Plan du projet

# Introduction

# Présentation du projet

## Définition générale du projet

Le projet « Grenouille et crapaud » est un jeu où deux équipes (une équipe crapaud et l’autre celle des grenouilles) situé des deux extrémités du plateau de jeu vont s’affronter jusqu’à bloquer l’adversaire ou être arrivé à l’autre bout du plateau le premier. Il y pourra aussi jouer dans un casse-tête, le but du casse-tête est que le joueur pourra contrôler les deux équipes, le jeu se terminera quand tous les pions auront traversé le plateau ou quand des pions seront bloqués.

## Cahier des charges

Le but de ce projet est de développer un programme permettant de jouer au jeu et au casse-tête des grenouilles et des crapauds.

#### Description du jeu des grenouilles et des crapauds

Le jeu oppose deux joueurs et consiste à déplacer des grenouilles et des crapauds sur un plateau quadrillé de N lignes et M colonnes. En début de partie, les grenouilles sont attribuées à l'un des joueurs, les crapauds à l'autre. Le détenteur des grenouilles commence, et ensuite les joueurs jouent à tour de rôle. Le premier qui ne peut plus avancer ses batraciens a perdu.

Les grenouilles avancent horizontalement de la gauche vers la droite, et les crapauds de la droite vers la gauche.

Si la case adjacente dans le sens de déplacement est libre, la grenouille (respectivement le crapaud) déplacée vient occuper cette case. Si, par contre, elle est occupée par un crapaud (respectivement une grenouille) et que la case suivante est libre, le batracien vient occuper cette case libre. Dans tous les autres cas, le batracien ne peut être déplacé.

**Exemple avec un plateau de 1 \* 5:**

1. Plateau initial

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | **G** |  | *C* | *C* |

* 1. 2 3 4 5

1. La grenouille de la case 2 avance en 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** |  | **G** | *C* | *C* |

* 1. 2 3 4 5

1. Le crapaud de la case 4 avance en 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* | **G** |  | *C* |

* 1. 2 3 4 5

1. La grenouille de la case 3 avance en 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* |  | **G** | *C* |

* 1. 2 3 4 5

1. Le crapaud de la case 5 avance en 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* | *C* | **G** |  |

* 1. 2 3 4 5

1. La grenouille de la case 4 avance en 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G** | *C* | *C* |  | **G** |

* 1. 2 3 4 5

1. Le joueur qui détient les crapauds a perdu puisque ceux-ci ne peuvent plus avancer.

#### Description du casse-tête des grenouilles et des crapauds

Les règles de déplacement sont identiques à celles du jeu. Cependant les déplacements des grenouilles et des crapauds n'alternent pas obligatoirement. Le but de l'unique joueur est d'amener toutes les grenouilles à droite et tous les crapauds à gauche du plateau. Si cette disposition est atteinte, la partie de casse-tête est gagnée.

Exemple de disposition gagnante :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *C* | *C* |  | **G** | **G** |
|  |  | | |  |
| **G** | *C* | *C* |  | **G** |

Exemple de disposition perdante :

#### Application à développer

A son lancement l’application proposera à l’utilisateur d’afficher une aide pour son utilisation.

L’application devra permettre :

* À deux joueurs de s’affronter
* À un unique joueur de jouer contre l’ordinateur
* À un unique joueur d’utiliser le casse-tête
* De gérer des configurations initiales

Dans le cas d’une partie entre deux joueurs, ceux-ci devront saisir leur nom ce qui permettra ensuite à l’application de bien spécifier lequel des deux doit jouer. Les grenouilles et les crapauds seront attribués aux deux joueurs de manière aléatoire. En fin de partie, le nom du gagnant sera clairement affiché.

Dans le cas d’une partie contre l’ordinateur, ou bien lors de l’utilisation du casse-tête, l’application proposera à l’unique joueur d’entrer son nom. S’il n’en donne aucun, il sera désigné par un « joueur humain ». En fin de partie, un message indiquera le résultat de celle-ci et spécifiera le nom de l’unique joueur.

Après une partie, l’application affichera à nouveau le menu principal, ce qui permettra à l’unique joueur ou aux deux joueurs de lancer, éventuellement, une nouvelle partie.

#### Configurations initiales

Une **configuration** est la description d'un plateau dans un état précis. Elle indique donc la taille du plateau (nombre de lignes et de colonnes) et la disposition des grenouilles et des crapauds présents sur le plateau. La taille du plateau aura une limite supérieure à fixer (liée aux contraintes d’affichage).

Pour spécifier la configuration initiale d’une partie, l'utilisateur choisira l'une des deux alternatives suivantes :

* Entrer au clavier cette configuration.
* Sélectionner une configuration préétablie. Par conséquent, le programme devra permettre de gérer un ensemble de configurations préétablies que l'utilisateur pourra modifier à son gré par insertion ou retrait d'une configuration. En particulier, lorsque l'utilisateur aura saisi une configuration initiale, le programme lui demandera s'il souhaite l'inclure dans l'ensemble des configurations préétablies

Le placement des batraciens sur le plateau est libre. Toutefois, l’application vérifiera qu’il y a autant de batraciens de chaque catégorie. Si ce n’est pas le cas, la configuration sera considérée comme invalide.

Exemple de configuration initiale :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **G** |  |  | *C* |  | *C* |
|  | *C* | **G** |  | **G** | **G** |  |
| **G** | **G** |  |  | *C* | *C* |  |
|  | **G** | *C* | **G** |  | *C* | ***G*** |
|  | **G** | *C* |  | *C* | *C* |  |

Les configurations préétablies devront être stockées en mémoire permanente afin d’être retrouvées lors du prochain lancement de l’application.

#### Remarque

L’application fonctionnera en mode console ou bien sera dotée d’une interface graphique, au choix du groupe réalisant le projet.

## Définitions et acronymes

## Charte du projet

### Objectif du projet

L’objectif de ce projet sera la réalisation d’un jeu vidéo « Grenouille et crapaud » avec une interface.

### Périmètre du projet

Les périmètres du projet sont :

* Gestion des Pions et des plateaux.
* Sauvegarde des données du jeu dans un plateau.
* Création d’une IA.
* Création d’une interface.

### Demandes hors périmètre

Les demandes hors périmètre sont :

* Création d’un compteur pour le nombre de déplacement dans le casse-tête.
* Création de plusieurs niveaux d’IA.

### Principaux livrable

1er Livrable :

Premier aperçu d’un jeu fonctionnel permettant de jouer contre un joueur mais aussi en casse-tête.

Premier aperçu de l’interface fonctionnelle mais qui n’est pas lier avec le code.

2eme Livrable :

Premier aperçu du jeu et de l’interface ensemble.

Possibilité de jouer contre une IA très facile à battre.

Possibilité de sauvegarder une configuration

Apparition d’un nombre de coups dans le casse-tête

### Cadre : coûts, délai, qualité

### Acteur du projet : équipe MOE, MOA, contrôle qualité

Les différentes équipes dans le projet :

L’équipe MOE :

* Mael Le folgoc pontis
* Piere marie Combalbert (Gestionnaire de configuration)
* Yoan Guiraud (secrétaire)
* Yanis Delmas (Chef de projet)

L’équipe MOA :

* Corinne Servieres

Contrôle qualité :

* Mael Le folgoc pontis
* Yanis Delmas

### Autre moyens et ressources

### Condition d’acceptation : exigences, contraintes, qualité, critères et mesures

### Principaux risque identifiés et politique de gestion des risques

#### Risque identifier

Les risques que nous avons identifiés :

* Des javadocs mal écrite ou oublié d’être remplis.
* Erreurs dans les conventions java.
* Erreur simple dans le code (oublie de vider le tampon, etc.).
* Mauvaise compréhension du code à programmer par les programmeurs.

#### Politique de gestion des risques

* Création automatique de « TODO Commentaire pas terminer » lors de la création d’un commentaire par Eclipse.
* Relecture du code par l’équipe du contrôle de qualité et vérification des conventions par Eclipse.
* Vérification par les tests et vérification usuel par l’équipe du contrôle de qualité (Test durant une partie de jeu pour découvrir des erreurs ou voir le bon fonctionnement du jeu).
* Réunion organisée pour éviter de mauvaise compréhension et réorganiser les idées du groupe.

## Etude générale du besoin

# Organisation du projet

## Présentation du cycle de vie et ses conséquences

## Identifier les rôles

## Plan de communication

### Localisation géographique des intervenants

### Moyens de communication utilisés

### Réunion projet MOE

### Comité de pilotage

## Assurance qualité

### Normes et standard de travail

### Manuel qualité et démarche qualité

## Ressource matérielle et logicielles

# Pilotage du projet

## Cycle de vie

## Planification prévisionnel initiale

### Ordonnancement des principales itérations/taches

### Identification des principaux jalons

### Calendrier prévisionnelle

## Organisation des réunions et comités de pilotage

## Suivi du projet

### Suivi d’avancement et mesure des écarts par rapport au prévisionnel

### Synthèse par tableau de bord

### Résultat des tests et recette de prototype de la période

### Résultats des test et recette de prototype de la période

### Identification des principaux écarts et problèmes constaté, solutions possibles

### Proposition de modification de la planification prévisionnelle pour tenir compte des corrections à apporter

### Comptes-rendus des réunions projets de la période

### Compte-rendu du comité de pilotage de la période

### Planification prévisionnelle révisée pour les périodes suivantes

# Bilan projet

# Glossaire

# Annexes