UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des sciences
DÉPARTEMENT D'INFORMATIQUE

Enseignants : Joël GOOSSENS, Christian HERNALSTEEN
Cours : INFO-F-209 – Projet d'année 2

CITYLORD : SOFTWARE REQUIREMENTS DOCUMENT

GROUPE 2
Zakaria AHARRAR – Hakim BOULAHYA – David FISHEL – Cédric ORINX – Gabriel ORTEGA – Kaio LOPES

Année académique 2014 – 2015

Table des matières

Chapitre 1

Introduction

1.1 But du projet

test Nous allons implémenter CityLord, un jeu de gestion de ville multi-joueur. Celui-ci se jouera entre 2 à 8 joueurs différents qui pourront acheter, vendre, ou améliorer des bâtiments, dans une même villes. Le but du jeu est le même pour chaque joueur, être le dernier propriétaire dans la ville, ou bien être celui avec le plus grand capital (argent+valeur des bâtiments) à la fin du temps imparti.

Au départ, chaque joueur commence avec une parcelle prise au hasard, et une somme modeste lui permettant d'acheter un nombre limité de propriété (1 voir 2 parcelles ou bâtiments).

Pour gagner de l'argent, des visiteurs (non-controlé par les joueurs) doivent entrer dans un de leur établissement, y rester une durée déterminée, leur donner l'argent adapté à l'établissement où il se trouve, et disparaitra ensuite de la partie. Un visiteur ne peut apparaitre qu'à des endroits prédéfini sur la carte (ex : station de métro), il suivra alors un chemin également prédéfini (ex : vers la sortie de la ville), et entrera dans un établissement qu'il croise, si celui-ci l'interesse. Ce sera donc aux joueurs de bien positionner leurs bâtiments là où ils ont le plus de chances d'attirer un grand nombre de visiteur.

Les bâtiments ne peuvent accueuillir qu'un nombre limité de visiteur en même temps, accepter une même somme par visiteur, et se débarasser d'un visiteur qu'après une certaine durée. Un joueur peut, s'il le souhaite et pour un certain coût, amélioré son bâtiment, et alors augmenter la capacité maximale, la recette par visiteur et diminuer la durée d'occupation d'un visiteur.

Si un joueur est en faillite, à cause d'un coût d'entretien trop élévé face aux recettes, il sera alors obligé de vendre des propriétés. Il mettra alors sa propriété (parcelle+bâtiment) dans le catalogue de vente, ouvert à tous les joueurs. Il peut également s'il le souhaite, supprimer son bâtiment, ce qui réduira grandement les coût d'entretien, mais rendera les recettes nulles. (Une parcelle vide à un coût d'entretien, mais pas de revenu, ceci est pour éviter qu'un joeur achète tous les emplacements simplement pour empêcher les autres de construire).

1.2 Glossaire

— **Système :** Un système est un ensemble d'éléments interagissant entre eux selon certains principes ou règles. Dans ce cas, il s'agit du programme.
— **Point-and-click :** se dit d'une des actions qu'un utilisateur peut effectuer sur une interface utilisateur graphique avec pointeur. L'utilisateur déplace le pointeur avec un dispositif de pointage (souris, manette de jeu) sur un emplacement particulier de l'écran d'ordinateur, puis il appuie sur un bouton du dispositif pour déclencher l'action .
— **Serveur :** Un serveur est ordinateur dédié à l'administration d'un réseau informatique. Il gère l'accès aux ressources et aux périphériques et les connexions des différents utilisateurs.
— **Réseau informatique :**Ensemble des moyens matériels et logiciels mis en œuvre pour assurer les communications entre ordinateurs.
— **Pseudo :**Un pseudonyme est un nom d'emprunt qu'une personne porte pour exercer une activité sous un autre nom que celui de son identité officielle.
— **Map :**Carte prédéfinie, représentant une ville.
— **Catalogue :**Liste reprenant toutes les propriétés mises en vente par les joueurs ou par la villes.

1.3 Historique du document

1.9 (06/02/15) : Modification du But du projet, ajout au glossaire & index +précondition "Construire" - David

1.8 (15/12/14) : Mise à jour des exigences de domaine - Cédric

1.7 (14/12/14) : Ajout des déscriptions textuelles des uses cases (Premiers achats,Construire-Améliorer-Détruire,Achats entre joueurs) - David

1.6 (12/12/14) : Rajout et correction des déscriptions textuelles des use case, et ajout des exigences de domaine - Hakim

1.5 (11/12/14) : Déscriptions textuelles de l'interface de connexion - Kaio

1.4 (11/12/14) : Exigences fonctionnelles (Besoins de l'utilisateur) et ajout dans le glossaire et l'index de termes - David

1.3 (11/02/14) : Ajout du diagramme de classe - Equipe

1.2 (10/02/14) : Ajout des premières *use case* - Zakaria

1.1 (10/02/14) : Première version. (Aharrar Zakaria) Contient les points 1.1, 1.2, 1.3, 2 et 2.1 (partiellement) - Zakaria

1.0 (09/12/14) : Création du document - Hakim

Chapitre 2

Besoins de l'utilisateur

2.1 Exigences fonctionnelles

2.1.1 Interface de connexion

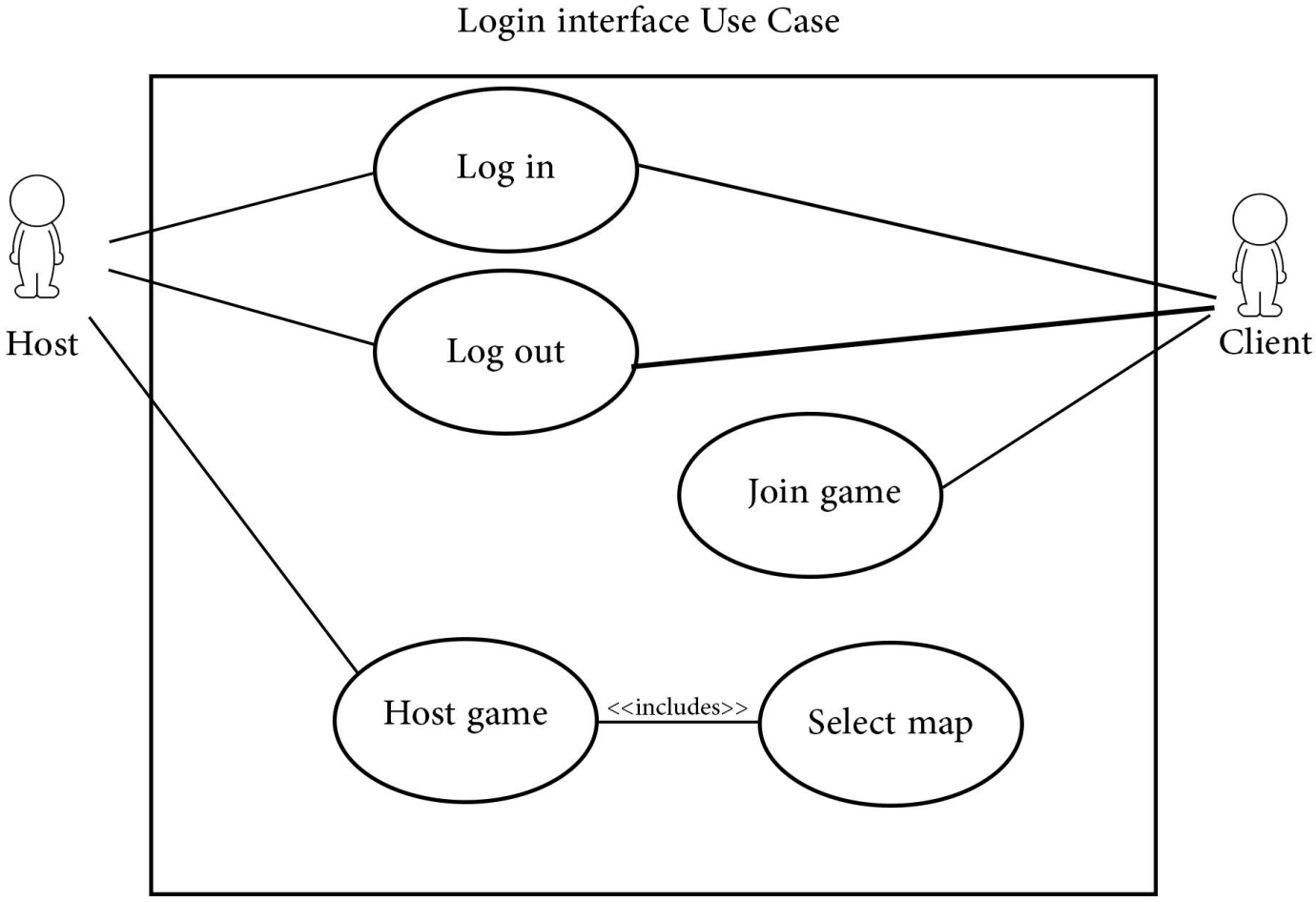

FIGURE 2.1 – Use Case Login/Logout

2.1.1.1 Log in

Préconditions
- Le jeu doit être ouvert.
- L'utilisateur ne doit pas être connecté.

Postconditions
- L'utilisateur est connecté.

Cas général Une fois le jeu démarrer, l'utilisateur peut se connecté insérant un nom d'utilisateur existant et le mot de passe associé [Exception : mauvais mot de passe ou pseudo non-existant]. Une fois toutes les informations entré, l'utilisateur appuie sur le bouton « *Connect* », et l'utilisateur est connecté. Un utilisateur non-enregistré peut également créer un nouveau pseudo unique et lui associé mot de passe en appuyant sur le bouton « *Create* »[Exception : pseudo déjà utilisé]. Une fenêtre va ensuite s'ouvrir et lui demandé d'entrer les informations nécessaires [Exception : aucun pseudo et/ou mot de passe entré]. Ceci ajoutera sur le serveur un nouvel utilisateur avec comme pseudo et mot de passe associés ceux entrés dans les zones appropriées, et connectera l'utilisateur automatiquement.

Exceptions
- *Mauvais mot de passe ou pseudo non-existant :* Après la pression du bouton « *Connect* »,sur la même vue, une remarque apparait, en une couleur fortement perceptible, et indique à l'utilisateur que les informations entrées sont incorrectes. La fenêtre de connexion reste ouverte et demande à l'utilisateur de réentrer son pseudo et son mot de passe.
- *Pseudo déja utilisé :* Après la pression du bouton « *Create* », sur la même vue, une remarque apparait, en une couleur fortement perceptible, et indique à l'utilisateur que le pseudo demandé est déjà utilisé. La fenêtre de création reste ouverte et demande à l'utilisateur de réentrer un nouveau pseudo.
- *Aucun pseudo et/ou mot de passe entré :* Dans les deux cas - de connexion ou de création - si l'utilisateur manque d'entrer une information, après du bonton associé à la fenêtre en premier plan, sur la même vue, une remarque apparait, en une couleur fortement perceptible, et indique à l'utilisateur l'information manquante. La fenêtre reste ouvert et demande à l'utilisateur d'entrer cette information pour continuer la connexion ou la création.

2.1.1.2 Log out

Préconditions
- L'utilisateur doit être connecté.

Postconditions
- L'utilisateur est déconnecté.
- La fenêtre du jeu est fermé.

Cas général Un joueur peut se déconnecté à n'importe quel moment. Si il est en partie, il la quitte automatiquement. L'utilisateur est ensuite déconnecté et la fenêtre du jeu se ferme.

Excéptions Néant.

2.1.1.3 Join game

Préconditions
— L'utilisateur est connecté.

Postconditions
— L'utilisateur est dans une partie.

Cas général Un utilisateur client peut rejoindre une partie en appuyant sur le bouton « *Join Game* »du menu principal. Il est ensuite envoyé dans l'écran de séléction du jeu, ou il lui est possible de voir toutes les parties lancées sur le même serveur et visualiser les différentes informations sur chacunes d'elles, comme le temps passé depuis le lancement de la partie ou le status de celle-ci. [Exception : la partie est complète] Après avoir rejoins une partie, si celle-ci n'est pas commencé le joueur est placé dans l'acceuil où il peut voir tous les joueurs en attente - du début de la partie -, dont l'hote de la partie, et la carte choisie. Il lui est également possible de chatter avec les autres joueurs. [Exception : Hôte ferme la partie] Dans le cas où la partie est déjà comencé le joueur reçoit de l'argent.

Exceptions
— *La partie est complète :* Une fenètre s'ouvre indiquant a l'utilisateur que la partie est complète et donc injoingnable. Une fois cette fenêtre fermée l'utilisateur est ramené au menu de selection.
— *Hôte ferme la partie :* L'utilisateur client est ramené au menu de selection [pour plus d'informations, voir Host Game].

2.1.1.4 Host game

Obligations spéciales
— Inclut Select Map.

Préconditions
— L'utilisateur doit être connecté.

Postconditions
— L'utilisateur hôte et les utilisateurs clients sont dans une partie.

Cas général L'utilisateur hôte est capable de créer une partie en appuyant sur le bouton « *Host game* »du menu principal. Une fois créée, l'hôte peut choisir une carte. L'hôte peut également chatter avec les autres utilisateurs. L'hôte peut également afermer l'acceuil en appuyer sur le bouton « *Close lobby* ». Tous les clients présent dans l'acceuil est renvoyé à l'écran de selection, tandis que l'hôte est renvoyé dans le menu principal. Quand il y a assez de joueur - minimum 2 et maximum 8 hôte inclut - le bouton « *Start game* »s'active. Quand il est cliqué, chaque utilisateur présent dans l'acceuil est lancé dans la partie et reçoit de l'argent.

Exceptions Néant.

2.1.1.5 Select map

Obligations spéciales
— Est inclut dans Host Game

Préconditions
— L'hôte doit être dans un acceuil.

Postconditions
— La carte choisie est chargée.

Cas général L'utilisateur hôte peut choisir une carte dans une liste prédéfinie de carte. Quand tous les éléments pour démarrer la partie sont réunis [pour plus d'informations, voir Host Game], la carte est chargée.

Exceptions Néant.

2.1.2 Achats libres

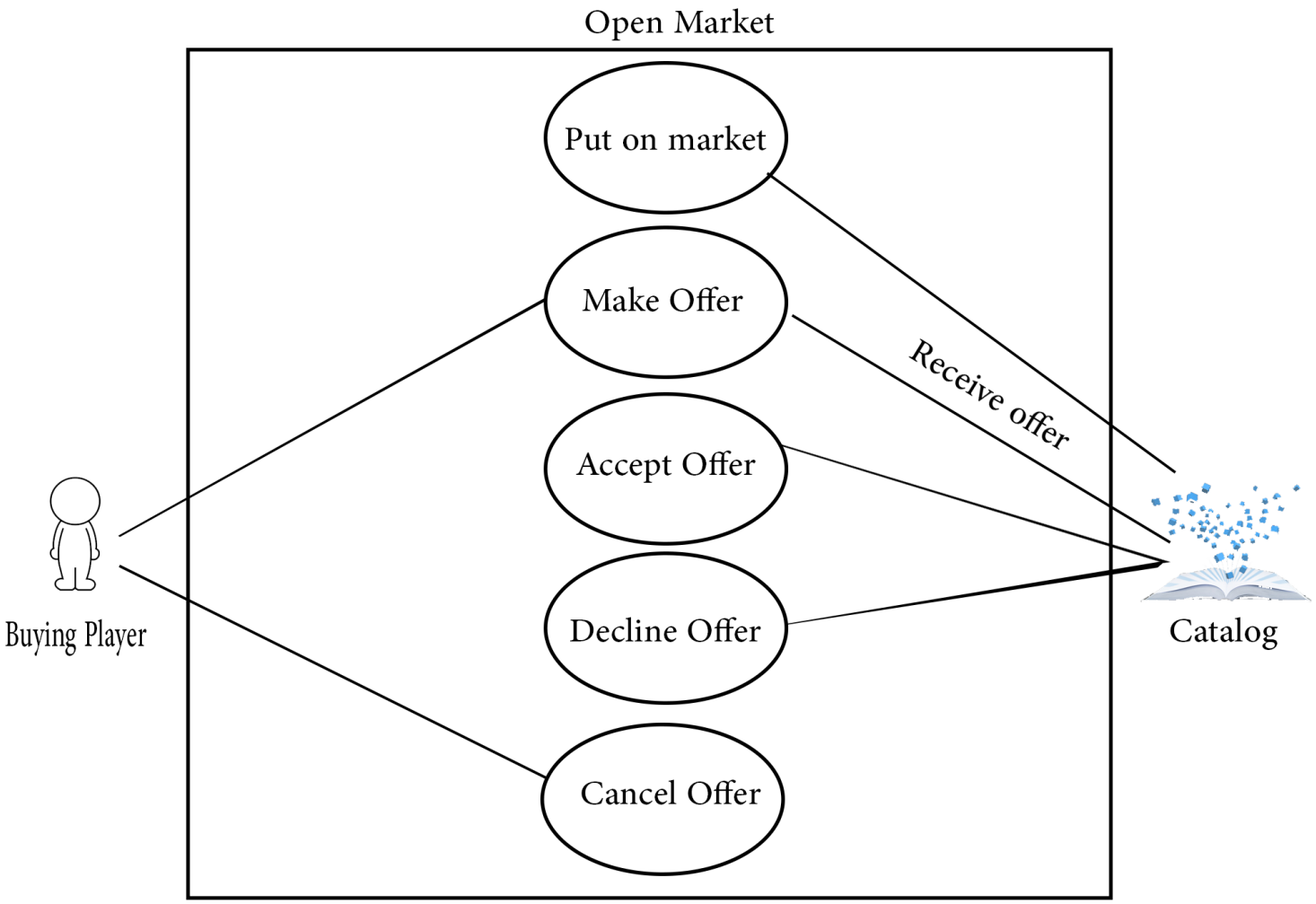

FIGURE 2.2 – Use Case Open Market

2.1.2.1 Faire une offre

Préconditions

— La partie est en cours.
— La propriété voulue n'appartient à personne.

Postconditions

— Une offre pour une propriété est envoyée.

Cas général A tout moment de la partie, n'importe quel joueur peut séléctionner une propriété dans le catalogue qui n'a pas de propriétaire, et faire une offre sur celle-ci. Après avoir séléctionner la propriété, ses informations sont affichées, et le joueur peut cliquer sur acheter.[Exception : Le joueur n'a pas assez d'argent]

Exceptions

— *Le joueur n'a pas assez d'argent :* Une fenètre s'ouvre indiquant au joueur qu'il ne possède pas assez d'argent.

2.1.2.2 Annuler une offre

Préconditions

— La partie est en cours.
— Le joueur à fait une offre qui n'as pas encore été validée.

Postconditions

— L'offre est annulée et retirée.

Cas général Dès que le joueur a mit une offre et tant que celle-ci n'a pas été acceptée, ce joueur peut annuler cette offre.

Exceptions Néant.

2.1.2.3 Accepter une offre

Préconditions

— La partie est en cours.
— Un joueur a fait une offre.

Postconditions

— Le joueur paie le prix de la propriété.
— Le joueur devient propriétaire de la propriété.

Cas général Lorsque le catalogue reçoit une offre sur un bâtiment, il accepte automatiquement celle-ci.[Exception : Deux joueur font la demande en même temps] Une fois l'offre acceptée, le joueur doit confirmer son achat ou l'annuler.

Exceptions

— *Deux joueur font la demande en même temps :* La première offre reçue sera prise en compte, l'autre sera annulée.

2.1.2.4 Mettre en vente

Préconditions

— La partie est en cours.
— Il existe un bâtiment sans propriétaire.

Postconditions

— Le bâtiment est en vente libre.

Cas général Dès qu'un bâtiment n'as pas de prioritaire, le catalogue le met en vente libre. Les bâtiments créés en début de partie seront mit dans le catalogue, ainsi que les batiments d'un joueur qui vient de perdre qui sont mit en vente libre.

Exceptions Néant.

2.1.3 Construire-Améliorer-Détruire

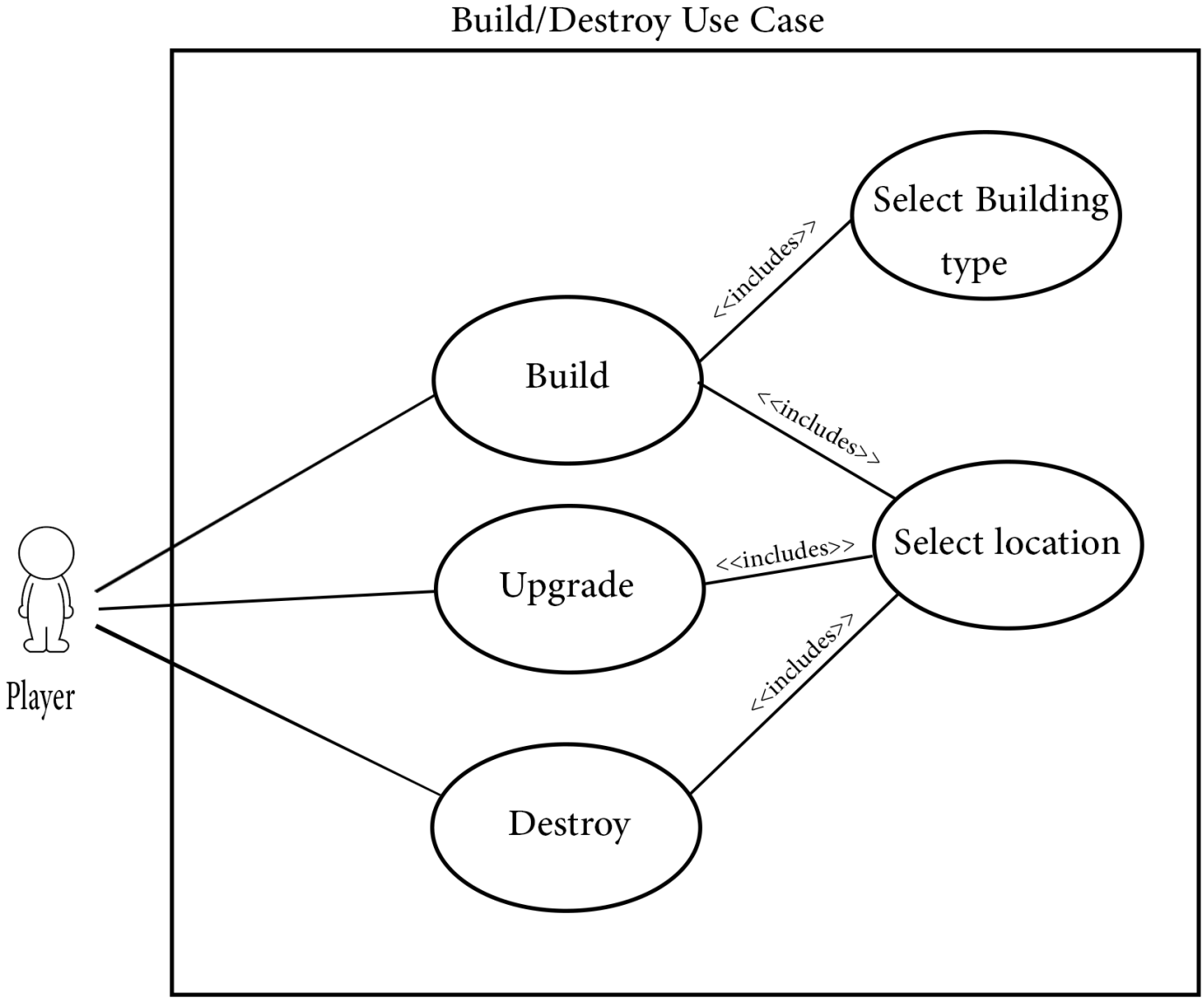

FIGURE 2.3 – Use Case Buid/Upgrade/Destroy

2.1.3.1 Construire

Obligations spéciales

— Inclut Select Building Map.
— Inclut Select Location.

Préconditions

— La partie est en cours.
— Le joeur possède au moins 1 parcelles vides.

Postconditions
- Un bâtiment est construit.
- Le joueur possède ce nouveau bâtiment.

Cas général Lorsque un joueur veut construire un nouveau bâtiment, il lui faudra simplement séléctionner l'option « *Construire* », il entre alors en mode Construction. Il peut à tout moment quitter ce mode et annuler la construction, en appuyant sur « *-ESC-* »ou en cliquant sur « *annuler* ». Un "catalogue" de type de bâtiment lui sera proposé, après qu'il en ai séléctionné un[Exception : Le joueur n'as pas assez d'argent], il devra cliquer sur le terrain sur lequel il veut construire.[Exception : Propriété séléctionnée non-valide]

Exceptions
- *Le joueur n'as pas assez d'argent :* Une fenètre s'ouvre indiquant au joueur qu'il ne possède pas assez d'argent.
- *Propriété séléctionnée non-valide :* Si le joueur séléctionne une proprité qui ne lui appartient pas, où qui ne possède déjà un bâtiment, une fenètre s'ouvre indiquant au joueur sa mauvaise sélection.

2.1.3.2 Upgrade

Obligations spéciales
- Inclut Select location.

Préconditions
- La partie est en cours.

Postconditions
- Un bâtiment est amélioré d'un niveau.

Cas général Un joueur peut à tout moment améliorer un de ses bâtiments. Il lui suffit de cliquer sur l'option « *Améliorer* », il entre alors en mode Amélioration. Il peut à tout moment quitter ce mode et annuler l'amélioration, en appuyant sur « *-ESC-* »ou en cliquant sur « *annuler* ». Et ensuite séléctionner une de ses propriétés.[Exception : Bâtiment amélioré au max]Des informations sur l'amélioration sont affichés, le prix de celle-ci, et les stats du bâtiments après celle-ci.Il ne reste plus qu'au joueur de confirmer l'amélioration en cliquant sur « *Ok* »ou annuler en cliquant sur « *Annuler* ».[Exception : Le joueur n'as pas assez d'argent][Exception : Propriété séléctionnée non-valide]

Exceptions
- *Le joueur n'as pas assez d'argent :* Une fenètre s'ouvre indiquant au joueur qu'il ne possède pas assez d'argent.
- *Propriété séléctionnée non-valide :* Si le joueur séléctionne une proprité qui ne lui appartient pas, où qui ne possède pas de bâtiment, une fenètre s'ouvre indiquant au joueur sa mauvaise sélection.
- *Bâtiment amélioré au max :* Les bâtiments ont un niveau maximum d'amélioration Une fenètre s'ouvre indiquant au joueur qu'il ne peut plus améliorer ce bâtiment.

2.1.3.3 Destroy

Obligations spéciales
— Inclut Select location.

Préconditions
— La partie est en cours.

Postconditions
— Un bâtiment est détruit et n'éxiste plus.

Cas général Un joueur peut également détruire un de ses bâtiments. Il lui suffit de cliquer sur l'option « *Détruire* », il entre alors en mode Destruction. Il peut à tout moment quitter ce mode et annuler la destruction, en appuyant sur « *-ESC-* »ou en cliquant sur « *annuler* ». Et ensuite séléctionner le bâtiment qu'il désire détruire.[Exception : Le joueur n'as pas assez d'argent] [Exception : Propriété séléctionnée non-valide]

Exceptions
— *Le joueur n'as pas assez d'argent :* Une fenètre s'ouvre indiquant au joueur qu'il ne possède pas assez d'argent.
— *Propriété séléctionnée non-valide :* Si le joueur séléctionne une proprité qui ne lui appartient pas, où qui ne possède pas de bâtiment, une fenètre s'ouvre indiquant au joueur sa mauvaise sélection.

2.1.3.4 Select Building type

Obligations spéciales
— Est inclut dans Build.

Préconditions
— Le joueur est entré en mode Construction.

Postconditions
— Un type est séléctionné pour la construction.

Cas général Un "catalogue" de types prédéfinis de bâtiment(bar,magasin,night-club,...) apparait, pour chaque type les statistiques (prix, coûts, capacité, heures,..) sont affichés. Il suffit au joueur alors de cliquer sur le type désiré.

Exceptions Néant.

2.1.3.5 Select Location

Obligations spéciales

— Est inclut dans Build.
— Est inclut dans Upgrade.
— Est inclut dans Destroy.

Préconditions

— Le joueur demande la construction/amélioration/destruction d'un bâtiment.

Postconditions

— Une propriété est séléctionnée pour la construction/amélioration/destruction et ses informations sont affichés.

Cas général Le joueur doit simplement cliquer, sur la carte, sur le terrain exigé.[Exception : Sélection invalide]

Exceptions

— *Sélection invalide :* Si l'emplacement séléctionné n'est pas un terrain/bâtiment (et donc une route ou un obstacle), la sélection n'est pas faite.

2.1.4 Achats entre joueurs

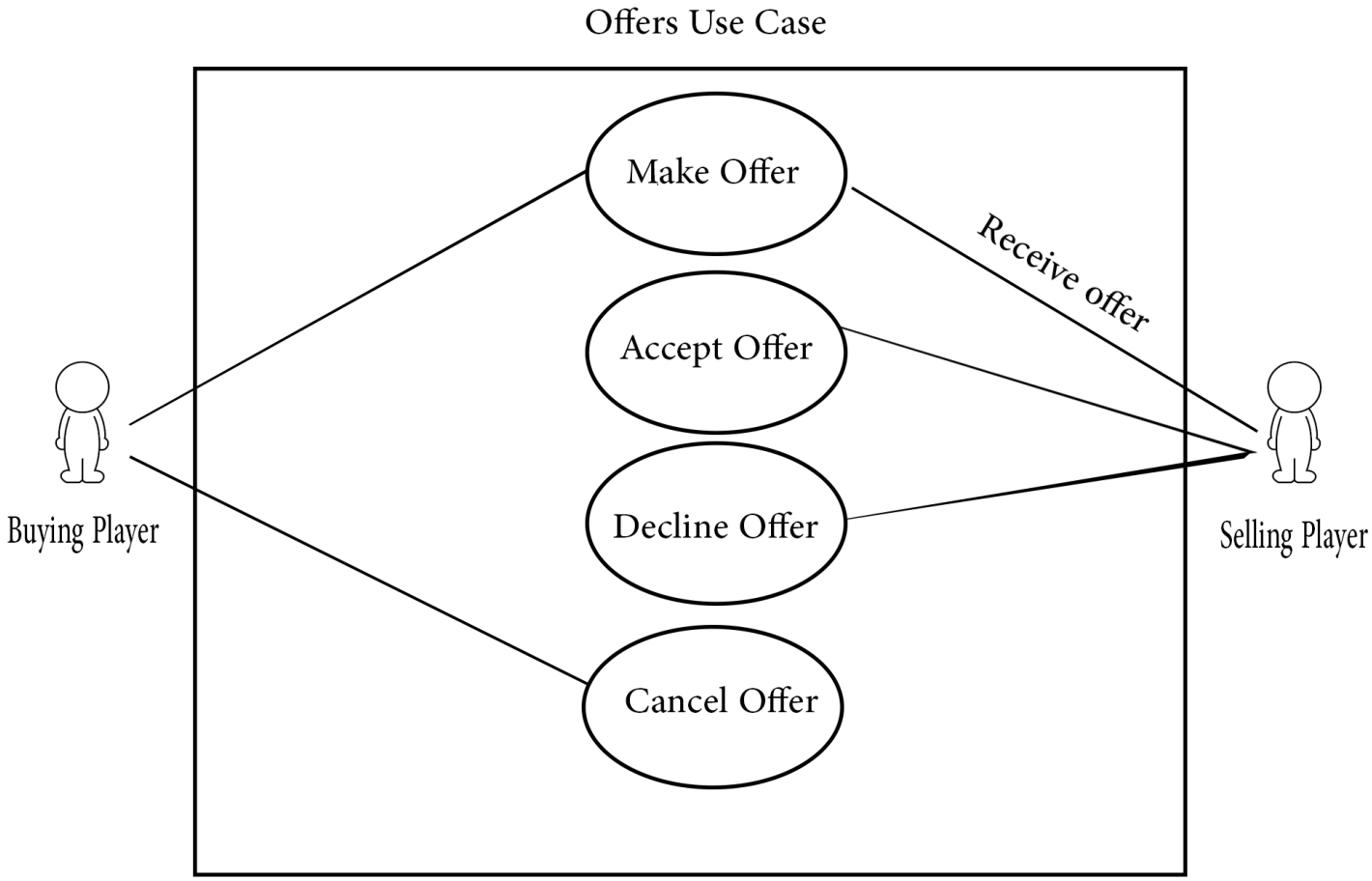

FIGURE 2.4 – Use Case between players sale

2.1.4.1 Faire une offre

Préconditions

— La partie est en cours.
— La propriété voulue appartient à un joueur.

Postconditions

— Une offre pour une propriété est envoyé à un autre joueur.

Cas général A tout moment de la partie, un joueur peut tenter d'acheter le bâtiment d'un autre. Il lui doit simplement de faire une offre auprès de ce joueur, pour ce bâtiment.[Exception : Le joueur a fait trop d'offre] Il suffit à l'acheteur de sélectionner le bâtiment voulu et de cliquer sur « *Faire une offre* », une fenêtre s'ouvre pour que le joueur puisse donner le prix qu'il est prêt a payer et confirme son offre.[Exception : Le joueur n'a pas assez d'argent]

Exceptions
- — *Le joueur n'a pas assez d'argent :* Une fenètre s'ouvre indiquant au joueur qu'il ne possède pas assez d'argent.
- — *Le joueur a fait trop d'offre :* Afin d'éviter tout spam, une limite d'offre sur un temps donné est instauré.

2.1.4.2 Annuler une offre

Préconditions
- — La partie est en cours.
- — Le joueur à fait une offre qui n'as pas encore été validée.

Postconditions
- — L'offre est annulée.

Cas général Dès que le joueur a mit une offre et tant que celle-ci n'a pas été acceptée, ce joueur peut annuler cette offre.

Exceptions Néant.

2.1.4.3 Accepter une offre

Préconditions
- — La partie est en cours.
- — Un joueur a reçu une offre.

Postconditions
- — L'acheteur devient propriétaire de la propriété.
- — Le vendeur reçoit une compensation monétaire et perd sa propriété.

Cas général Lorsque un joueur reçoit une offre sur un bâtiment, une fenêtre s'affiche avec les informations sur cette offre, il peut accepter celle-ci en cliquant sur « *Accepter* ».

Exceptions Néant

2.1.4.4 Refuser une offre

Préconditions
- — La partie est en cours.
- — Un joueur a reçu une offre.

Postconditions
- — L'offre est annulée.

Cas général Lorsque un joueur reçoit une offre sur un bâtiment, une fenêtre s'affiche avec les informations sur cette offre, il peut refuser celle-ci en cliquant sur « *Refuser* ».

Exceptions Néant

2.2 Exigences non fonctionnelles

2.2.1 Service

Le système fournira en premier lieu, l'assurance de la fonctionnalité du programme, et ce, sans bug ni ruptures.

Les différents outils du programme accompliront exactement la tâche décrite et demandée.

De plus, ces dites tâches seront claires et ne laisseront pas place à une ambiguïté quelconque. Chaque outil du programme sera représenté par une image explicitant la fonction, et une explication claire sera accompagnée.

Le système sera esthétique et disposera donc d'une apparence conviviale, ludique, mais également professionnelle. Celle-ci se distinguera par la qualité des textures utilisées, ainsi que par leur justesse par rapport aux informations qu'elles représentent.

Enfin, le système jouira d'une utilisabilité et d'une ergonomie agréable.

2.2.2 Contraintes

Afin de fonctionner correctement, et d'avoir une expérience de jeu optimale, le système devra être lancé sur un ordinateur possédant la configuration nécessaire. En cas d'environnement peu performant, il est possible que des ralentissements, ou qu'un manque de fluidité se fasse ressentir.

Il est également nécessaire que le système soit accompagné d'une connexion internet stable et rapide. Sans cela, la communication avec le serveur et entre les joueurs serait endommagée, et endommagerait la jouabilité de tous les joueurs.

2.3 Exigences de domaine

- Le jeu est multijoueurs, et doit donc permettre aux différents utilisateurs de communiquer entre eux.
- Une partie doit être composé d'au minimum 2 joueurs et au maximum 8 joueurs.
- Les 8 premiers joueurs qui rejoingnent la partie seront les seules 8 joueurs capable de la rejoindre.
- Pour qu'un joueur puisse continuer à jouer, il doit posséder soit des terrains, des batiments ou encore avoir un capital.
- Le capitale de chaque joueur varie tout au long de la partie en fonction des frais et des gains de chaque batiment.
- Les gains qu'apporte chaque batiment seront représentés par des visiteurs, qui, en suivant un chemin, pourront rentrer dans un batiment qui augmentera le capital de son propiétaire.
- Si le joueur est en faillite, la partie se terminera pour ce joueur et il ne sera plus possible pour lui de continuer de jouer dans cette partie, mais il pourra toute fois la regarder.
- La partie se finie uniquement si il ne reste plus qu'un joueur propriétaire.

Chapitre 3

Besoins du système

3.1 Exigences fonctionnelles

3.2 Exigences non fonctionnelles

3.3 Design et fonctionnement du système

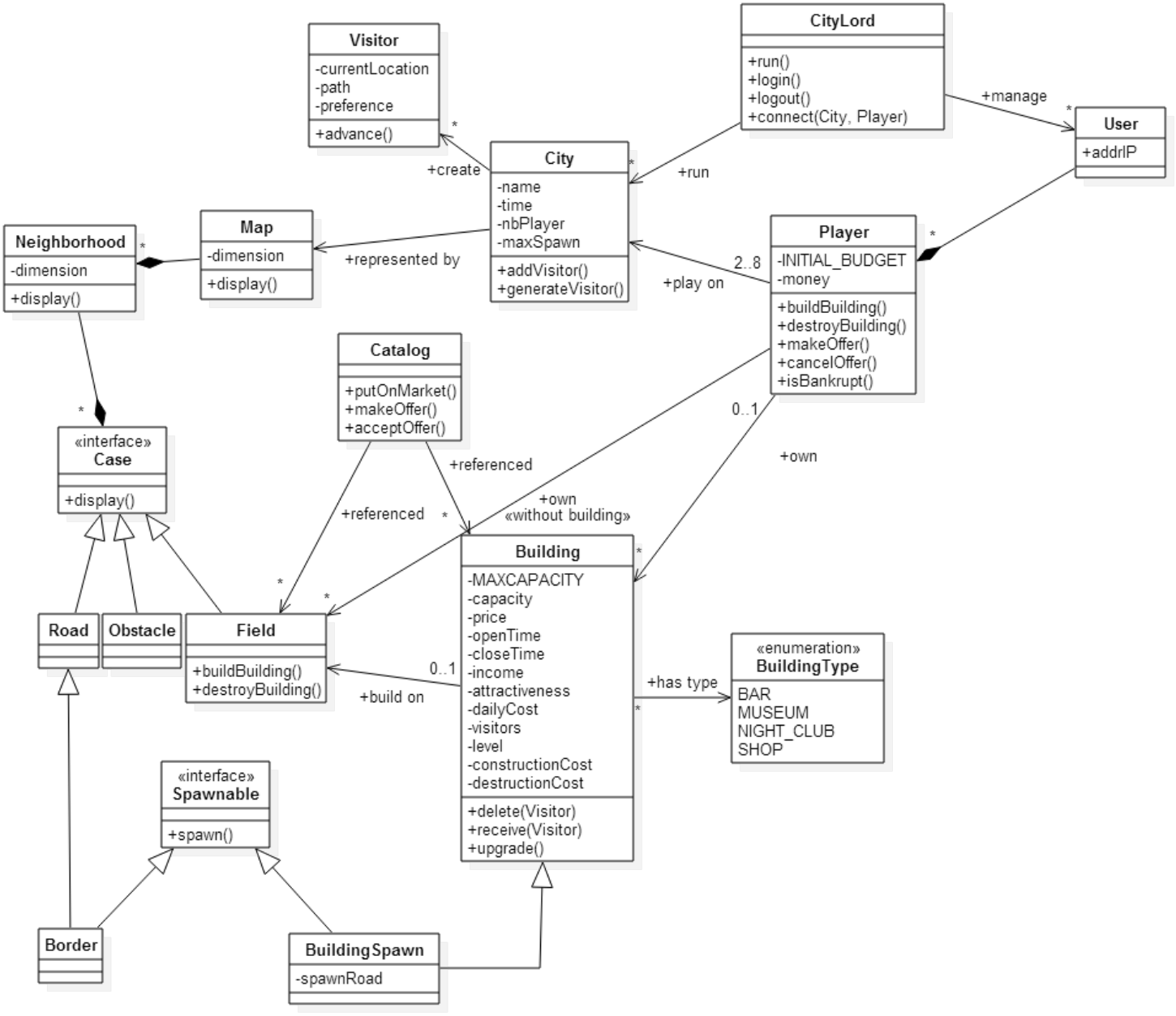

FIGURE 3.1 – Diagramme de classe : Base du système

Chapitre 4

Index des termes utilisés

- **Catalogue**
- **Map**
- **Point-and-click**
- **Pseudo**
- **Réseau informatique**
- **Serveur**
- **Système**