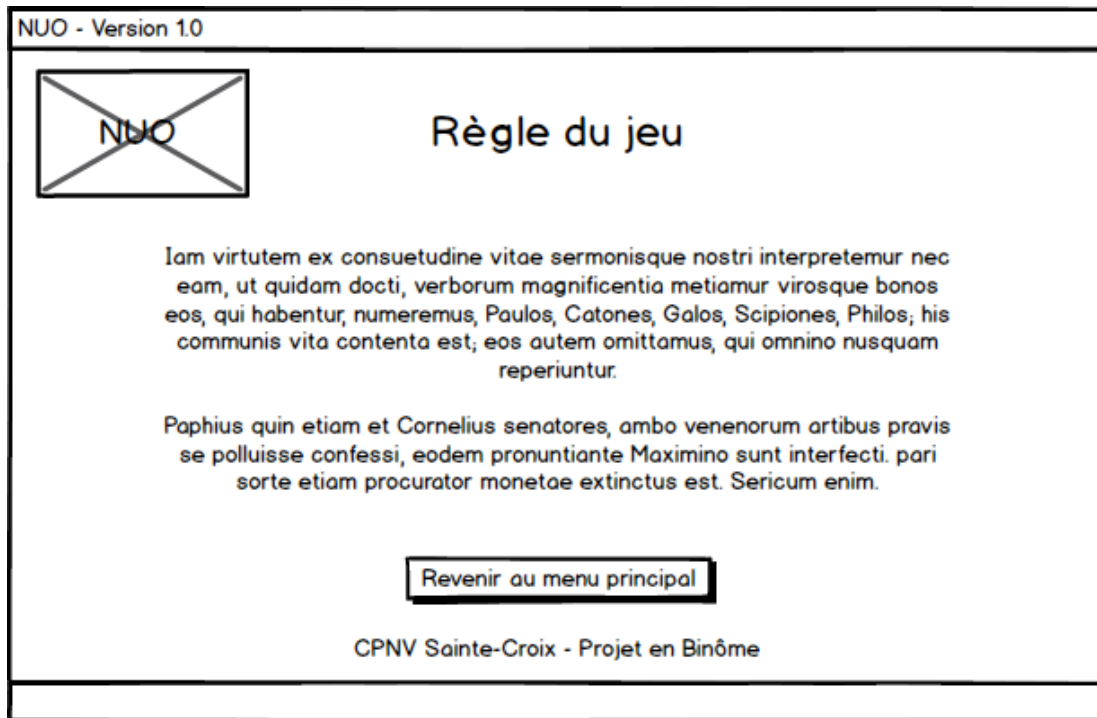
**Plateau de jeu avec les carte du joueur et le bouton « NUO », « Contre NUO »,
la pile et le tas principal du jeu**
(Suite à la validation du nombre d'adversaires)



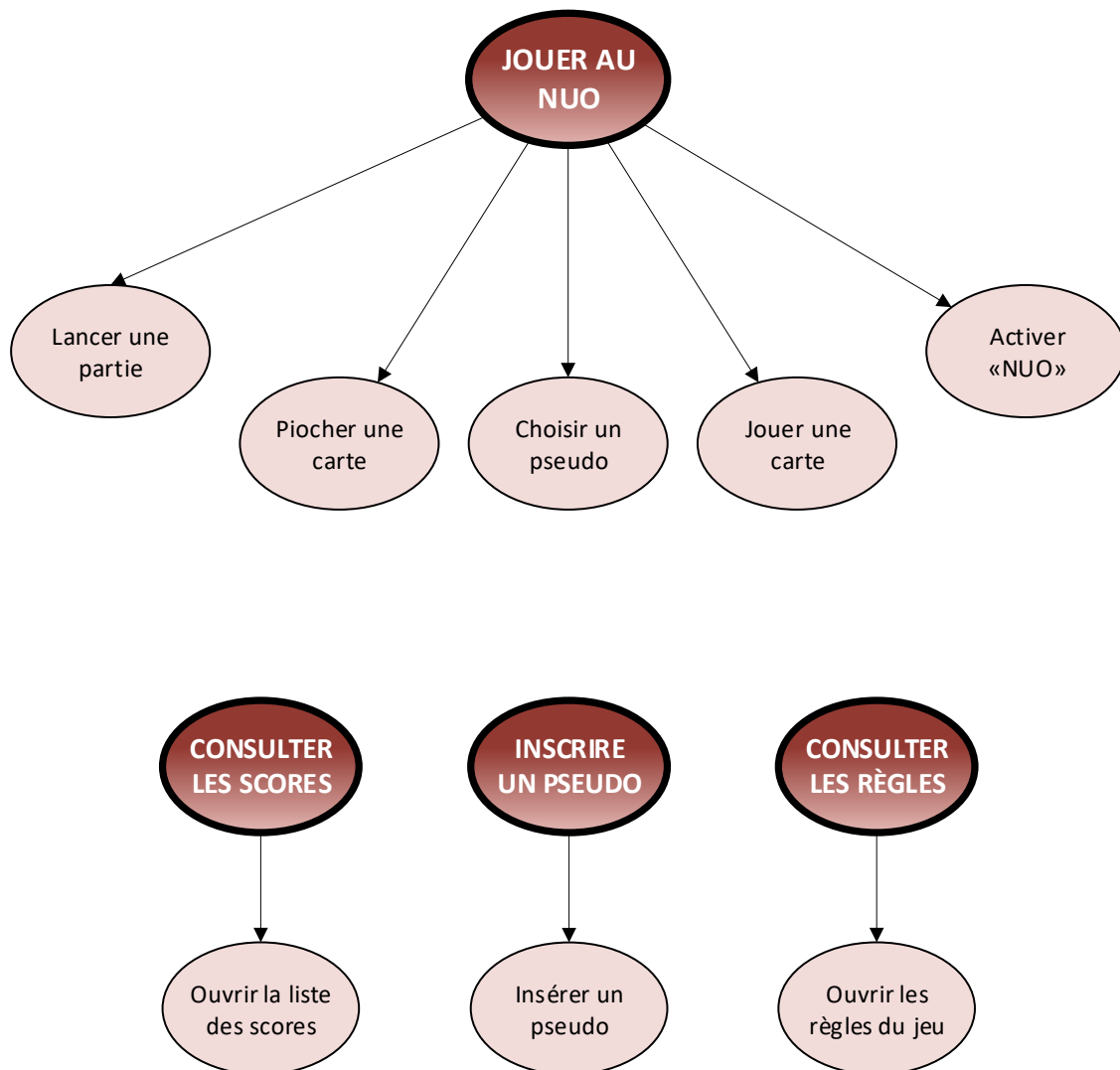
Page des 10 premiers scores
(Suite au bouton « Top 10 scores »)



Page des règles du jeu
(Suite au bouton « Règle du jeu »)



2.2 Cas d'utilisation et scénarios



2.2.1 Jouer au NUO

Scénario 2.2.1.1 – Lancer une partie

Action	Condition	Réaction
Après avoir lancé l'exécutable et validé son pseudo, l'utilisateur appuie sur « lancer une partie »		Le programme demande à l'utilisateur de choisir le nombre de IA (intelligence artificielle) contre lesquelles il veut jouer
L'utilisateur choisit un nombre et appuie sur « Valider »		La partie se lance avec le nombre d'IA sélectionné

Scénario 2.2.1.2 – Jouer une carte

Action	Condition	Réaction
Lors du tour de l'utilisateur, ce dernier sélectionnera, par un clique, une des cartes qu'il a en main et qu'il peut mettre en jeu		La carte se pose sur le tas de cartes principal du jeu et le tour de l'utilisateur se termine.
	L'utilisateur met plus de 10 secondes à jouer	L'utilisateur reçoit une pénalité de 1 carte automatiquement.

Scénario 2.2.1.3 – Piocher une carte

Action	Condition	Réaction
Lors du tour de l'utilisateur, ce dernier cliquera sur la pile de cartes retournées		La carte s'insère dans la main du joueur

Scénario 2.2.1.4 – Activer le mode « NUO »

Action	Condition	Réaction
L'utilisateur clique sur le bouton « NUO » afin d'annoncer qu'il ne lui reste plus qu'une carte	Un contre NUO a été prononcé par l'une des IA	L'utilisateur reçoit une pénalité de 2 cartes qu'il doit piocher et passe son tour
		Le jeu continue normalement

Scénario 2.2.1.5 – Activer le mode « Contre NUO »

Action	Condition	Réaction
L'utilisateur clique sur le bouton « Contre NUO » afin de contrer l'annonce d'un joueur	L'IA a pu prononcer son « NUO »	Le jeu continue normalement
		L'adversaire reçoit une pénalité de 2 cartes à piocher

2.2.2 Consulter les scores

Scénario 2.2.2.1 – Ouvrir la liste des scores

Action	Condition	Réaction
Après avoir lancé l'exécutable et valider son pseudo, l'utilisateur appuie sur « Top 10 scores »		Le programme ouvre le top 10 des scores enregistrés avec les pseudos en question

2.2.3 Inscrire un pseudo

Scénario 2.2.3.1 – Insérer un pseudo

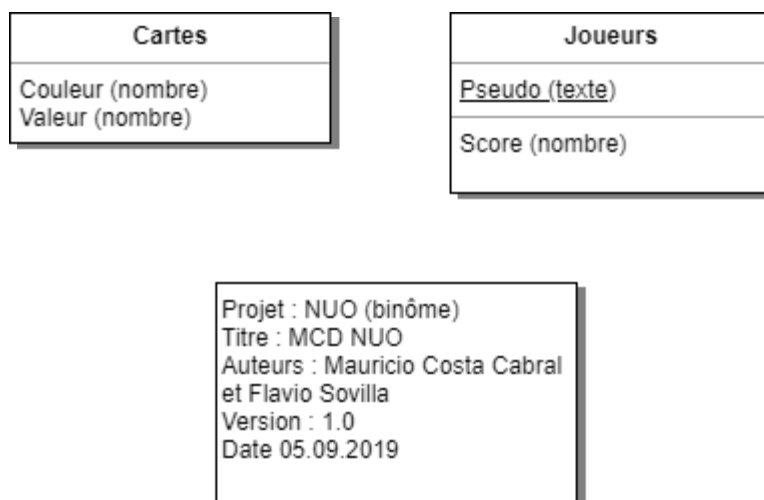
Action	Condition	Réaction
L'utilisateur lance l'exécutable		Le programme s'ouvre avec une page d'accueil demandant à l'utilisateur d'inscrire un pseudo et de le valider par le bouton en question
L'utilisateur inscrit un pseudo et valide ce dernier via le bouton en question		Le menu principal du programme apparaît

2.2.4 Consulter les règles du jeu

Scénario 2.2.4.1 – Ouvrir les règles du jeu

Action	Condition	Réaction
Après avoir lancé l'exécutable et validé son pseudo, l'utilisateur appuie sur « Règle du jeu »		Le programme affiche les règles du jeu

2.3 Modèle Conceptuel de Données



2.4 Stratégie de tests

Concernant notre stratégie de tests, cette dernière a été composée principalement de **tests manuels** exécutés tout au long du développement du jeu, notamment en démarrant l'application.

- Affichage de la valeur espérée
- Vérification de l'action d'un bouton
- Lancement de l'application avec des actions inattendues de la part de l'utilisateur
- ...

Cependant, nous avons également procédé à quelques **tests unitaires automatiques** qui nous ont permis de gagner un peu de temps sur cette partie du projet.

2.5 Budget

- Dépense en ressources humaines
 - 2 personnes pour environ 40 heures de travail chacun
- Dépense financière
 - Aucune

3 Implémentation

3.1 Vue d'ensemble

Notre jeu interagit très peu avec son entourage. Cependant, nous pouvons mentionner les éléments suivants :

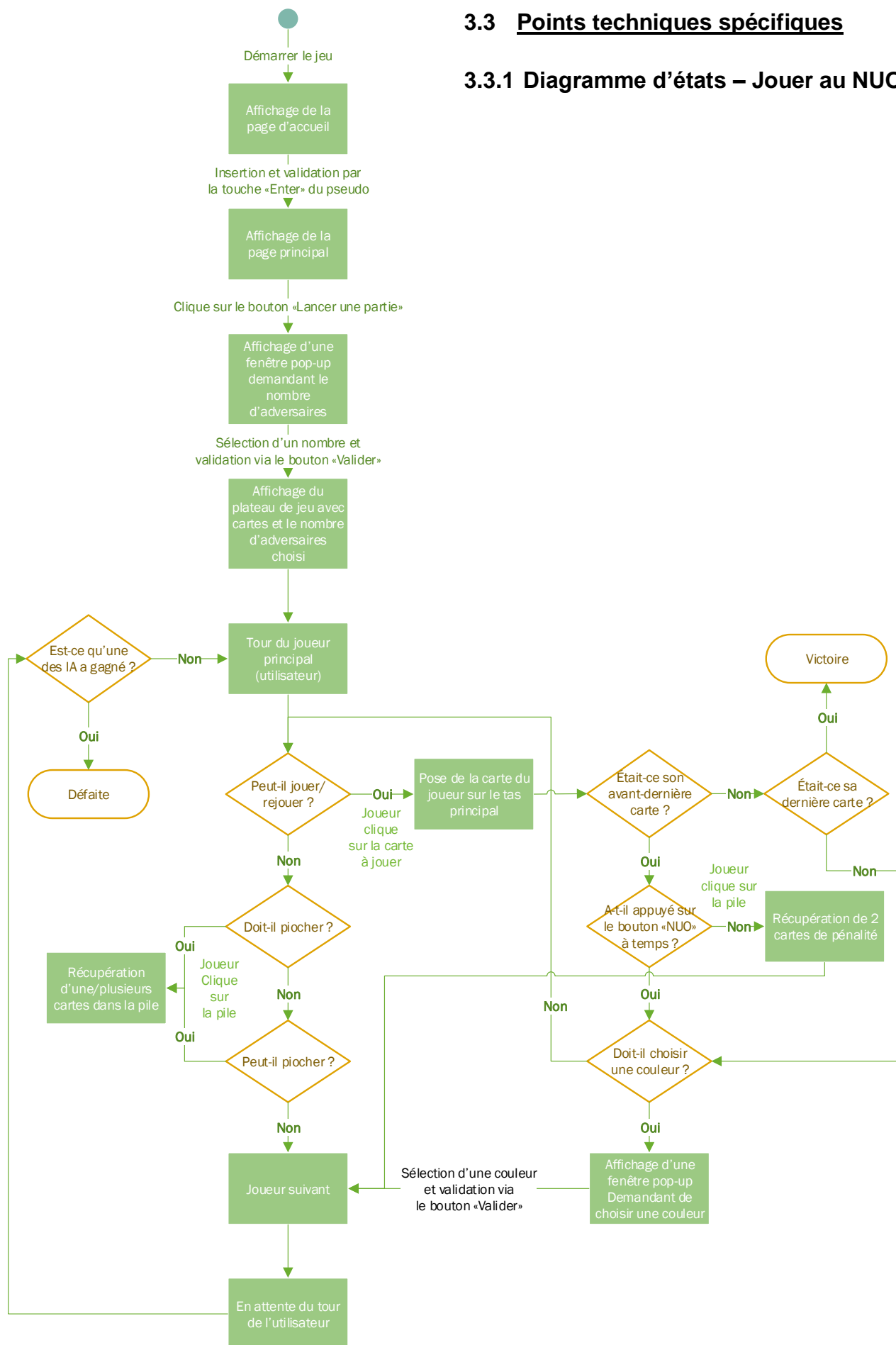
- L'utilisateur humain, qui sera le seul joueur concret du jeu
- La base de données, qui stocke les cartes du jeu ainsi que les pseudos et les scores des précédentes parties

3.2 Choix techniques

- Deux ordinateurs en configuration standard CPNV avec accès à internet
 - Outil de production imposé
- Outil de modélisation de base de données
 - Bases de données contenant les cartes et les pseudos/scores
- Environnement de développement C# (Visual Studio)
 - Logiciel où nous avons le plus connaissance et pratiqué
 - Utilisation du paquet System.Data.SQLite.Core
- Outil de versionning (Git)
 - Logiciel où nous avons le plus connaissance et pratiqué
- Base de données SQLite
 - Logiciel où nous avons le plus connaissance et pratiqué

3.3 Points techniques spécifiques

3.3.1 Diagramme d'états – Jouer au NUO



3.3.2 Description de la distribution aléatoire des cartes

Nous souhaitons expliquer davantage cette partie qui est une étape essentielle à l'ouverture du plateau de jeu.

Pour ce faire, nous nous sommes basés sur les identifiants des cartes provenant de la base de données et plus précisément de la table contenant les cartes. Globalement, nous en avons extraites le nombre nécessaire au début de la partie (en fonction du nombre de joueurs) de façon aléatoire et nous les avons distribuées également de manière aléatoire entre chaque joueur. Suite à cela, nous pouvons faire le lien entre ces données et les images des cartes du jeu, stocké dans un dossier. Ces dernières sont nommées uniquement à l'aide de nombre qui feront le lien avec les identifiants de la base de données.

```
//Objet permettant l'appel de la classe en lien avec la génération de nombre aléatoire
Random rand = new Random();

//Objet permettant l'appel de la classe en lien avec notre base de données
DBConnection nuodb = new DBConnection();

//Permettant l'appel de la classe en lien avec l'utilisateur principal
MainPlayer playerClass = new MainPlayer() ;

//Permettant l'appel de la classe en avec les intelligences artificielles
IA IA1Class = new IA();
IA IA2Class = new IA();
IA IA3Class = new IA();

//Stockage des identifiants (id) de toutes les cartes de la base de données
List<int> pile = nuodb.GetIdCards();

//Objet contenant l'ensemble des cartes choisies aléatoirement devant être distribuées
List<int> selectedIdCards = new List<int>();

//Boucle en fonction du nombre de joueurs permettant de choisir aléatoirement des cartes dans la pile (ex : 2 IA + le joueur principal → 3 joueurs * 7 cartes
for (int i = 1; i < (numberIA + 1) * 7 + 1; i++)
{
    //Stockage du nombre généré aléatoirement en fonction du nombre totale de cartes restantes
    int index = rand.Next(0, pile.Count);

    //Stockage de la carte choisie dans la variable des cartes sélectionnées
    selectedIdCards.Add(pile[index]);

    //Suppression de la carte choisie précédemment de la pile
    pile.RemoveAt(index);
}

//Tant que la variable contenant les cartes sélectionnées n'est pas vide (égale à 0) → distribution de 7 cartes à chaque joueur en commençant par le joueur principal (l'utilisateur)
while (selectedIdCards.Count() != 0)
{
    do
    {
```

```
//Stockage du nombre généré aléatoirement de 0 au nombre de cartes
sélectionnées
int index = rand.Next(0, selectedIdCards.Count);

//Ajout de la carte sélectionnée dans l'objet playerClass
playerClass.Cartes.Add(selectedIdCards[index]);

//Suppression de la carte provenant de la pile de cartes sélectionnées
selectedIdCards.RemoveAt(index);

//Boucle permettant de n'avoir que 7 cartes par joueur
} while (playerClass.Cartes.Count < 7);

//Répétition du code de distribution aléatoire pour les adversaires.

do
{
    int index = rand.Next(0, selectedIdCards.Count);
    IA1Class.Cartes.Add(selectedIdCards[index]);
    selectedIdCards.RemoveAt(index);

} while (IA1Class.Cartes.Count < 7);

//Si le nombre de cartes sélectionnées est plus grand que 0, cela signifie
qu'il y a un adversaire en plus dans la partie. Même condition
exécutée pour le 3ème et dernier adversaire, s'il reste des cartes
sélectionnées dans la variable « selectedIdCards »
if (selectedIdCards.Count > 0)
{
    do
    {
        int index = rand.Next(0, selectedIdCards.Count);
        IA2Class.Cartes.Add(selectedIdCards[index]);
        selectedIdCards.RemoveAt(index);

    } while (IA2Class.Cartes.Count < 7);
}
```

3.4 Livraisons

08/11/2019 – Rendu final, convenue avec la cliente et le chef de projet

- Cahier des charges
- Documentation du projet
- Documentation Doxygen
- Règle du jeu
- L'application (Base de données comprise)
- Lien Github

4 Tests

4.1 Tests effectués

Scénario	07/11/2019	07/11/2019
	Flavio SOVILLA (Élève)	Mauricio COSTA CABRAL (Élève)
	Windows 10	Windows 10
	Version 1.0	Version 1.0
Lancer une partie	X	X
Jouer une carte	Pas implémenté	Pas implémenté
Piocher une carte	Pas implémenté	Pas implémenté
Intercepter une carte	Pas implémenté	Pas implémenté
Activer le mode « NUO »	Pas implémenté	Pas implémenté
Choisir un pseudo	X	X
Ouvrir la liste des scores	X	X
Ouvrir les règles du jeu	X	X

4.2 Erreurs restantes

- Le déroulement du jeu en lui-même
 - Le plateau et la distribution aléatoire des cartes étant présentes, l'interaction de ces dernières ainsi qu'avec les joueurs n'existent pas. La complexité et le temps manquant nous ont empêché de développer cette partie.

5 Conclusion

Ce module nous a permis d'en apprendre davantage sur le déroulement d'un projet. Nous avons pu mettre en pratique différentes notions dont la méthode Agile, la communication avec le client, la supervision et l'approbation de notre chef de projet, notamment à l'aide de « stand-up meeting ».

Malheureusement, nous n'avons pas réussi à compléter tous les points de notre cahier des charges. En effet, la réalisation d'un jeu, comme le « UNO® », demande un niveau de connaissance en programmation assez conséquent. Nous avons fait de notre mieux en faisant énormément de recherches sur divers sites internet, mais le temps nous a manqué.

Voici un résumé, selon notre analyse, de l'état des objectifs devant être atteints en lien avec notre cahier des charges :

OBJECTIFS	ATTEINT ?
Respect des conventions de nommage	OUI
Architecture du code (SOC, POO, fonctions, ...)	OUI
Au moins un test unitaire est implémenté, à jour et pertinent	OUI
La gestion du travail en équipe est clairement identifiable et pertinente dans le rapport	OUI
L'interface devra être ergonomique afin d'être agréable pour le joueur, les choix devront être argumentés dans le rapport	PAS TERMINÉ
Le joueur doit pouvoir, à tout moment, savoir où le jeu en est	PAS TERMINÉ
Toutes les cartes doivent être différentes et clairement identifiables	OUI













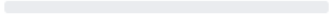
Malgré le défaut d'avoir une application inachevée, nous avons pu implémenter plusieurs fonctionnalités :

- Page d'accueil demandant un pseudo avec stockage de ce dernier dans la base de données.
- Connexion avec la base de données
- Affichage du menu principal avec les différentes options
- Affichage des règles du jeu
- Affichage des scores provenant de la base de données

- Fenêtre pop-up demandant à l'utilisateur le nombre d'adversaires
- Plateau de jeu avec distribution des cartes aléatoires.
- ...

La partie manquante de l'application porte essentiellement sur le développement concret du déroulement d'une partie (intelligences artificielles, interaction avec le joueur et les cartes, victoire/défaite d'une partie, etc...). Comme cité précédemment, le manque de temps et la complexité ne nous ont pas permis d'intégrer cette partie de l'application.

Le planning que nous avons élaboré, au début du projet, fut assez fidèle au rythme de notre développement réel de l'application. La méthode Agile ayant été employée, nous avons pu terminer nos **trois premiers Sprints** dans les temps. Suite à cela, des adaptations de notre planning ont dû être réalisées afin de se caler avec la réalité.

Planning général  Updated now	Sprint 1 - Finir le cahier des charges, use cases et scénarios Sprint 2 - Coder les menus et le plateau du jeu Sprint 3 - Créer la base de données et la plateforme Sprint 4 - Lancement avec choix IA et score Sprint 5 - Commentaires code, documentation et présentation Sprint 6 - Créer IA et gestion des cartes et finaliser les tests	...
Sprint 1 (28.08.19)  Updated on 4 Sep 	<ul style="list-style-type: none"> • Créer les cahier des charges (parties 3, 4 et 6) • Créer les use cases et scénarios • Planifier le tout sur Github (Issues et Projects avec basic kanban) 	...
Sprint 2 (12.09.19)  Updated on 11 Sep 	<ul style="list-style-type: none"> • Créer une page avec les règles du jeu • Créer le menu avec les différentes options • Créer le menu d'entrée graphique • Créer un MCD • Mettre à jour la documentation 	...
Sprint 3 (26.09.19)  Updated 26 days ago 	<ul style="list-style-type: none"> • Créer une base de données pour les scores • Créer une base de données pour les cartes • Créer la plateforme du jeu • Mettre à jour la documentation 	...
Sprint 4 (10.10.19)  Updated 26 days ago 	<ul style="list-style-type: none"> • Créer le choix du lancement du jeu avec 1, 2 ou 3 IA • Créer un tableau avec les 10 meilleures scores • Mettre à jour la documentation • Finaliser la plateforme du jeu 	...
Sprint 5 (07.11.19)  Updated 2 hours ago 	<ul style="list-style-type: none"> • Commenter le code avec Doxygen • Vérifier la convention de nommage • Mettre à jour la documentation 	...
Sprint 6 (05.12.19)  Updated 2 hours ago 	<ul style="list-style-type: none"> • Créer un IA • Faire la gestion des cartes • Tester le jeu final • Mettre à jour la documentation • Si le temps le permet, créer de nouvelles règles 	...

Finalement, un sprint est resté inachevé. Ce dernier comporte les tâches en lien avec le développement concret d'une partie.

Durant ce projet, nous avons pu relever plusieurs points positifs mais aussi négatifs. En effet, nous avons trouvé que le temps accordé à l'élaboration de ce projet était beaucoup trop court. L'étude du développement de l'application, les documentations, la présentation, les recherches internet, la résolution de pannes, les tests et surtout le développement du code en lui-même ne nous ont pas permis de terminer un travail d'une telle ampleur et complexité.

Hormis l'inachèvement de l'application, nous sommes largement satisfaits de la gestion du groupe. Nous avons eu une très bonne cohésion d'équipe ainsi qu'une excellente entente au sein du groupe. Nous avons pu, entre autre, nous répartir les tâches de façon équitable et juste afin que chacun puisse contribuer au bon déroulement du projet.

Pour terminer, ce projet nous rappelle encore une fois, que l'organisation, la communication et la collaboration sont les principaux piliers qui permettent d'augmenter significativement les chances de réussites des projets. Hormis l'acquisition de nouvelles connaissances en programmation, ce dernier nous a considérablement préparé à la réalisation de notre travail pratique individuel (TPI) que nous devons élaborer très prochainement.

6 Annexes

6.1 Sources – Bibliographie

- Sites internet
 - <https://openclassrooms.com/fr/courses/392266-developpement-c-net/391696-les-enumerations-et-structure>
 - <https://openclassrooms.com/fr/courses/2818931-programmez-en-orientee-objet-avec-c/2818971-la-poo-et-le-c>
 - <https://stackoverflow.com/questions/2706500/how-do-i-generate-a-random-int-number>
 - <https://stackoverflow.com/questions/30014901/generating-random-numbers-without-repeating-c>
- Personnes externes
 - Madame ANDOLFATTO Frédérique
 - Monsieur VIRET Loic

6.2 Journal de bord du projet

Date	Événement
28.08.2019	<ul style="list-style-type: none"> — Détermination des caractéristiques principales du jeu — Création d'un dépôt GitHub — Préparation et réalisation de la documentation (Maquette, use cases, scénarios, planning général, ...) — Réalisation du cahier des charges
04.09.2019	<ul style="list-style-type: none"> — Documentation (Use cases, scénarios, description, ...) — Codage (Règle du jeu)
05.09.2019	<ul style="list-style-type: none"> — Documentation (Description, organisation, MCD, ...)
11.09.2019	<ul style="list-style-type: none"> — Documentation (Diagramme d'états, corrections proposées par le chef de projet, ...) — Entretien avec le chef de projet (validation/correction des use cases et scénarios, planification générale) — Codage (Menu d'accueil avec insertion de pseudo, Menu principal) — Réalisation d'un logo
12.09.2019	<ul style="list-style-type: none"> — Documentation (Correction, ...) — Réalisation de la base de données (SQLite) — Etude de l'implémentation du plateau de jeu
25.09.2019	<ul style="list-style-type: none"> — Documentation (Correction, ...) — Codage (Règle du jeu, nombre d'adversaires) — Etude de l'implémentation du plateau de jeu
26.09.2019	<ul style="list-style-type: none"> — Documentation — Codage (Règle du jeu, nombre d'adversaires) — Etude de l'implémentation du plateau de jeu
02.10.2019	<ul style="list-style-type: none"> — Documentation — Etude de l'implémentation du plateau de jeu

03.10.2019	<ul style="list-style-type: none"> — Documentation — Codage (scores, liaison entre la BD scores et l'application) — Etude et implémentation du plateau de jeu
09.10.2019	<ul style="list-style-type: none"> — Documentation — Etude et implémentation du plateau de jeu
10.10.2019	<ul style="list-style-type: none"> — Documentation — Implémentation du plateau de jeu — Résolution de pannes (Conflit de merge Versionning - Git)
30.10.2019	<ul style="list-style-type: none"> — Documentation — Codage (Tests unitaires, plateau de jeu, correction convention de nommage, commentaires)
31.10.2019	<ul style="list-style-type: none"> — Documentation — Modification et finalisation des commentaires sur tout le code — Codage (plateau de jeu)
06.11.2019	<ul style="list-style-type: none"> — Documentation — Préparation des commentaires pour Doxygen — Codage (plateau de jeu)
07.11.2019	<ul style="list-style-type: none"> — Documentation — Préparation de la présentation