**PFA**

**Cahier des charges**

**Sommaire**

**I) Introduction**

**II) Description de la demande**

1. **Les objectifs**
2. **Produit du projet**
3. **Les fonctions du produit**

**a/ L’interface graphique**

**b/ Les événements**

**c/ Le binaire du jeu**

**III) Contraintes**

1. **Contraintes ergonomiques**
2. **Contraintes des évènements**
3. **James !**

**IV) Déroulement du projet**

1. **Planification**

**I) Introduction**

A l'issue du module PFA (Projet de Fin d'Année), notre groupe a décidé de créer, un logiciel de création de jeux. En effet notre projet s'inspire du célèbre RPG Maker qui est un parfait exemple pour illustrer ce type d'outil.

Par ailleurs contrairement à RPG Maker qui se veut spécialiste comme son nom l'indique en création de RPG (Roleplaying Game), notre logiciel se focalisera sur les jeux de plate-forme 2d de type exploration.

**II) Description de la demande**

*Définir les résultats que le projet doit atteindre*

**1- Les objectifs**

L'objectif final de TG Maker est la création d'un jeu vidéo, via une interface simple et ergonomique.

Ce jeux vidéo sera compatible sur toute plateforme Unix et le jeu généré sur toute plateforme remplissant les prérequis.

**2- Produit du projet**

TG Maker est un logiciel qui permet de créer des jeux. L'utilisation de TG Maker se veut simple, notamment grâce à une interface intuitive, et évite aux utilisateurs d'avoir recours à des techniques plus avancées telles que l'utilisation de langages informatiques. Ils n'ont donc pas besoin de savoir programmer pour créer des jeux, mais juste d'avoir de l'imagination (compétence essentielle pour ce type de jeu).

**3- Les fonctions du produit**

Les principales fonctionnalités de TG Maker sont :

**a**/ L’interface graphique

Celle - ci permet à l'utilisateur de créer son propre jeu vidéo sous forme de projet, qu'il peut : sauvegarder, éditer, tester ou générer si celui-ci est terminé.

L'utilisateur pourra, à l'aide d'un éditeur, créer une carte simple ou complexe au format tile, à l'aide de tile définis par défaut, ou de ses propres tiles. Divers outils comme le pinceau (une case), le rectangle de sélection (plusieurs cases) ou le pot de peinture (toutes les cases) seront disponibles. D'autres outils permettront à l'utilisateur de typer ou de customiser sa carte comme il le souhaite.

Le gestionnaire de ressources offre la possibilité à l'utilisateur d'importer ses propres tiles, images, sons et de les visualiser à l'aide d'un viewer ou écouter à l'aide d'un player. L'utilisateur pourra importer ses ressources aux formats attendu par TG Maker.

**b**/ Les évènements

L'utilisateur peut définir une liste d'évènements, sorte de pseudo-code, et y associer des cases de la carte pour ainsi en déclencher l'exécution lors du passage du joueur. Ces évènements possèdent également un délai avant activation ainsi qu'un nombre d'exécutions disponibles.

Le système d'évènements se veut générique et complet dans son fonctionnement. Il est donc probable que celui-ci demande un temps d'adaptation et de réflexion de l'utilisateur. C'est pourquoi seront disponibles plusieurs tutoriels pour les situations basiques des jeux.

Pour construire ces évènements, l'utilisateur dispose d'une liste d'entiers sur lesquels tester des conditions ainsi qu'une liste d'actions pouvant être effectuées.

Voici ce qui est à disposition de l'utilisateur pour la création d'évènements:

Structures de contrôle:

if

ifnot

else

while

Opérateurs binaires:

==

<

>

<=

>=

Entiers:

l'abscisse du joueur\*

l'ordonnée du joueur\*

la direction du joueur\*

la collision d'une case\*

l'état d'une case\*

le délai avant activation d'un événement\*

le compteur d'un événement\*

la possession d'un objet par le joueur

la pression d'une touche du clavier

Actions:

modification des entiers suivis d'une astérisque ('\*')

ajout d'un objet pour le joueur

suppression d'un objet du joueur

affichage de texte à l'écran

suppression du texte à l'écran

appel d'un autre événement

lancement d'un son

arrêt d'un son

c / le binaire de jeu

La sortie finale du programme, accessible grâce à l'interface graphique est une archive zip contenant deux fichiers:  
-un fichier texte de type 'Lisez-moi' nommé README.txt  
-un exécutable auto-extractible écrit en script shell (TITRE\_DU\_JEU\_\_install.sh) dont le rôle est de se compiler lui-même et de générer un unique exécutable de type variable selon la machine (TITRE\_DU\_JEU(.exe)).  
  
L'installation est compatible avec tout système pouvant exécuter les programmes suivants: sh, gcc, zip, base64, ainsi que des programmes UNIX de base forcement inclus avec ceux cités précédemment.  
La compilation du jeu se faisant sur la machine de l'utilisateur final, tous les systèmes pouvant exécutés les programmes cités précédemment sont compatibles. Cela peut même inclure Microsoft(TM) Windows(TM) à condition d'installer un utilitaire comme MinGW (procédure décrite en appendice dans le fichier README.txt).  
L'exécutable final, permettant de jouer au jeu, est totalement standalone au niveau des ressources. Il ne requiert que la bibliothèque SFML version 1.6 installée sur la machine pour pouvoir fonctionner.  
Un mécanisme de sauvegarde est inclus (jusqu'à trois sauvegardes par jeu, un fichier externe TITRE\_DU\_JEU\_saveN.sav généré pour chaque sauvegarde). Les possibilités de sauvegarde dépendent du jeu.  
  
La fenêtre de jeu est de résolution 800\*600, avec possibilité de redimensionnement (qui provoquera un étirement de l'image).

**III) Contraintes**

**1- Contraintes ergonomiques**

L'une des principales contraintes de l'interface graphique est l'ergonomie. Même s'il est vrai que le développement de celle-ci implique de nombreuses notions en développement, la difficulté majeur de cette interface et de la rende simple et efficace. L'un des aspects fondamentaux de TG Maker est de permettre à un utilisateur ayant peu de notions en programmation de créer son propre jeu.

**2- Contraintes des évènements**

La principale difficulté de la gestion des évènements est la traduction en code, puis dans le jeu généré, du pseudo-code de l’utilisateur, ainsi que sa sérialisation pour sauvegarder des projets depuis l’interface ou des parties depuis le jeu même.

**3- Contraintes de la génération**

La création du binaire final implique l’interprétation et la génération de code C++ à partir d’une classe qui contient toutes les informations du projet. Il nécessitera donc un convertisseur de type ‘bin2c’ prenant en paramètre un fichier binaire et donnant du code C++.

**IV) Déroulement du projet**

**1- Planification**

Le développement de TG Maker est composé de 3 parties distinctes qui sont :

- L'interface graphique

- Les évènements

- La génération du binaire

Afin d'optimiser le développement de TG Maker mais aussi de répondre aux attentes de chacun, chaque membre du groupes s'est attribué la partie qui lui correspondait le mieux en terme de capacité, méthode de travail et répartition du travail.

Ainsi Roger et David s'occuperont de l'interface graphique, Nathan et Benoit des évènements et James de la génération du binaire.

Les 3 groupes définiront ensemble les structures communes du projet pour ensuite développer le logiciel de manière autonome.