

Fais-moi un dessin
Spécifications des requis du système (SRS)

Version 2.5

Historique des révisions

Date	Version	Description	Auteur
2020-01-21	1.0	Rédaction initiale	Maxime Bernier, Florence Cimon-Paquet, Genevieve Laroche, Annie Rochette, Manyl Telmat
2020-01-22	1.1	Ajout dans la section description globale pour les interfaces	Florence Cimon-Paquet
2020-01-22	1.2	Ajout dans la section des exigences non-fonctionnelles	Maxime Bernier
2020-01-24	1.3	Ajout dans la section des exigences fonctionnelles	Geneviève Laroche, Annie Rochette, Maxime Bernier, Manyl Telmat
2020-01-25	1.4	Modification dans les exigences fonctionnelles et non-fonctionnelles	Florence Cimon-Paquet
2020-01-25	1.5	Révision du SRS	Florence Cimon-Paquet Annie Rochette
2020-02-01	2.0	Ajout dans la section des exigences fonctionnelles	Florence Cimon-Paquet
2020-02-04	2.1	Correction des sections 1 et 2	Florence Cimon-Paquet
2020-02-05	2.2	Correction et réorganisation des exigences fonctionnelles	Florence Cimon-Paquet
2020-02-06	2.3	Correction des exigences non-fonctionnelles	Manyl Telmat
2020-02-07	2.4	Ajout de spécifications pour les interfaces du serveur	Manyl Telmat
2020-02-07	2.5	Révision finale du SRS	Manyl Telmat

Table des matières

1. Introduction	5
1.1. But	5
1.2. Définitions, acronymes et abréviations	5
1.3. Vue d'ensemble du document	5
2. Description globale	6
2.1. Caractéristiques des usagers	6
2.2. Interfaces	6
2.2.1. Interfaces usagers	6
2.2.2. Interfaces matérielles	6
2.2.3. Interfaces logicielles	6
2.2.4. Interfaces de communication	6
2.3. Contraintes générales	6
2.4. Hypothèses et dépendances	7
3. Exigences fonctionnelles	7
3.1 Client lourd	7
3.1.1 Clavardage	7
3.1.2. Profil utilisateur et historique	7
3.1.3. Modes de jeu	8
3.1.4. Création d'un jeu	11
3.1.5. Personnalité des joueurs virtuels	12
3.1.6. Effets visuels et sonores	12
3.1.7. Tutoriel	13
3.1.8. Paramétrisation	13
3.1.9 Liste d'amis	13
3.2 Client léger	14
3.2.1 Clavardage	14
3.2.2. Profil utilisateur et historique	14
3.2.3. Modes de jeu	14
3.2.6. Effets visuels et sonores	14
3.2.7. Tutoriel	14
3.2.8. Paramétrisation	14
3.2.9 Liste d'amis	14
4. Exigences non-fonctionnelles	15
4.1. Utilisabilité	15
4.2. Fiabilité	15

4.3. Performance	15
4.4. Maintenabilité	15
4.5. Contraintes de conception	15
4.6. Sécurité	16

Spécifications des requis du système (SRS)

1. Introduction

1.1. But

Le SRS décrit le comportement externe d'une application. Il décrit aussi les exigences non fonctionnelles, les contraintes de conception, ainsi que les autres facteurs nécessaires à la description complète des exigences du logiciel à développer.

1.2. Définitions, acronymes et abréviations

Architecture client/serveur: Type de communication entre plusieurs client d'un réseau. Cette architecture va permettre aux utilisateurs de jouer avec les différents modes de jeu ainsi que communiquer entre eux.

BD: Base de données

C#: Langage de programmation utilisé pour le client lourd

Client lourd: Client exécutable dans un environnement Windows sur un ordinateur

Client léger: Client exécutable dans un environnement Android sur une tablette tactile

Dessin: Représentation d'une forme, d'un objet ou d'une expression

Dessinateur: Rôle qui consiste à représenter un mot par un dessin.

Devineur: Rôle qui consiste à deviner ce que le dessinateur tente de représenter.

GUI: Interface graphique utilisateur (*Graphic User Interface*)

Jeu: Mot ou expression à deviner

Joueur virtuel: Joueur de type ordinateur qui a des comportements préétablis

Partie: Ensemble de jeux permettant à un ou plusieurs utilisateurs de deviner et de dessiner, selon les modes offerts, les mots ou expressions mentionnés. La partie se termine lorsque le temps est écoulé. Un score sera attribué aux utilisateurs après chaque partie.

SVG: Graphique vectoriel adaptable, basé sur XML, représente un ensemble de graphiques vectoriels (*Scalable Vector Graphics*)

Tour: Les joueurs doivent deviner le mot ou l'expression. Pour le mode Mêlée générale, un rôle, dessinateur ou devineur, est assigné à chaque joueur pour chaque jeu d'une partie.

UX: Expérience utilisateur (*User Experience*)

WPF: Framework UI pour créer des applications pour le client lourd

1.3. Vue d'ensemble du document

Dans les sections ci-dessous, il sera question de décrire de manière globale le produit (section 2) et établir les exigences fonctionnelles et non fonctionnelles du logiciel (sections 3 et 4).

2. Description globale

Le projet consiste à construire un logiciel comportant trois parties: le client lourd, le client léger et le serveur. *Fais-moi un dessin* est un logiciel qui consiste à deviner l'image dessinée par un autre joueur. Il est possible de jouer soit en mode solo ou en mode multijoueur. Les différents clients sont capables de communiquer entre eux grâce à un serveur.

2.1. Caractéristiques des usagers

L'application a comme public cible des étudiants en génie logiciel. Ces utilisateurs doivent être familiers avec le système d'exploitation Windows 10 et avoir en leur possession un ordinateur portable supportant ce système ou doivent posséder une tablette tactile de type Galaxy Tab supportant les applications Android.

2.2. Interfaces

2.2.1. Interfaces usagers

Les interfaces usagers sont très simples et permettent aux usagers de naviguer dans l'application avec aisance dès la première utilisation grâce au guidage présent sur les interfaces et aussi en utilisant des éléments qu'on retrouve généralement sur les interfaces des applications du même type. L'interface du client lourd sera programmée en C#/WPF et celle du client léger en Java. L'interface du client léger devra être adaptée afin de supporter le support tactile.

2.2.2. Interfaces matérielles

Les interfaces matérielles utilisées pour *Fais-moi un dessin* sont les interfaces matérielles standards pour le client lourd (clavier, souris, écran, système de son) et une tablette tactile de style Galaxy Tab supportant des applications Android pour le client léger.

2.2.3. Interfaces logicielles

Le client lourd doit être fonctionnel sous Windows 10. Le framework utilisé pour ce dernier sera .NET (C#/WPF). Le client léger utilisera le framework JAVA API framework. L'environnement de développement pour le client léger sera Android Studio. Le serveur sera développé en utilisant nodeJS et la base de données utilisée sera MongoDB. L'engin de conversion SVG sera utilisé afin de convertir les fichiers d'images.

2.2.4. Interfaces de communication

Le type de serveur qui sera utilisé pour l'application est Node.js. Le serveur pourra communiquer avec la base de données, MongoDB, afin d'y ajouter des données. Le Wi-Fi de l'université Polytechnique Montréal sera utilisé afin de se connecter pour jouer à *Fais-moi un dessin*.

2.3. Contraintes générales

Le serveur mis en place doit être capable de supporter jusqu'à 8 utilisateurs simultanément. Un maximum de quatre utilisateurs peut joindre une même partie en mode multijoueurs. Le client lourd doit fonctionner sous Windows

tandis que le client léger doit fonctionner au moins sous Android 6.

2.4. Hypothèses et dépendances

Nous supposons que les utilisateurs possèdent une connexion internet suffisamment performante pour pouvoir utiliser toutes les fonctionnalités de l'application.

3. Exigences fonctionnelles

3.1 Client lourd

3.1.1 Clavardage

Intégration

3.1.1.1 Le système doit posséder une zone de clavardage à même l'environnement. (Essentiel)

3.1.1.1.1 Le système doit assurer la disponibilité du clavardage à l'utilisateur en tout temps.

Canaux de discussion

3.1.1.2 Le système doit permettre à l'utilisateur de créer un nouveau canal de discussion. (Essentiel)

3.1.1.3 Le système doit permettre à l'utilisateur de joindre un canal de discussion. (Essentiel)

3.1.1.4 Le système doit permettre à l'utilisateur de joindre plusieurs canaux de discussion. (Essentiel)

3.1.1.5 Le système doit permettre à l'utilisateur de quitter un canal de discussion. (Essentiel)

3.1.1.6 Le système doit permettre à l'utilisateur d'envoyer une invitation à un autre utilisateur à partir de sa liste d'amis ou de la liste des personnes en ligne pour qu'il rejoigne le canal de discussion. (Essentiel)

3.1.1.7 Le système doit permettre à l'utilisateur d'afficher l'historique de clavardage complet du canal. (Essentiel)

3.1.1.8 Le système doit permettre à l'utilisateur d'accéder le clavardage tout au long de l'utilisation du jeu. (Essentiel)

3.1.1.9 Le système doit censurer les mots offensants qui sont envoyés dans le clavardage en les remplaçant par des astérix. (Souhaitable)

3.1.2. Profil utilisateur et historique

3.1.2.1 Le système doit permettre à un nouvel utilisateur de s'enregistrer pour authentification future. (Essentiel)

3.1.2.2 Le système doit limiter l'accès au jeu à l'utilisateur authentifié. (Essentiel)

3.1.2.3 Le système doit garder de manière confidentielle le nom et le prénom de l'utilisateur. (Essentiel)

3.1.2.4 Le système doit garder de manière confidentielle le mot de passe de l'utilisateur. (Essentiel)

3.1.2.5 Le système doit camoufler le mot de passe de l'utilisateur lorsque celui-ci se connecte à son compte. (Essentiel)

3.1.2.6 Le système doit représenter l'utilisateur par un pseudonyme. (Essentiel)

3.1.2.7 Le système doit représenter l'utilisateur par un avatar. (Essentiel)

3.1.2.8 Le système doit offrir à l'utilisateur la possibilité de changer son avatar en important une image à partir de

fichiers externes. (Essentiel)

3.1.2.9 Le système doit permettre à un utilisateur de consulter des statistiques sur son utilisation du jeu. (Essentiel)

3.1.2.9.1 Le système doit permettre à l'utilisateur de consulter le nombre de parties qu'il a jouées.

3.1.2.9.2 Le système doit permettre à l'utilisateur de consulter son pourcentage de victoires.

3.1.2.9.3 Le système doit permettre à l'utilisateur de le temps moyen des parties qu'il a joué.

3.1.2.9.4 Le système doit permettre à l'utilisateur de consulter le temps total qu'il a joué.

3.1.2.9.5 Le système doit permettre à l'utilisateur de consulter les dates et heures de connexion/déconnexion.

3.1.2.9.6 Le système doit permettre à l'utilisateur de consulter la date et l'heure du début d'une partie jouée.

3.1.2.9.7 Le système doit permettre à l'utilisateur de consulter le nom des joueurs avec lesquels il a joué chaque partie.

3.1.2.9.8 Le système doit permettre à l'utilisateur de consulter le résultat pour une partie qu'il a joué.

3.1.2.9.9. Le système doit permettre à l'utilisateur de consulter son meilleur score en mode solo.

3.1.2.10 Le système doit offrir accorder un badge d'accomplissements à l'utilisateur selon son succès. (Souhaitable)

3.1.2.10.1 Le système doit permettre à l'utilisateur de voir ses accomplissements et ceux restants.

3.1.2.10.2 Le système doit offrir un trophée après que l'utilisateur ait joué sept jours consécutifs.

3.1.2.10.3 Le système doit offrir un trophée si l'utilisateur atteint la première place dans le classement général.

3.1.2.10.4 Le système doit offrir un trophée à l'utilisateur une fois qu'il ait joué au moins une fois à chaque mode de jeu offert.

3.1.2.10.5 Le système doit offrir un trophée à l'utilisateur si possède au moins 5 amis dans sa liste d'amis.

3.1.2.10.6 Le système doit offrir un trophée à l'utilisateur s'il joue une partie entre 3h à 5h du matin.

3.1.2.10.7 Le système doit offrir un trophée à l'utilisateur s'il gagne 5 parties de suite dans le mode "Mêlée générale".

3.1.2.10.8 Le système doit offrir un trophée à l'utilisateur s'il gagne 25 parties jouées.

3.1.2.10.9 Le système doit offrir un trophée à l'utilisateur s'il perd 5 parties de suite dans le mode "Mêlée générale".

3.1.2.10.10 Le système doit offrir un trophée à l'utilisateur s'il joue 10 parties de suite en "mode solo".

3.1.2.10.11 Le système doit offrir un trophée à l'utilisateur s'il possède tous les autres trophées.

3.1.3. Modes de jeu

3.1.3.1 Le système doit gérer les règles de jeu communes à tous les modes de jeu. (Essentiel)

3.1.3.1.1 Le système doit assigner un mot au dessinateur parmi les mots reconnus, soit ceux qui ont un dessin associé.

- 3.1.3.1.2 Le système doit offrir des indices aux joueurs durant la partie.
- 3.1.3.1.3 Le système fournit les indices à des intervalles définis.
- 3.1.3.1.4 Le système doit offrir la possibilité de désactiver l’affichage d’indices durant la partie.
- 3.1.3.1.5 Le système doit évaluer la validité de la réponse du devineur.
- 3.1.3.1.6 Le système doit offrir la possibilité d’enlever un ou plusieurs joueurs virtuels avant de débiter une partie selon le mode de jeu sélectionné.
- 3.1.3.1.7 Le système doit automatiquement ajouter tous les joueurs participant à une même partie dans un canal de discussion.
- 3.1.3.1.8 Le système doit mettre le bouton «Démarrer la partie» disponible lorsque le minimum de joueur est atteint pour démarrer la partie selon le mode de jeu sélectionné.
- 3.1.3.2 Le système doit permettre aux utilisateurs de jouer en mode “Mêlée générale”. (Essentiel)
 - 3.1.3.2.1 Le système doit accepter entre 2 et 4 joueurs en “Mêlée générale”.
 - 3.1.3.2.2 Le système doit établir le rôle du joueur entre dessinateur et devineur.
 - 3.1.3.2.3 Le système établit une alternance des rôles après chaque image.
 - 3.1.3.2.3.1 Le système s’assure qu’il y a uniquement un seul dessinateur par image.
 - 3.1.3.2.3.2 Le système doit inclure les joueurs virtuels dans la rotation des rôles.
 - 3.1.3.2.3.3 Le système doit s’assurer que tout joueur virtuel est passif dans le rôle de devineur.
 - 3.1.3.2.5 Le système doit accorder des points à l’utilisateur après chaque dessin.
 - 3.1.3.2.5.1 Le système doit accorder des points au dessinateur en fonction du nombre de bonnes réponses des devineurs.
 - 3.1.3.2.5.2 Le système doit accorder des points à chacun des devineurs ayant obtenu la bonne réponse selon le temps écoulé et le nombre de devineurs ayant déjà trouvé la bonne réponse.
 - 3.1.3.2.5.3 Le système doit accorder des points aux joueurs humains.
 - 3.1.3.2.6 Le système doit permettre au dessinateur humain d’accéder à l’interface PolyPaint pour représenter le mot ou l’expression déterminé par le système.
 - 3.1.3.2.7 Le système doit tracer l’image dessinée par le dessinateur humain en temps réels pour les autres joueurs humains.
 - 3.1.3.2.8 Le système doit permettre aux utilisateurs de répondre à tout moment lorsqu’un nouveau tour est commencé.
- 3.1.3.3 Le système doit permettre aux utilisateurs de jouer en mode “Sprint solo”. (Essentiel)
 - 3.1.3.3.1 Le système doit accepter un seul joueur réel pour le mode “Sprint solo” qui sera le devineur.
 - 3.1.3.3.2 Le système doit assigner aléatoirement un joueur virtuel qui sera le dessinateur.
 - 3.1.3.3.3 Le système doit afficher en tout temps le temps restant à l’utilisateur, son score et son nombre d’essais restants.

- 3.1.3.3.4 Le système doit accorder un nombre d'essais maximum par image pour deviner prédéfinis lors de la création du jeu.
- 3.1.3.3.5 Si le joueur ne donne pas de bonnes réponses et épuise son nombre d'essais maximum, le système passe à la prochaine image.
- 3.1.3.3.6 Le système doit accorder le bon nombre de point à l'utilisateur et ajuster le temps restant après chaque dessin.
 - 3.1.3.3.6.1 Le système accorde un point et du temps bonus au compte à rebours pour chaque bonne réponse.
- 3.1.3.3.7 Le système doit mettre fin à la partie quand le temps est écoulé.
- 3.1.3.3 Le système doit offrir le mode de jeu "Défi (Challenge)" (Souhaitable)
 - 3.1.3.3.1 Le système doit permettre au joueur d'indiquer sa volonté de défier un ou plusieurs amis avant de jouer une partie "Sprint Solo".
 - 3.1.3.3.1.1 Le système doit permettre à l'utilisateur de défier ses amis à partir de sa liste d'amis ou les joueurs connectés sur un simple clique d'un bouton "Challenge".
 - 3.1.3.3.2 Le système doit mémoriser l'ordre des images de la partie faisant l'objet d'un défi.
 - 3.1.3.3.3 Le système doit mémoriser le score de la partie faisant l'objet d'un défi.
 - 3.1.3.3.4 Le système doit envoyer une notification aux joueurs défiés.
 - 3.1.3.3.4.1 Le système doit indiquer le joueur qui a lancé le défi dans la notification.
 - 3.1.3.3.4.2 Le système doit indiquer le score à battre dans la notification.
 - 3.1.3.3.4.3 Le système doit offrir la possibilité d'accepter ou de rejeter le défi.
 - 3.1.3.3.5 Le système doit soumettre le joueur défié au même jeu que le joueur qui a défié si le défi est accepté.
 - 3.1.3.3.5.1 Le système doit s'assurer que le jeu ait les mêmes images.
 - 3.1.3.3.5.2 Le système doit s'assurer que le jeu se déroule avec les images dans le même ordre.
 - 3.1.3.3.6 Le système doit envoyer une notification au joueur qui a défié pour indiquer si son adversaire a réussi à dépasser le score à battre.
- 3.1.3.4 Le système doit offrir le mode de jeu "Sprint coopératif". (Souhaitable)
 - 3.1.3.4.1 Le système doit accepter entre 2 et 4 joueurs humains afin de débiter le sprint coopératif.
 - 3.1.3.4.2 Le système doit assigner aléatoirement un joueur virtuel qui sera le dessinateur.
 - 3.1.3.4.3 Le système doit afficher en tout temps le temps restant à l'utilisateur, le score commun de l'équipe et le nombre d'essais restants.
 - 3.1.3.4.4 Le système doit accorder un nombre d'essais par image pour deviner prédéfinis aux joueur, si les joueurs ne donnent pas de bonnes réponses et épuise le nombre d'essais communs, le système passe à la prochaine image.

- 3.1.3.4.5 Le système doit accorder le bon nombre de point à l'équipe et ajuster le temps restant après chaque dessin.
 - 3.1.3.4.5.1 Le système doit accorder un point et du temps bonus au compte à rebours pour chaque bonne réponse.
 - 3.1.3.4.5.2 Le système doit accorder aucun point et aucun temps bonus si l'équipe n'arrive pas à deviner le dessin et épuise son nombre d'essais alloués.
- 3.1.3.4.6 Le système doit mettre fin à la partie quand le temps est écoulé.
- 3.1.3.5 Le système doit offrir le mode de jeu "Dessin libre" pour le client léger. (Souhaitable)
 - 3.1.3.5.1 Le système doit permettre à l'utilisateur de dessiner librement avec les outils de PolyPaint.
 - 3.1.3.5.2 Le système doit permettre à l'utilisateur d'enregistrer son oeuvre dans sa tablette.
 - 3.1.3.5.3 Le système doit convertir le dessin en jpeg avant d'enregistrer le fichier dans la tablette.
 - 3.1.3.5.4 Le système doit offrir la possibilité à l'utilisateur de mettre son oeuvre comme avatar.
- 3.1.4. Création d'un jeu**
 - 3.1.4.1 Le système doit permettre à l'utilisateur de créer un jeu à partir d'un mot ou une expression. (Essentiel)
 - 3.1.4.1.1 Le système permet à l'utilisateur d'entrer des indices en lien avec l'expression choisie par le créateur
 - 3.1.4.1.2 Le système permet à l'utilisateur de dessiner un dessin en lien avec l'expression choisie
 - 3.1.4.2 Le système doit permettre au joueur de choisir le niveau de difficulté du jeu créé (la difficulté définit la vitesse de dessin du joueur virtuel, le temps alloué et le nombre de tentatives maximales). (Essentiel)
 - 3.1.4.3 Le système doit permettre à l'utilisateur de choisir le mode de dessin des joueurs virtuels. (Essentiel)
 - 3.1.4.3.1 Le mode classique qui dessine chaque trait dans le même ordre que le dessin effectué par l'utilisateur.
 - 3.1.4.3.2 Le mode aléatoire qui dessine les traits de façon aléatoire et celle-ci change à chaque partie.
 - 3.1.4.3.3 Le mode panoramique qui dessine selon les axes cartésiens c'est-à-dire de haut en bas, de gauche à droite et vis-versa.
 - 3.1.4.3.4 Le mode centré qui dessine l'image selon sa distance par rapport au centre. L'utilisateur aura le choix intérieur vers extérieur ou le contraire.
 - 3.1.4.4 Le système permet à l'utilisateur d'importer une image format bitmap qui fera office de dessin. (essentiel)
 - 3.1.4.4.1 Le système pourra convertir l'image fournie par l'utilisateur en un dessin fait de traits noirs
 - 3.1.4.4.2 Le système permet à l'utilisateur de choisir le mode de dessin (mode classique non disponible si l'utilisateur choisi cette option). (Essentiel)
 - 3.1.4.5 Le système doit permettre à l'utilisateur de créer un jeu à partir de la banque d'image et de mots de "Quick Draw" de Google. (Essentiel)

- 3.1.4.5.1 Le système permettra au joueur de cliquer sur un bouton et ainsi changer le mot et l'image jusqu'à ce que celui-ci trouve un jeu qui lui convient.
- 3.1.4.5.2 Le système donnera l'occasion au joueur de fournir un ou plusieurs indices en lien avec ce mot.
- 3.1.4.6 Le système doit permettre à l'utilisateur de créer un jeu à partir de Google Image selon une expression ou un mot choisi par l'utilisateur. (Souhaitable)
 - 3.1.4.6.1 Le système permettra au joueur de cliquer sur un bouton et ainsi changer la série d'images présentées à l'utilisateur jusqu'à ce que celui-ci trouve une image qui lui convient.
 - 3.1.4.6.2 Le système doit procéder à la conversion de l'image avec l'outil Potrace.

3.1.5. *Personnalité des joueurs virtuels*

- 3.1.5.1 Le système doit permettre à un utilisateur de jouer avec des joueurs virtuels. (Essentiel)
- 3.1.5.2 Les joueurs virtuels doivent pouvoir passer un commentaire lors d'une bonne réponse. (Essentiel)
- 3.1.5.3 Les joueurs virtuels doivent pouvoir passer un commentaire lors d'une mauvaise réponse. (Essentiel)
- 3.1.5.4 Les joueurs virtuels doivent pouvoir passer un commentaire lorsque 75% du temps alloué pour deviner une image s'est écoulé. (Essentiel)
- 3.1.5.5 Les joueurs virtuels doivent pouvoir passer un commentaire après une séquence de 3 ou plus bonnes réponses. (Essentiel)
- 3.1.5.6 Les joueurs virtuels doivent pouvoir passer un commentaire après une séquence de 3 ou plus mauvaises réponses. (Essentiel)
- 3.1.5.7 Les joueurs virtuels doivent débiter la partie avec un message d'accueil. (Essentiel)
- 3.1.5.8 Les joueurs virtuels doivent terminer une partie avec un message d'au revoir. (Essentiel)
- 3.1.5.9 Les joueurs virtuels doivent pouvoir référencer des événements passés. (Souhaitable)
 - 3.1.5.9.1 Les joueurs virtuels doivent pouvoir référencer les statistiques des joueurs.
 - 3.1.5.9.2 Les joueurs virtuels doivent pouvoir référencer l'historique des parties des joueurs.
- 3.1.5.10 Le système doit offrir des joueurs virtuels ayant des personnalités différentes. (Essentiel)
 - 3.1.5.10.1. Le système doit offrir un joueur virtuel avec une personnalité gentille.
 - 3.1.5.10.2. Le système doit offrir un joueur virtuel avec une personnalité méchante.
 - 3.1.5.10.3. Le système doit offrir un joueur virtuel avec une personnalité comique.

3.1.6. *Effets visuels et sonores*

- 3.1.6.1. Le système doit posséder deux groupes d'effets sonores et visuels. (Essentiel)
 - 3.1.6.1.1 Le système doit posséder un effets visuel (feux d'artifices) quand l'utilisateur fini la partie.

- 3.1.6.1.2 Le système doit posséder un effet sonore quand l'utilisateur donne une bonne réponse.
- 3.1.6.2. Le système doit posséder un troisième groupe d'effets sonores et visuels. (Souhaitable)
 - 3.1.6.2.1 Le système doit posséder un groupe d'effets quand l'utilisateur reçoit un nouveau message.

3.1.7. Tutoriel

- 3.1.7.1 Le tutoriel s'activera automatiquement lors de la première utilisation de l'application. (essentiel)
- 3.1.7.2 Le système doit permettre à un utilisateur d'activer un tutoriel. (essentiel)
- 3.1.7.3 Le tutoriel sera non interactif de type « suite d'images explicatives » pour les différents modes de jeu. (essentiel)
- 3.1.7.4 Le système permet à l'utilisateur de passer les images du tutoriel par le simple clique d'un bouton "suivant". (essentiel)

3.1.8. Paramétrisation

- 3.1.8.1 Le système doit permettre à un joueur de choisir entre un mode de couleur standard ou sombre. (Souhaitable)
- 3.1.8.2 La sélection du mode de couleur doit être conservé entre les connections. (Souhaitable)
- 3.1.8.3 La sélection du mode de couleur doit être conservé si l'utilisateur change de client (lourd/léger). (Souhaitable)
- 3.1.8.4 Le système doit permettre à un joueur de bloquer d'autres joueurs, virtuels et réels. (Souhaitable)
 - 3.1.8.4.1 Le système n'affiche plus les messages des joueurs bloqués à l'utilisateur.
 - 3.1.8.4.2 Les joueurs bloqués peuvent être "débloqués" par l'utilisateur qui les a bloqué.
 - 3.1.8.4.3 Les joueurs bloqué ne pourra plus jouer avec le joueur qui l'a bloqué.
 - 3.1.8.4.4 Le système ne fournira aucune notification pour avertir un utilisateur qu'il a été bloqué par un autre joueur
- 3.1.8.5 Le système doit permettre à un joueur de désactiver certaines personnalités des joueurs virtuels afin que leurs messages n'apparaissent pas dans le clavardage. (Souhaitable)

3.1.9 Liste d'amis

- 3.1.9.1 Le système doit permettre à l'utilisateur d'ajouter d'autres joueurs dans une liste d'amis. (Souhaitable)
 - 3.1.9.1.1 Le joueur aura accès à sa liste d'amis ainsi qu'à leur statut de connection.
 - 3.1.9.1.2 Le système envoi une notification au joueur lorsque celui-ci reçoit une demande d'amitié.
 - 3.1.9.1.3 Le système permet à l'utilisateur d'accepter ou de refuser une demande.
- 3.1.9.2 Le système doit permettre à l'utilisateur de connaître le statut de connexion des autres joueurs présents dans sa liste d'amis. (Souhaitable)

3.2 Client léger

3.2.1 Clavardage

Intégration

Les mêmes exigences que le client lourd sont appliquées.

3.2.1.1 Le système doit posséder une zone de clavardage à même l'environnement. (Souhaitable)

Canaux de discussion

Les mêmes exigences que le client lourd sont appliquées.

3.2.2. Profil utilisateur et historique

Les mêmes exigences que le client lourd sont appliquées.

3.2.2.1 Le système doit offrir à l'utilisateur la possibilité de changer son avatar en prenant une photo sur la tablette à même l'application. (Souhaitable)

3.2.3. Modes de jeu

Les mêmes exigences que le client lourd sont appliquées.

3.2.3.1 Le système doit offrir le mode de jeu "Dessin libre" pour le client léger. (Souhaitable)

3.2.3.1.1 Le système doit permettre à l'utilisateur de dessiner librement avec les outils de PolyPaint.

3.2.3.1.2 Le système doit permettre à l'utilisateur d'enregistrer son oeuvre dans sa tablette.

3.2.3.1.3 Le système doit convertir le dessin en jpeg avant d'enregistrer le fichier dans la tablette.

3.2.3.1.4 Le système doit offrir la possibilité à l'utilisateur de mettre son oeuvre comme avatar.

3.2.6. Effets visuels et sonores

Les mêmes exigences que le client lourd sont appliquées à l'exception de l'exigence x.

3.2.7. Tutoriel

Les mêmes exigences que le client lourd sont appliquées.

3.2.8. Paramétrisation

Les mêmes exigences que le client lourd sont appliquées.

3.2.9 Liste d'amis

Les mêmes exigences que le client lourd sont appliquées.

4. Exigences non-fonctionnelles

4.1. Utilisabilité

- 4.1.1 L'interface doit être suffisamment intuitive pour qu'un joueur arrive à joindre une partie en moins de 5 minutes lors de la première utilisation.
- 4.1.2 L'utilisateur doit pouvoir compléter le tutoriel en moins de 20 minutes.

4.2. Fiabilité

- 4.2.1 La couverture des tests doit être de 90% à l'exception des composantes graphiques.
- 4.2.2 Les composantes graphiques seront testées manuellement.
- 4.2.3 Le serveur doit fonctionner pendant 23 heures par jour avec 1 heure accordée pour la maintenance.
- 4.2.4 Le client doit pouvoir utiliser le jeu sans avoir de bogues l'empêchant de jouer. Cela n'inclut pas les problèmes de connexion du côté client.

4.3. Performance

- 4.3.1 Le serveur doit pouvoir supporter simultanément la connexion de 8 utilisateurs.
- 4.3.2 Les clients doivent posséder un taux de rafraîchissement d'image de 30 FPS.
- 4.3.3 Les clients ne doivent pas percevoir de délai (délai de maximum 50 ms) entre les joueurs d'une même partie. (partie synchronisée entre tous les joueurs).

4.4. Maintenabilité

- 4.4.1 L'écriture du code doit respecter les normes suivantes:
 - Le nom des fonctions et méthodes doit respecter le camelCase.
 - Pour les blocs if, else, for, do, while, switch et try, l'accolade d'ouverture ({) doit terminer une ligne contenant du code.
 - Pour les blocs if, else, for, do, while, switch et try, l'accolade de fermeture (}) doit être sur une nouvelle ligne.
 - Pour les poursuites de blocs else, else if, catch et while, cela sera mis sur la même ligne que l'accolade de fermeture (}).
 - Le nom des classes doit respecter le PascalCase.
 - Les commentaires doivent être sur une ou des lignes ne contenant pas de code.

4.5. Contraintes de conception

- 4.5.1 Le client lourd doit être programmé en WPF (C# et .NET).
- 4.5.2 Le client léger doit être programmé en Java.

4.5.3 L'outil de dessin du client léger doit être programmé à l'aide de l'API Android Canvas.

4.6. Sécurité

4.6.1 Les mots de passes des utilisateurs doivent être cryptés de façon à être illisibles.