

Projet « Qualité-Test »

Camille Constant

Cahier des charges

Monkeys

*Durée du projet : 40 heures.**Date de début du projet : 16 décembre 2015.**Date de fin du projet : 31 janvier 2016.*

Auteur	Contact
Camille Constant	camille.constant@eseo.fr

Relecteur	Contact
Matthias Brun	matthias.brun@eseo.fr

Version	Date	Commentaires
3.3	15/12/2015	Ajout de quelques précisions, notamment sur les livrables.
3.2	18/10/2015	Ajout des incréments.
3.1	16/10/2015	Modification nom option et date de fin de projet.
3.0	13/10/2015	Mise à jour pour projet 2015-2016 (les bananes deviennent du rhum).
2.0	13/10/2014	Mise à jour pour projet 2014-2015.
1.8	13/01/2014	Modifications du protocole.
1.7	17/12/2013	Ajout du protocole.
1.6	11/12/2013	Légères corrections.
1.5	04/12/2013	Introduction du concept de <i>piratophagie</i> . Modifications de l'initialisation de la position de départ du pirate.
1.4	03/12/2013	Simplification de l'énergie dans l'IHM.
1.3	29/11/2013	Correction dans les exigences de validation.
1.2	27/11/2013	Figure déploiement, nom, livrables (exigences), corrections générales.
1.1	26/11/2013	Description détaillée, IHM, corrections générales.
1.0	25/11/2013	Objectif, description générale, déploiement, description détaillée, livrables.

Table des matières

1	Introduction	4
1.1	Objectif du cahier des charges	4
1.2	Portée du cahier des charges	4
2	Description générale du projet	5
2.1	L'île	5
2.2	Les personnages	5
3	Déploiement	6
4	Protocole de communication	7
5	IHM	9
5.1	Interface graphique	9
5.1.1	Vue du jeu	9
5.2	Contrôles	9
6	Description détaillée de <i>Monkeys</i>	10
6.1	Connexion et déconnexion du jeu	10
6.2	Initialisation d'une partie	10
6.2.1	La carte	10
6.2.2	Configuration d'une partie	10
6.3	Déroulement d'une partie	10
6.3.1	Les éléments statiques	11
6.3.2	Les personnages	11
6.4	La fin de partie	11
7	Livrables	13
7.1	Incréments	13
7.2	Exigences qualité	13
7.3	Exigences de validation	13
7.4	Productions demandées	14
8	Définitions des termes et acronymes	15
8.1	Définitions	15
8.2	Acronymes	15

1 Introduction

1.1 Objectif du cahier des charges

L'objectif de ce document est de décrire :

- le système informatique à développer durant le projet,
- les productions à livrer (documents, codes sources).

1.2 Portée du cahier des charges

Ce document concerne tous les intervenants du projet.

2 Description générale du projet

L'objectif de ce projet est de réaliser, **en respectant les exigences qualité demandées**, et **valider** un premier prototype de jeu de plateau multi-joueurs, nommé *Monkeys*. Ce jeu consiste à être le premier pirate à découvrir un trésor caché sur une île peuplée de singes *piratophages* et d'éventuels autres pirates.

2.1 L'île

L'île aura la forme d'une grille pouvant contenir deux types de cases : les cases Mer, non accessibles à un personnage, et les cases Terre sur lesquelles les personnages peuvent se déplacer. Les cases Terre peuvent contenir :

- un trésor caché (un par île),
- une bouteille de rhum,
- un personnage.

2.2 Les personnages

Il existe trois types de personnages :

- les pirates : chaque pirate correspond à un joueur. Celui-ci a un niveau énergétique qui diminue lorsqu'il se déplace et augmente lorsqu'il boit du rhum ;
- les singes erratiques : ces singes se déplacent aléatoirement et mangent un pirate dès qu'ils en rencontrent un ;
- les singes chasseurs : ces singes cherchent à atteindre le pirate le plus proche pour le manger.

3 Déploiement

La figure 1 décrit le déploiement du serveur et des clients de *Monkeys*.

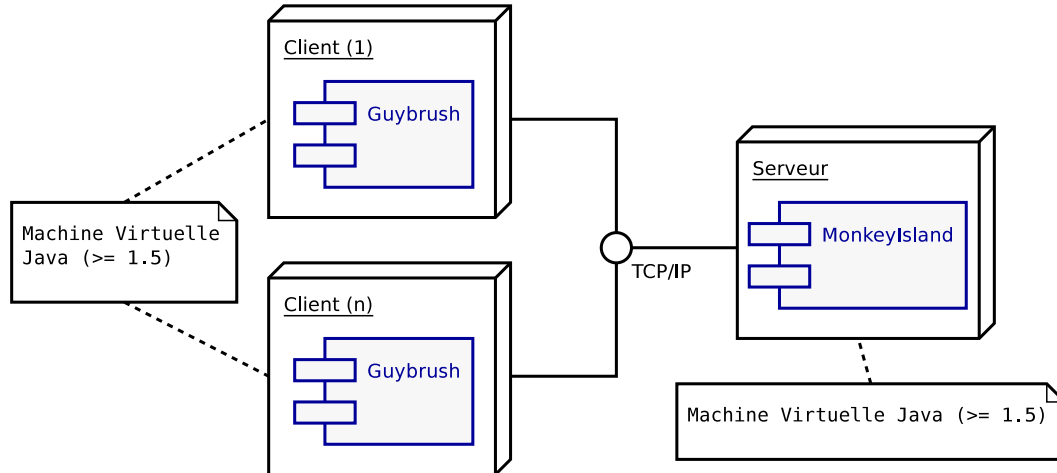


FIG. 1 – Déploiement de *Monkeys*.

Le jeu étant multi-joueurs, celui-ci se décompose en deux composants logiciels :

- MonkeyIsland : le serveur du jeu, gérant le déroulement du jeu décrit en section 6,
- Guybrush : le client du jeu, communiquant au serveur les déplacements des pirates.

Ces composants sont programmés en Java. Leur exécution nécessite une machine virtuelle Java (ou *Java Virtual Machine JVM*) en version supérieure ou égale à 1.5.

Les composants MonkeyIsland et Guybrush communiquent via un réseau TCP/IP (*Transmission Control Protocol / Internet Protocol*) décrit en section 4.

4 Protocole de communication

Sont présentés en figure 2 les messages pouvant être envoyés par le client *Guybrush* au serveur *MonkeyIsland* et en figure 3 les messages pouvant être envoyés par le serveur *MonkeyIsland* au client *Guybrush*.

Commande	Commentaire
/I	Demande d'inscription du pirate.
/D directionX directionY	Demande de déplacement du pirate dans la direction indiquée. directionX et directionY sont des entiers compris entre -1 et 1 sachant qu'une des deux directions est à 0 .

FIG. 2 – Protocole de communication du client *Guybrush* au serveur *MonkeyIsland*.

Lorsqu'un client *Guybrush* fait une demande d'inscription au serveur *MonkeyIsland*, la première commande que lui renvoie celui-ci est celle indiquant la carte de l'île.

Commande	Commentaire
/i id-posX-posY-energie	Indication de l'identifiant du pirate (le port de la socket côté serveur) avec sa position et son énergie initiales.
/R	Refus de déplacement du pirate.
/A posX-posY-energie	Acceptation de déplacement du pirate à la position indiquée avec l'énergie donnée.
/P id1-pos1X-pos1Y_____idn-posnX-posnY	Indication des autres pirates présents sur la carte avec leur position.
/n id-posX-posY	Indication d'un nouveau pirate à la position donnée.
/s id	Suppression du pirate indiqué.
/p id-posX-posY	Déplacement du pirate identifié aux positions indiquées.
/C l h $c_{11} - \dots - c_{1l} - \dots - c_{hl} - \dots - c_{hl}$	Indication de la carte avec sa largeur (l), sa hauteur (h) et des entiers compris entre 0 et 1 indiquant le contenu des cases (0 pour mer et 1 pour terre) (Exemple : /C 2 2 0-1-1-0).
/e pos1X-pos1Y_____posnX-posnY	Indication des positions des singes erratiques sur la carte.
/c pos1X-pos1Y_____posnX-posnY	Indication des positions des singes chasseurs sur la carte.
/B pos1X-pos1Y-vis1_____posnX-posnY-visn	Indication des positions des bouteilles de rhum sur la carte et de leur visibilité. L'identifiant des bouteilles de rhum correspond à sa position dans la liste. La visibilité vaut 1 si la bouteille est visible, 0 sinon.
/b id-visibilite	Indication de la visibilité de la bouteille indiquée.
/T posX-posY	Indication de la position du trésor lorsque celui-ci a été découvert.
/N	Indication d'une nouvelle partie.

FIG. 3 – Protocole de communication du serveur *MonkeyIsland* au client *Guybrush*.

5 IHM

L'interface graphique et les contrôles de *Monkeys* sont présentés dans cette section.

5.1 Interface graphique

5.1.1 Vue du jeu

Cette vue doit contenir la carte ainsi que tous les éléments apparaissant sur celle-ci (cf section 6). Tous les joueurs en ligne ont un même avatar de pirate. Le joueur distingue son propre avatar par une particularité physique. Le niveau énergétique du pirate du joueur est indiqué à côté de la carte.

5.2 Contrôles

Le joueur manipulera son pirate via les flèches du clavier.

Pour quitter le jeu, le joueur devra fermer la fenêtre du jeu.

6 Description détaillée de *Monkeys*

L'objectif est, dans un premier temps, de réaliser un prototype du jeu *Monkeys*, dont voici les contraintes et exigences.

6.1 Connexion et déconnexion du jeu

Afin de participer à une partie, un joueur doit se connecter, via le client Guybrush, au serveur MonkeyIsland.

6.2 Initialisation d'une partie

Le jeu se joue sur une carte. Celle-ci doit donc être initialisée et doit contenir chaque objet du jeu (cf. section 6.3).

6.2.1 La carte

La carte est en forme de grille composée de deux types de cases : la mer et la terre.

La mer. Ces cases représentent des cases non accessibles sur lesquelles aucun élément statique ni aucun personnage ne peut être placé.

La terre. Ces cases représentent l'île. Tout élément statique et tout personnage peut être placé sur ce type de case. Néanmoins, une case ne peut contenir qu'un personnage à la fois. Si la case contient un élément statique, celui-ci est « consommé » lorsqu'un pirate se déplace dessus.

6.2.2 Configuration d'une partie

Le prototype ne contiendra qu'un niveau décrit dans un fichier de configuration. Ce fichier de configuration permettra de déterminer :

- la taille de la grille,
- la forme de l'île (cases Terre et Mer),
- l'emplacement initial du trésor caché,
- l'emplacement de chacune des bouteilles de rhum,
- l'emplacement initial de chacun des singes,
- la valeur énergétique des bouteilles de rhum,
- le niveau énergétique maximum des pirates,
- le temps nécessaire à la réapparition d'une bouteille de rhum,
- la vitesse de déplacement des singes erratiques,
- la vitesse de déplacement des singes chasseurs.

6.3 Déroulement d'une partie

Au moins un joueur doit être connecté au serveur MonkeyIsland via le client Guybrush pour que la partie débute.

Le jeu se joue avec différents objets : les éléments statiques et les personnages. Les éléments statiques ont chacun des particularités décrites en section 6.3.1. Quant aux personnages du jeu, ils ont chacun une « vie » propre. Ils évoluent dans un temps continu, en parallèle. Leur description est faite en section 6.3.2.

6.3.1 Les éléments statiques

Le trésor. Le trésor est caché, il n'apparaît donc sur la carte que lorsqu'un joueur se déplace sur sa case. Sa découverte déclenche alors la victoire du joueur concerné.

Les bouteilles de rhum. Les bouteilles de rhum procurent une certaine valeur énergétique (toutes les bouteilles ont la même valeur énergétique) lorsqu'elles sont bues. Celles-ci sont bues quand un pirate se déplace sur une case en contenant. Les bouteilles réapparaissent au bout d'un certain temps. Si un pirate est sur la case de la bouteille au moment de sa réapparition, celle-ci est aussitôt bue.

6.3.2 Les personnages

Les personnages se déplacent horizontalement ou verticalement mais pas en diagonale.

Les singes erratiques. Les singes erratiques se déplacent à une certaine vitesse aléatoirement. Si un pirate se trouve sur la même case que lui, le pirate est mangé.

Les singes chasseurs. Les singes chasseurs se déplacent à une certaine vitesse vers le pirate le plus proche. Si un pirate se trouve sur la même case que lui, le pirate est mangé.

Les pirates. Chaque joueur manipule un pirate. Le pirate est initialisé aléatoirement sur une case vide de personnage et sans trésor. Le pirate peut être déplacé de case en case horizontalement ou verticalement mais pas en diagonale. Chaque déplacement lui coûte un point en énergie.

Afin de récupérer de l'énergie, un pirate boit des bouteilles de rhum. L'énergie du pirate augmente de la valeur énergétique de la bouteille bue sans dépasser sa valeur énergétique maximale.

Lorsqu'un pirate est mangé par un singe, son énergie est à 0. Le joueur a alors perdu.

6.4 La fin de partie

La victoire d'un joueur. En cas de victoire d'un des joueurs, le jeu se fige deux secondes avant de se réinitialiser pour commencer une nouvelle partie. Les joueurs connectés restent connectés.

La défaite d'un joueur. Lorsqu'un pirate n'a plus d'énergie, il revient à la vie à son emplacement initial à la prochaine partie s'il reste connecté.

Fin de partie. La partie est finie lorsqu'il n'y a plus aucun joueur connecté au jeu. Le jeu est donc réinitialisé et se met en attente d'une connexion d'un joueur.

7 Livrables

7.1 Incréments

Le projet étant conséquent, celui-ci sera décomposé en incréments. **Chaque incrément doit être validé avant de commencer l'incrément suivant.**

Incrément 1 : Cet incrément inclut les fonctionnalités suivantes :

- le déplacement des singes erratiques,
- la gestion des pirates (déplacement et décès),
- la gestion du trésor,
- la communication client-serveur.

Incrément 2 : Cet incrément inclut les fonctionnalités suivantes :

- la gestion du fichier de configuration,
- la gestion de l'énergie de chaque pirate,
- la gestion des bouteilles de rhum,
- la gestion des parties.

Incrément 3 : Cet incrément inclut les fonctionnalités suivantes :

- la gestion des singes chasseurs.
- ...

7.2 Exigences qualité

Un ou plusieurs audits qualité seront effectués durant le projet afin de vérifier que le code réalisé correspond à un certain niveau de qualité. Ce niveau sera atteint grâce à l'utilisation d'outils tels que Subversion, Checkstyle et Findbugs. L'audit qualité aura pour objectif de vérifier l'utilisation de ces outils au travers de la pertinence de leur configuration et de l'application des règles de qualité configurées.

7.3 Exigences de validation

Un ou plusieurs audits de validation seront effectués durant le projet afin de vérifier que tout code produit a été testé et si possible validé selon un certain niveau d'exigence. Ce niveau sera évalué au travers de la pertinence du plan de test et du cahier de tests décrits sous Testlink. La qualité des tests sera évaluée sur les tests réalisés. Les tests effectués seront des tests unitaires via JUnit (et EasyMock), des tests d'intégration et de charge via JMeter et des tests de validation. La couverture du code par les tests unitaires sera calculée par l'outil Emma.

7.4 Productions demandées

À la fin du projet, seront demandées les productions suivantes :

- le plan de test et cahier de tests (sous Testlink),
- la conception objet du projet (diagrammes de classes),
- le code Java dont la qualité répond aux exigences qualité décrites en section 7.2, ainsi que les fichiers de configuration de Checkstyle,
- le code de test dont la qualité répond aux exigences qualité décrites en section 7.2, ainsi que le plan de test (et de travail) de JMeter,
- un rapport de synthèse (au format PDF, pouvant être fait en Latex) comportant un résumé de ce qui a été réalisé et validé, un résumé de ce qui reste à faire, une éventuelle critique sur ce qui a été fait, un retour sur le projet.

8 Définitions des termes et acronymes

8.1 Définitions

- **piratophage** (*Terme métier (jargon)*) : désigne tout être particulièrement friand de chaire de pirate.
(à compléter)

8.2 Acronymes

(à compléter)