20/10/2014

Dossier des spécifications détaillées

Projet : Les trésors de Java

Crépin Aurélia, Levrat Gaël, Ribierre Paul, Sagon Nicolas, Veys Arthur

CPE Groupe E

#### Objet du document :

Ce dossier des spécifications détaillées sert à décrire l'ensemble des fonctionnalités du système et des spécifications générales afin d’assurer que le produit remplira les fonctions attendues par le client.

#### Responsabilité du document :

|  |  |  |
| --- | --- | --- |
| Rédacteurs | Vérificateurs | Validateur |
| CREPIN Aurélia - Expert Environnement et Qualité | LEVRAT Gaël - Expert  Modélisation | Comité de projet |
| LEVRAT Gaël - Expert  Modélisation | RIBIERRE Paul - Chef de Projet |  |
| RIBIERRE Paul - Chef de Projet  (Gestion et Management de projet) |  |  |
| SAGON Nicolas - Expert Programmation Objet |  |  |
| VEYS Arthur - Expert Technique et Environnement |  |  |

#### Historique des modifications et validation du document :

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Nature du  changement | Statut |
| V0 | 30/09/2014 | Création | Obsolète |
| V1 | 22/10/2014 | Publication | Applicable |

#### Diffusion du document :

|  |  |
| --- | --- |
| Entités | Noms |
| Comité de projet | CREPIN Aurélia  LEVRAT Gaël  RIBIERRE Paul  SAGON Nicolas  VEYS Arthur |
| Examinateur CPE | TROUILLOT Xavier |

#### État du document : Applicable

Table des matières

[1. Contexte, enjeux et objectif 4](#_Toc401863406)

[1.1. Objectif du Projet 4](#_Toc401863407)

[1.2. Rappels du cahier des charges du système 4](#_Toc401863408)

[1.2.1. Enjeux 4](#_Toc401863409)

[1.2.2. Besoins 4](#_Toc401863410)

[1.2.3. Solutions 4](#_Toc401863411)

[1.2.4. Contraintes 4](#_Toc401863412)

[1.2.5. Scénario retenu 5](#_Toc401863413)

[2. La définition des usages du système 6](#_Toc401863414)

[2.1. Définition des utilisateurs et de leurs rôles 6](#_Toc401863415)

[2.2. Définitions des cas d’usage du logiciel 6](#_Toc401863416)

[3. Périmètre fonctionnel 8](#_Toc401863417)

[3.1. La solution fonctionnelle et le besoin 8](#_Toc401863418)

[3.2. Décomposition du logiciel 8](#_Toc401863419)

[3.3. Charte graphique 8](#_Toc401863420)

[3.4. Ergonomie de l’écran de Gameplay 8](#_Toc401863421)

[3.5. Description de la cinématique d’usage du logiciel 9](#_Toc401863422)

[3.6. Description synthétique des Ecrans ou fonctionnalités 9](#_Toc401863423)

[3.6.1. Fonctionnalités détaillées de l’écran de Menu 9](#_Toc401863424)

[3.6.2. Fonctionnalités détaillées de l’écran de Gameplay 10](#_Toc401863425)

[4. Architecture applicative et données 11](#_Toc401863426)

[4.1. Définition du logiciel réalisé 11](#_Toc401863427)

[4.2. Architecture des logiciels 11](#_Toc401863428)

[5. Définition du périmètre technologique 12](#_Toc401863429)

[5.1. Principes techniques généraux et normes utilisées 12](#_Toc401863430)

[5.2. L’architecture applicative et technologique 12](#_Toc401863431)

[5.3. Infrastructure matérielle 12](#_Toc401863432)

[6. Description détaillée des modules, fonctions 13](#_Toc401863433)

[6.1. Menu Principal 13](#_Toc401863434)

[6.2. Menu réparation 14](#_Toc401863435)

[6.3. Déplacement 15](#_Toc401863436)

[6.4. Menu Combat 16](#_Toc401863437)

[6.5. Menu Fin de partie 17](#_Toc401863438)

[7. Performances et limites du système 18](#_Toc401863439)

[7.1. Limitations du cahier des charges 18](#_Toc401863440)

[7.2. Risques anticipés 18](#_Toc401863441)

# Contexte, enjeux et objectif

## Objectif du Projet

Le projet vise au développement d’un jeu numérique basé sur le jeu de plateau “Attention Pirates!”. Les attentes et contraintes liées à la mise en œuvre de ce logiciel sont définies dans le cahier des charges fourni par le maître d’ouvrage Mr Xavier TROUILLOT représentant la société CPE. Cette alternative numérique sera nommée “Trésors de Java”.

Ce document décrira les spécifications détaillées du système en accord avec le cahier des charges qui nous a été fournis.

## Rappels du cahier des charges du système

### Enjeux

Dans le cadre de la 4e année de formation, nous devons réaliser un logiciel reprenant les principes du jeu de société “Attention Pirates!”. Ce projet nous permettra de valider le module de gestion de projet.

### Besoins

Le besoin exprimé par le client est de pouvoir bénéficier d’une version électronique fonctionnelle du jeu de plateau « Attention Pirates ! ». Ce jeu consiste en un plateau découpé en hexagones représentant une mer avec plusieurs repaires. Chaque joueur possède un bateau avec des pirates et des canons. Le but est pour chaque joueur d’obtenir des trésors en pillant les repaires présents sur la carte et de les ramener au port. Les joueurs peuvent s’attaquer les uns aux autres et voler les trésors.

### Solutions

Nous allons utiliser les compétences enseignées, à la fois en cours de gestion de projet, génie logiciel et de programmation pour mener à bien ce projet. La solution sera un logiciel JAVA qui devra être exécutable sur les machines de CPE sous Linux/Ubuntu et sous Windows 7. Il fera l’objet d’une présentation et d’une évaluation par la maitrise d’ouvrage.

### Contraintes

#### Contraintes technologiques

L’utilisation de JAVA est obligatoire pour le développement logiciel. Il nous sera aussi nécessaire d’utiliser une bibliothèque graphique afin de rendre un aspect le plus abouti possible au logiciel. La deuxième contrainte est l’utilisation de libGDX comme Framework, pour lequel nous ne possédons pas suffisamment d’expérience.

Pour résoudre les difficultés liées aux compétences techniques, nous avons réalisé une grille de compétence pour déterminé les rôles dans le projet. Ainsi les personnes les plus compétentes dans un domaine seront en charge de celui-ci. De plus les personne possèdent des compétences formerons le reste de l’équipe dans leur domaine pour permettre d’avoir la totalité de l’équipe opérationnel au besoin et notamment ne pas souffrir de possible d’absentéisme.

#### Contraintes organisationnelles

La réalisation nécessite une équipe de développeurs de 5 personnes sur 12 demi-journées pour créer un logiciel répondant à l’intégralité des spécifications dans le temps impartis. Le travail devra être réparti uniformément entre les développeurs, de manière à optimiser cette phase. Nous utiliserons les systèmes Git et Redmine pour nous assister dans la gestion du projet.

#### Contraintes budgétaires

Nous possédons une capacité budgétaire limitée, représentée par des heures d’autonomie, qui nous est allouée. Tout dépassement sera à notre charge. Arthur Veys, membre du groupe projet, nous prête à titre gracieux, et pendant la durée du projet, un espace serveur dédié qui hébergera nos différents outils. Il est important de noter qu’aucun retard ne sera toléré par le client et que, sauf cas exceptionnel, celui-ci entrainera la non validation du module.

#### Contraintes humaines

Une des contraintes humaines est que nous n’avons pas choisi notre groupe de travail pour ce projet. Afin de maintenir une certaine cohésion du projet, nous utilisons des moyens de communications à notre disposition via réseaux sociaux et services Cloud.

Nous avons également voté de manière unanime le chef de projet en comité. De sorte que l’organisation et l’orientation du projet est décidé par le comité mais en cas de désaccord, le chef de projet est là pour le résoudre.

### Scénario retenu

La livraison du projet ce découpera ainsi en fonction de l’avancement :

* Lot1. Cœur du jeu:

Ce lot doit permettre le déroulement complet d’une partie. Il prend en compte toutes les règles du jeu et une interface non-graphique.

* Lot2. Adaptation graphique

Ce lot est destiné à ajouter une couche graphique sommaire afin de mieux visualiser le déroulement du jeu.

* Optionnel Lot3. Meilleure expérience utilisateur

Ce lot doit permettre une meilleure expérience utilisateur, en ajoutant une interface graphique complète.

* Optionnel Lot3. Jeu en réseau

Ce Lot permettra de jouer en réseau Local.

* Optionnel Lot4. Intelligence artificielle

Ce lot permettra de jouer contre un ou plusieurs joueurs gérés par l’ordinateur.

# La définition des usages du système

## Définition des utilisateurs et de leurs rôles

L’utilisateur primaire est le joueur. Il peut y avoir entre 2 et 6 joueurs pour une même partie. Chaque joueur accède successivement à l’interface de jeu et procède au déroulement de son tour. Il peut aussi créer une partie et définir le nombre de joueurs, ainsi que leurs noms et couleurs et sauvegarder la partie.

Actions possibles:

* Nouvelle partie
* Se déplacer
* Attaquer
* Réparer
* Sauvegarder
* Quitter

## Définitions des cas d’usage du logiciel

Avant toute chose une partie doit être créée, comportant au moins 2 joueurs. Lorsque la partie démarre chaque joueur suit ce déroulement pour jouer son tour:

* 1ere étape: Je peux attaquer, me déplacer ou passer l’étape.
* 2e étape: Je peux attaquer, réparer mon bateau ou passer l’étape.
* 3e étape: Fin du tour, c’est au joueur suivant de jouer

Ensuite les actions au sein d’un cas d’utilisation se déroulent successivement :

* Créer une partie : En tant que joueur, je veux créer une nouvelle partie afin de jouer
  + Fonction : Nouvelle partie : Je crée une nouvelle partie

Condition : aucune

Action : Je crée une partie

* + Fonction : Ajouter joueur : J’ajoute autant de joueurs que necessaire

Condition : Entre 2 et 6 joueurs

Action : J’ajoute des joueurs

* + Fonction : Débuter partie : Je lance la partie

Condition : Joueurs initialisés

Action : je lance une partie

* Attaque: En tant que joueur, je veux attaquer un repaire ou bateau ennemi afin de récupérer les coffres qu’il possède.
  + Fonction: Abordage: J’attaque un repaire ou bateau ennemi

Condition: J’ai au moins un pirate et au moins un canon

Action: Je lance autant de dés que j’ai de canons.

* + - Les chiffres sur les dés me donnent les numéros d’emplacement détruit chez mon adversaire.
  + Fonction: Riposte: Mon adversaire riposte

Condition: Il lui reste au moins un pirate et au moins un canon.

Action: Il lance autant de dés qu’il a de canons

* + - Les chiffres sur les dés donnent les numéros d’emplacement détruit sur mon bateau.
  + Fonction : Piller : Je vole les trésors de mon adversaire

Condition : Mon adversaire n’a plus au moins un canon et au moins un pirate

Action : Je vole les trésors de mon adversaire.

* Se déplacer : En tant que joueur, je veux me déplacer afin de trouver des trésors
  + Fonction: Se déplacer:

Condition: J’ai au moins un pirate.

Action: Je me déplace d’autant de cases que j’ai de pirates

* + Fonction: Débarquer:

Condition: J’ai un/des trésors à bord, et je suis arrivé au port

Action: Je débarque mes trésors et crédite mon compteur.

* Réparer: En tant que joueur, je répare mon bateau pour ajouter des pirates et/ou des canons
  + Fonction: Affréter: Je répare complètement mon bateau

Condition: Je suis au port

Action: Je peux modifier complètement mon équipage et armement en ajoutant autant de canons et pirates que possible.

* + Fonction: Petite Réparation

Condition: je suis en mer

Action: Je peux ajouter ou modifier 2 éléments (pirate et/ou canon) de mon bateau.

* Passer : En tant que joueur, je passe mon tour
  + Fonction Passer: Je passe à l’étape suivante(2)
* Sauvegarder: En tant que joueur, je veux sauvegarder la partie
  + Fonction: Sauvegarder

Action: Je sauvegarde la partie

* Quitter: En tant que joueur, je veux quitter la partie
  + Fonction: Quitter:

Action: Je quitte la partie

# Périmètre fonctionnel

## La solution fonctionnelle et le besoin

La solution va permettre de jouer à une version numérique du jeu « Attention, pirates ! » dans le respect de ses règles et de ses graphismes.

La solution permettra un jeu multi-joueurs, toutes les actions du jeu de société (se déplacer, attaquer, se réparer/réorganiser) et de sauvegarder sa partie pour la reprendre plus tard.

## Décomposition du logiciel

La solution se composera de trois écrans principaux :

* L’écran Menu, permettant de choisir le nombre de joueurs, le nom de chaque joueurs et de débuter la partie.
* L’écran Gameplay, où la partie se déroulera. Il se décompose en 3 parties
  + La partie actions où l’on sélectionnera l’action à jouer
  + La partie carte où l’on verra le plateau de jeu dans sa globalité
  + La partie statut où l’on verra l’état de notre bateau
* L’écran de fin de partie avec un affichage des scores des différents participants

L’écran de Gameplay se décompose en 3 modules : Attaquer, Se déplacer, et Réparer

## Charte graphique

Charte couleur (RGB) :

bleu.png        marron.png        sable.png

*98,137,255                204,130,58                 255,220,162*



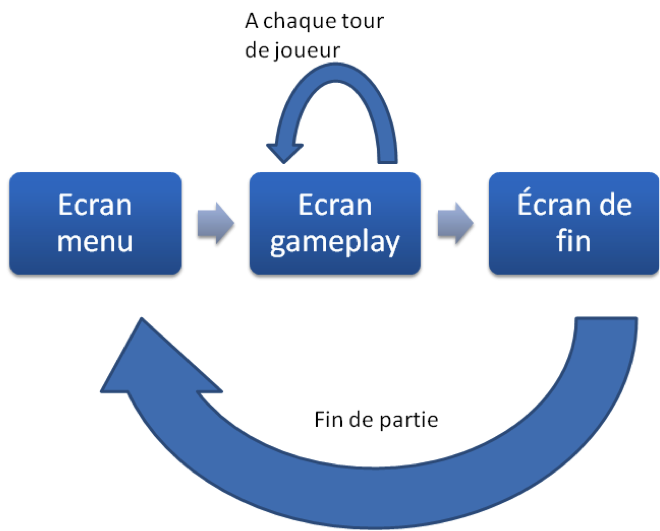
Exemple de texture : Bois

## Ergonomie de l’écran de Gameplay

Les boutons d’actions seront groupés en haut à droite de l’écran. Le statut du bateau en bas à droite (positionnement des pirates, canons et trésors).Le reste de l’écran sera constitué du plateau de jeu.

Les alertes paraitront en haut de l’écran.

## Description de la cinématique d’usage du logiciel



## Description synthétique des Ecrans ou fonctionnalités

### Fonctionnalités détaillées de l’écran de Menu

* Nom de l’écran : Menu

Entrée :

Sorties : liste des joueurs

Traitement mis en œuvre : sélection des joueurs pour la partie

Erreurs possible :

* + Nom des joueurs incorrect
  + Pas assez de joueurs
  + Trop de joueurs
* Nom de l’écran : Fin de partie

Entrée : Tableau des scores

Sorties :

Traitement mis en œuvre :

Erreurs possible :

* + Erreur dans les scores

### Fonctionnalités détaillées de l’écran de Gameplay

* Nom de l’écran : Se déplacer

Entrée : Le joueur à déplacer

Sorties : La nouvelle position du joueur

Traitement mis en œuvre : Déplacement du joueur

Erreurs possible :

* + La position initiale est incorrecte
  + La position finale ne respecte pas les règles
* Nom de l’écran : Attaquer

Entrée : La cible à attaquer

Sorties : Résultat de l’attaque

Traitement mis en œuvre : attaque d’un joueur ou d’un repaire

Erreurs possible :

* + La cible n’existe pas
  + La cible n’est pas à portée
  + Pas de canons pour attaquer
* Nom de l’écran : Réparer

Entrée : le bateau à réparer

Sorties : Le nouvel état du bateau

Traitement mis en œuvre : Ajout de canon ou pirate au bateau

Erreurs possible :

* + Il n’y a plus de place sur le bateau
  + Il n’y a pas assez de pirate ou de canon
* Nom de l’écran : Prendre butin

Entrée : Le repaire à dévaliser

Sorties : Le nouvel état du bateau

Traitement mis en œuvre : Ajout de coffre en remplacement de pirate ou canon

Erreurs possible :

* + Il n’y a plus de place sur le bateau
  + Il n’y a pas assez de pirate ou de canon
  + Il y a trop de coffres dans le bateau

# Architecture applicative et données

## Définition du logiciel réalisé

Ce logiciel sera développé avec le langage de programmation Java. Nous généreront donc un fichier “Les trésors de Java.jar”. L’extension de fichier “jar” est un fichier compressé qui permet de distribuer des classes Java. Il suffit ensuite de lancer ce fichier jar à l’aide de la console pour pouvoir exécuter le logiciel. Nous avons décidé de produire ce logiciel pour le système d’exploitation Linux.

Nous n’avons pas prévu d’utiliser de base de données, nous utiliserons la sérialisation pour la sauvegarde des données. Nous avons choisis cette solution car nous n’aurons pas beaucoup de données à sauvegarder et à gérer donc le déploiement d’une base de données prendrait beaucoup de temps et ne nous serait pas très utiles.

## Architecture des logiciels

Notre logiciel repose sur un jeu de société donc il doit respecter exactement toutes les règles de ce jeu. Ce point sera la plus grosse contrainte de ce logiciel. De plus nous allons utiliser une bibliothèque graphique “LibGDX”, ce qui va donc beaucoup influencer la conception de notre logiciel que nous allons réaliser en UML.

LibGDX va influencer l’architecture de notre logiciel car pour l’utiliser on doit suivre un cycle de vie précis de l’application :

Nous devons créer l'application puis ensuite passer par une phase de redimensionnement. Cette phase permet de définir la taille de la fenêtre de l’application. Ensuite on rentre dans la boucle principale qui recalculera la position et l’état graphique de tous les éléments affichés.

Quand le logiciel a fini son travail, il passe par une phase qui permet de le fermer proprement.

# Définition du périmètre technologique

## Principes techniques généraux et normes utilisées

Nous utiliserons les bonnes pratiques de développement et les patterns de conception afin de mener ce projet à bien.

Nous devrons aussi utiliser les normes en vigueur dans “LibGDX”, pour pouvoir utiliser cette bibliothèque il faudra utiliser les classes principales qu’elle nous fournit. Par exemple il faudra que l’on utilise la classe LwjglApplication, qui est à la base de tout développement avec “libGDX”. Cette classe permet ensuite d’utiliser tout le cycle de vie propre à “libGDX” que l’on a présenté plus haut.

## L’architecture applicative et technologique

Pour notre application, nous aurons très peu de pré-requis, pour le côté système d’exploitation : le logiciel doit pouvoir fonctionner normalement sur les machines de CPE. Ce qui nous impose le fonctionnement sur Linux (Ubuntu). La version 7 de Java au minimum devra être installée. Notre projet est développé en Java et nous utilisons une librairie graphique très puissante, donc celui-ci pourrait théoriquement tourner sur Windows et Android.

Ensuite nous n’aurons pas de pré-requis avec un système de gestion de base de données car nous avons fait le choix de ne pas utiliser de base de données pour notre logiciel. Le principal logiciel que nous aurons à mettre en œuvre est Eclipse pour nos développements. Il apporte beaucoup d’outils très puissants pour le développement car une très grande communauté l’utilise et il est basé sur un système de plugins qui permet d’ajouter des outils pour faciliter le développement ou bien pour gérer le travail en équipe. Eclipse ne prend pas en charge nativement les dépôts de fichier GIT mais nous allons pouvoir utiliser un plugin pour gérer notre code sur notre dépôt de fichier GIT. Eclipse permet aussi de pouvoir gérer directement les tests unitaires. De plus Eclipse est très utilisé dans les entreprises c’est donc un gage de qualité par rapport à ses concurrents.

## Infrastructure matérielle

Pour notre logiciel, l’infrastructure matérielle nécessaire sera très simple, il suffira d’avoir un ordinateur sous Ubuntu muni d’un écran, d’un clavier et d’une souris. La machine devra être suffisamment puissante et récente (2Ghz, 2Go de RAM) et posséder suffisamment d’espace disque pour accueillir le jeu (1Go d’espace libre). Pour notre premier lot aucune connexion réseau ne sera nécessaire mais pour les futurs lots si l’utilisateur souhaite jouer en multi joueur sur avec un amis en ligne, il faudra ajouter une contrainte matériel sur la connexion réseau.

# Description détaillée des modules, fonctions

Le jeu est représenté par la suite des maquettes ci-dessous. Chaque maquette  représente une interface du jeu. Une cinématique proposant les actions des interfaces sera décrite pour chaque maquette.

## Menu Principal

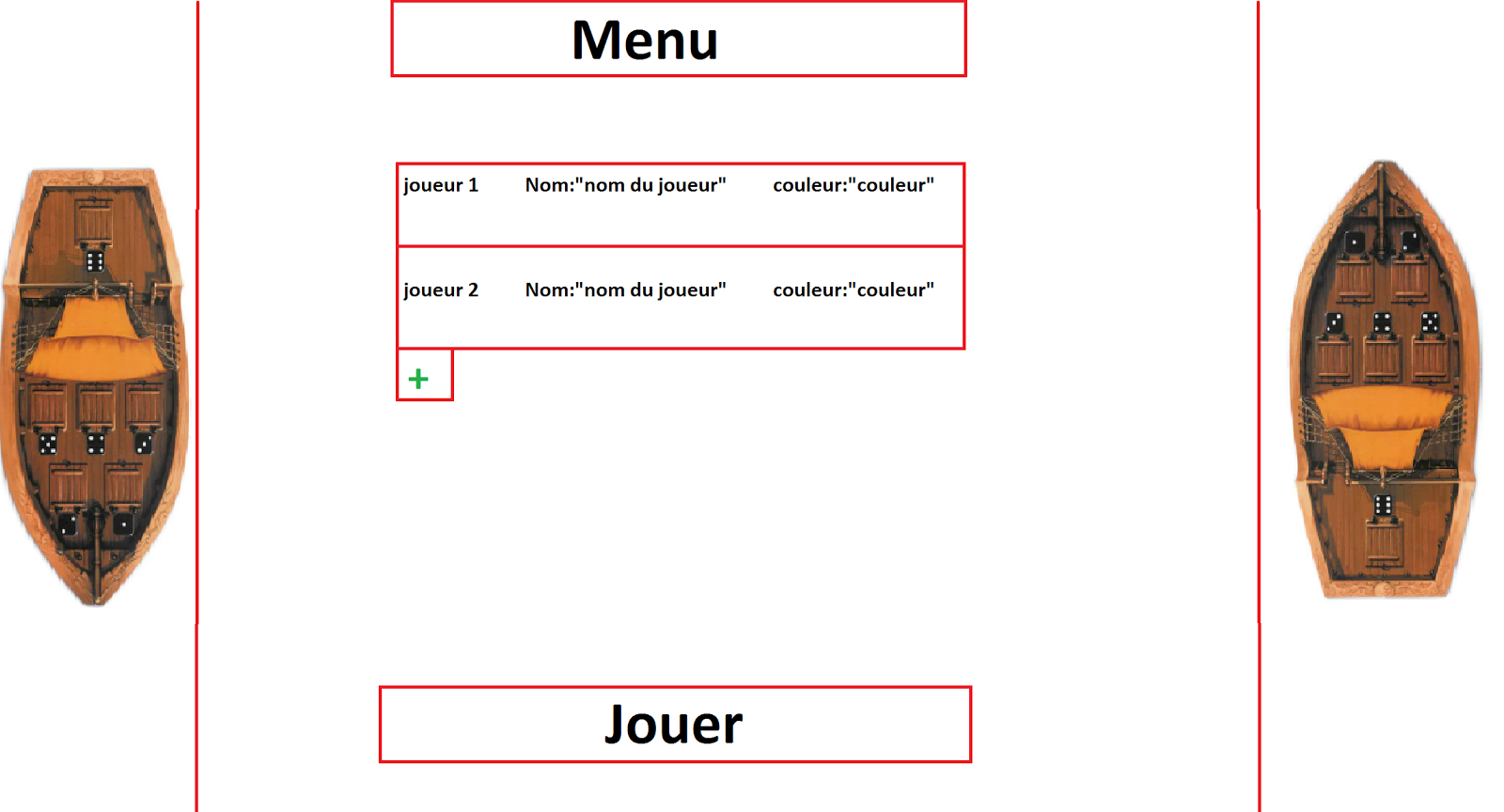


Figure : Maquette du menu principal

Le menu principal permet de définir le nom et la couleur de l’ensemble des joueurs. Le minimum de joueur est de deux et le maximum de six. Le « + » vert permet l’ajout de nouveau joueur. Un signe -sera disponible à partir de 3 joueurs pour en supprimer. Le bouton jouer lance le jeu.

## Menu réparation

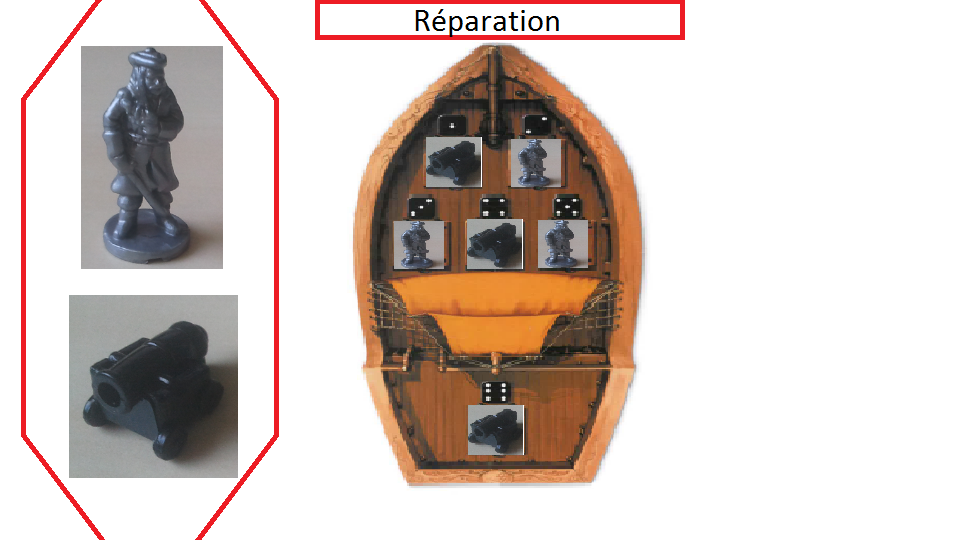


Figure : Maquette de l’interface de réparation.

La phase de réparation permet aux joueurs d’ajouter des canons ou pirates sur son bateau.

Elle est tout d’abord utilisée avant le premier tour de jeu pour que tous les joueurs puissent remplir leur bateau. Lorsqu’un bateau se trouve au port le joueur peut compléter ou modifier 6 emplacements. S’il se trouve en mer il peut n’en modifier que 2.

Pour ajouter de l’équipage sur son bateau, il suffit de sélectionner le canon ou le pirate sur la partie gauche de l’interface, puis de sélectionner l’emplacement sur le bateau.

Pour modifier l’emplacement de son équipage; il suffit de sélectionner deux emplacements du bateau et le(s) canon(s) et/ou pirate(s) seront permutés.

Un drag and drop pourra être ajouté dans le lot 3 pour améliorer l’expérience utilisateur.

## Déplacement

****

Figure : Maquette du plateau de jeu.

Le plateau du jeu permet d’effectuer les actions principales du jeu: Attaquer, se déplacer et réparer. Les actions attaquer et réparer ouvrent leurs propres interfaces (cf.: maquette de l’interface de combat contre un bateau/repère, maquette de l’interface de réparation).

La zone rouge en haut à gauche permet de sélectionner une action.

Quand on sélectionne l’action “attaquer”  les bateaux et repères disponibles pour une attaque sont affichés en dessous dans la zone rouge. Il suffit par la suite de sélectionner le bateau ou repère à attaquer et valider pour  aller à l’interface combat.

Quand on sélectionne l’action “réparer”, il nous suffit de valider pour aller à l’interface de réparation.

Quand on sélectionne l’action “se déplacer”, les cases mer accessibles s’illuminent en vert. Il suffit par la suite de cliquer sur une de ces cases pour s’y rendre.

Le plateau de jeu indique de plus le contenu de son bateau (en bas à droite de l’interface), le pseudo du joueur, le tour actuel et le nombre de trésors déjà acquis.

Un effet de loupe pour voir le contenu des bateaux sera ajouté pour améliorer l’expérience utilisateur en lot 3. Cet effet de loupe permettra aussi lors des phases d’attaque de choisir directement le bateau ou le repère sur la carte plutôt qu’il soit affiché dans le cadre rouge.

## Menu Combat

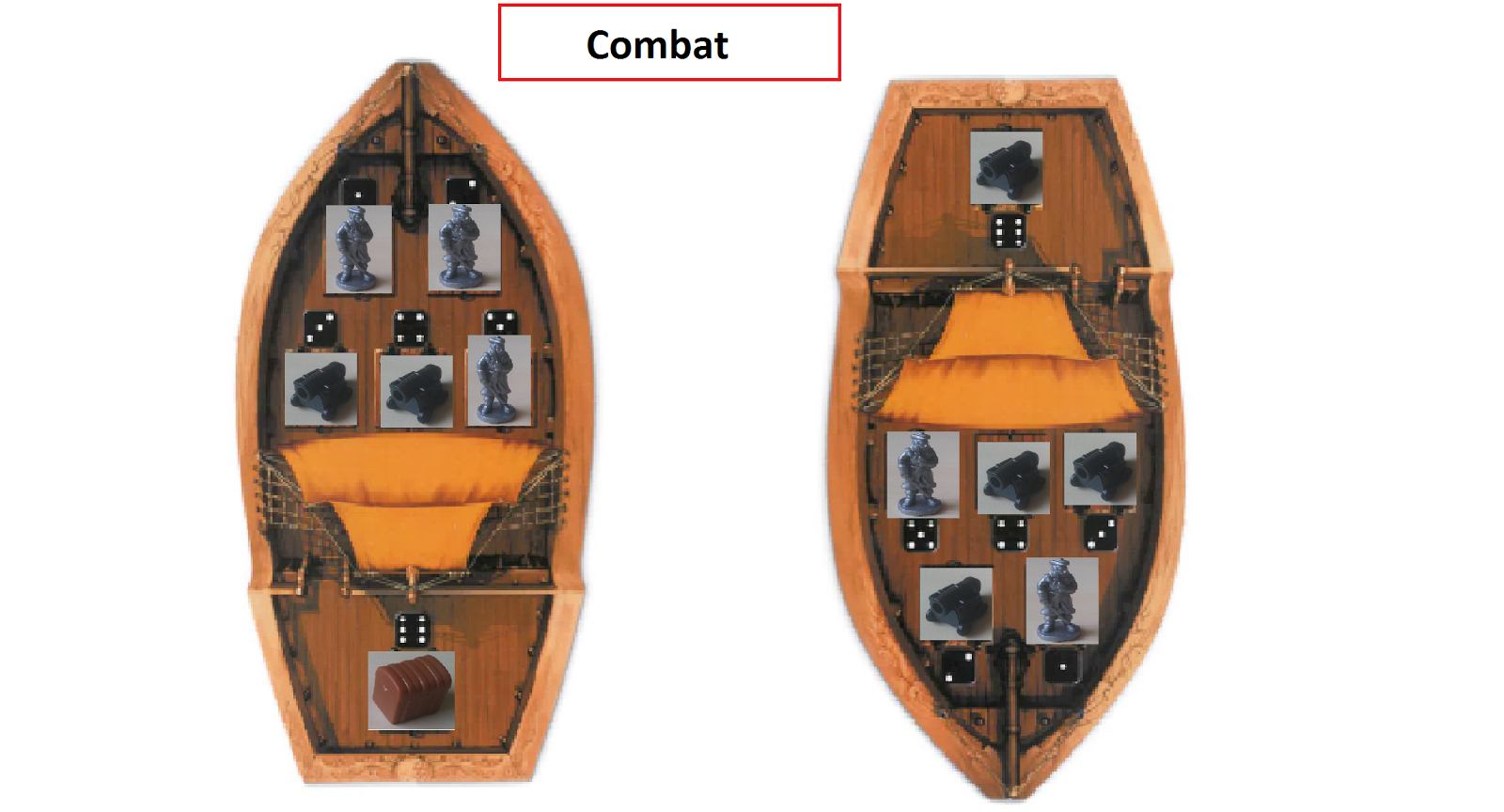


Figure : Maquette de l’interface de combat contre un bateau.

La phase de combat est simple. Des chiffres représentant les lancés de dés seront affichés. Ils indiquent quels sont les emplacements du bateau ennemi qui sont détruits. Une actualisation des emplacements du  bateau ennemi sera faite. Le bateau ennemi (s’il en est capable)  fera une riposte avec l’affichage de nouveaux chiffres et une actualisation des emplacements de votre bateau sera faite.

A la fin d’une phase de combat, si le bateau emporte 4 trésors ou plus, une phase de réparation spéciale est lancée. Dans cette phase, on pourra seulement ajouter les trésors sur un emplacement de son choix.

Un effet de lancer de dés sera mis en place pour améliorer l’expérience utilisateur avec le lot 3.

Des effets sonore et visuel peuvent aussi être ajoutés en lot 3.

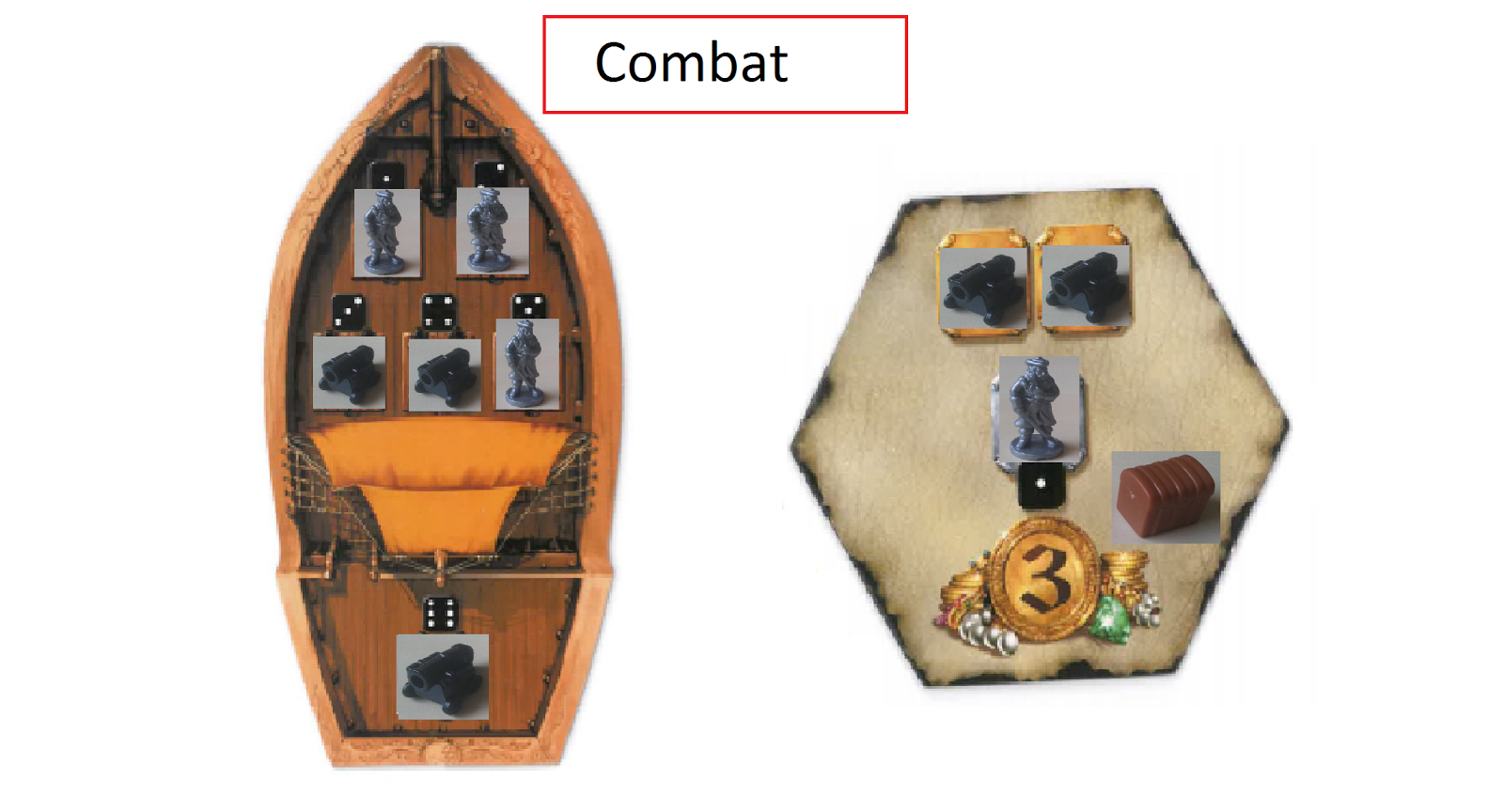


Figure : Maquette de l’interface de combat contre un repère.

## Menu Fin de partie

****

Figure : Maquette du menu de fin de partie.

Le menu de fin de partie affiche les statistiques générales de la partie : quel est le gagnant, quels sont les perdants, le nombre de trésors acquis par joueur, la durée de la partie

# Performances et limites du système

## Limitations du cahier des charges

Les fonctionnalités du jeu seront principalement limitées par notre budget ou temps de développement. Nous ne voulons pas exclure définitivement certaines fonctionnalités pouvant améliorer l’expérience utilisateur; c’est pour cela que le comité a préféré réaliser un système fonctionnel répondant au cahier des charges. Dans un second temps nous réaliserons des lots optionnels en fonction de notre temps restant.

Ainsi nous nous assurons de répondre au besoin du client, mais nous n’excluons pas la possibilité d’avoir un système plus user-friendly.

Nous avons fixé les limitations de notre système dans le but de  réaliser le jeu “attention pirates!” qui  démarrera sur un ordinateur avec 2 à 6 joueurs humains, avec une interface graphique simple et efficace.

## Risques anticipés

Les limitations et exclusions ne sont pas uniquement dues au cahier des charges, mais également aux risques que nous avons anticipés dans le PAQ. Un autre risque apparait à cette étape est la prise en main du Framework LIBGDX qui pourrait s’avérer complexe et consommatrice de temps.

Pour limiter cet impact, nous réaliserons en premier lieu une interface graphique basique, pour uniquement développer de meilleures interactions si le temps nous le permet.