

Eurêka !

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs.....	3
1.3	Planification initiale	5
2	Analyse / Conception.....	6
2.1	Concept	6
2.1.1	Uses cases et scénarios	6
2.1.2	Diagramme de flux.....	7
2.2	Stratégie de test.....	11
2.2.1	Test Unitaire	11
2.2.2	Tests d'acceptation.....	11
2.3	Risques techniques	11
2.4	Planification	11
2.5	Dossier de conception	11
3	Réalisation.....	12
3.1	Dossier de réalisation	12
3.2	Description des tests effectués	12
3.3	Erreurs restantes	12
3.4	Liste des documents fournis	12
4	Conclusions	13
5	Annexes.....	14
5.1	Résumé du rapport du TPI / version succincte de la documentation	14
5.2	Sources – Bibliographie.....	14
5.3	Journal de travail	14
5.4	Manuel d'Installation	14
5.5	Manuel d'Utilisation.....	14
5.6	Archives du projet.....	14

NOTE L'INTENTION DES UTILISATEURS DE CE CANEVAS:

Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.

De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.

1 Analyse préliminaire

1.1 Introduction

Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.

Ces éléments peuvent être repris des spécifications de départ.

Eurêka est une serious game de résolution de puzzle, créer avec le langage C# et le moteur graphique unity, il a pour but de proposer au joueur la possibilité de résoudre des puzzles sous la forme d'un schéma avec des portes logiques générées aléatoirement, le joueur devra deviner le résultat de tous les résultats afin de réussir le puzzle, plusieurs niveaux de difficulté s'offre à lui, il peut aussi créer son propre puzzle.

Il a aussi la possibilité de prendre en photo son puzzle, afin de le sauvegarder sur le disque.

1.2 Matériel à disposition

Liste de matériel physique et de logiciel mis à disposition

- 1 PC du CPNV
- Unity 2020
- Visual Studio 2020
- Visual Studio Code
- Suite office

1.3 Prérequis

- Formation de base du CPNV
- Connaissances en POO
- Maîtrise de Unity

1.4 Objectifs

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

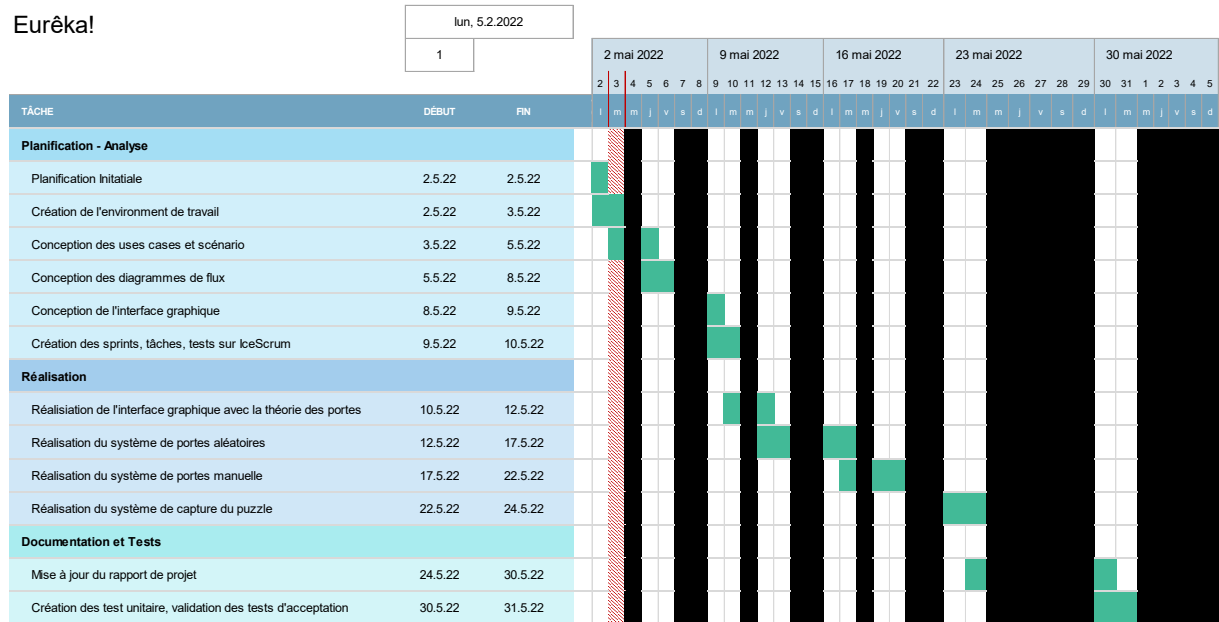
Les objectifs du projet étant :

- Résolution de puzzle
 - Le but étant que le joueur puisse résoudre le puzzle en devinant la sortie(résultat) du puzzle (Exemple : une porte AND avec en entrée 1 et 0, donne 0 comme sortie)

- **Puzzle Aléatoire**
 - Création Aléatoire des puzzles, chaque porte est générée aléatoirement, un nombre fixe de porte est présent, qui change en fonction du niveau de difficulté choisi par le joueur : en mode facile il y'a 3 portes logique, en mode moyen, il y'a 9 portes logique, en mode difficile, il y'a 19 portes logique.
- **Puzzle manuel**
 - Création manuel des puzzles, avec la possibilité de choisir les logiques portes dans le jeu, comme pour le puzzle aléatoire, les portes sont déjà prédéfinie par la difficulté, ils ne peuvent être déplacée, par contre le joueur peut choisir quelle type de porte (AND, XOR, etc...), et par la suite, de compléter le puzzle par la suite, comme un puzzle aléatoire
- **Capture de puzzle**
 - Permet de prendre en photo le puzzle courant, qui est sauvegarder dans sur le disque
- **Théorie des portes logiques**
 - Affiche sur l'écran comment les portes logique courante sur le puzzle courant du comment fonctionne, leur entrée, sortie, avec une table de vérité

1.5 Planification initiale

Eurêka!



2 Analyse / Conception

2.1 Concept

Le concept complet avec toutes ses annexes:

Par exemple :

- *Multimédia: carte de site, maquettes papier, story board préliminaire, ...*
- *Bases de données: interfaces graphiques, modèle conceptuel.*
- *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle...*
- ...

2.1.1 Uses cases et scénarios

Afin de comprendre comment, chaque fonctionnalité doit être programmé, et aussi d'avoir un scénario type pour gérer les exceptions j'ai fait des uses cases et scénario, afin de pouvoir bien se mettre d'accord sur ce qu'il faut faire

2.1.1.1 Puzzle aléatoire

Nom	PuzzleAléatoire-01	
en tant que	Joueur	
Je veux	Avoir une puzzle aléatoire à chaque partie	
Pour	finir le puzzle	
Priorité	M	
Action	Condition	Réaction
Lancement du jeu	Le joueur choisi un puzzle aléatoire	Un puzzle aléatoire est généré
La partie est en échec		la partie ne peux pas se terminer
La partie est gagnée		le jeu affiche les resultat, temps, et termine la partie
La partie courante est terminée		Un puzzle aléatoire est généré
Fermeture du jeu	la partie est terminée	le jeu ce ferme
Fermeture du jeu	une partie est en cours	le jeu demande une confirmation avant de ce fermer

2.1.1.2 Puzzle Manuel

Nom	PuzzleManuel-01	
en tant que	Joueur	
Je veux	Avoir une puzzle manuel à chaque partie	
Pour	finir le puzzle	
Priorité	M	
Action	Condition	Réaction
Lancement du jeu	Le joueur choisi un Manuel	Un puzzle Manuel est généré
Le joueur choisit glisse une porte de la liste		la porte est placé dans le jeu
Le joueur choisit glisse une porte de la liste	il relache le bouton prématurément	l'action est annulé
le joueur relie les portes		un fil est créer entre les portes
le joueur relie les portes	une fil entrave un autre fil	le fil n'est pas créer
La partie est en échec		la partie ne peux pas se terminer
La partie courante est terminée		Un puzzle Manuel est généré
Fermeture du jeu	la partie est terminée	le jeu ce ferme
Fermeture du jeu	une partie est en cours	le jeu demande une confirmation avant de ce fermer

2.1.1.3 Résolution de puzzle

Nom	RésolutionPuzzle-01	
en tant que	Joueur	
Je veux	Avoir une puzzle	
Pour	finir le puzzle	
Priorité	M	
Action	Condition	Réaction
Le joueur entre une résultat	le résultat est faux	une erreur est affichée
Le joueur entre une résultat	le résultat est juste	le jeu se termine en afficher le temps passé sur le puzzle
Le joueur n'entre pas résultat		rien ne se passe
Le joueur entre une résultat	la valeur est supérieur à un	le résultat est considéré comme faux
Le joueur entre une résultat	la valeur est inférieur à un	le résultat est considéré comme faux
Le joueur entre une résultat	la valeur n'est pas en décimal	le résultat est considéré comme faux

2.1.1.4 Capture d'écran

Nom	CaptureD'écran-01	
en tant que	Joueur	
Je veux	pouvoir avoir en image le puzzle	
Pour	l'observer	
Priorité	M	
Action	Condition	Réaction
Le joueur prends en photo le puzzle		la photo est sauvegardée dans le disque
Le joueur prends en photo le puzzle	les résultat de la partie est affiché	le bouton n'est pas visible ni actif
Le joueur prends en photo le puzzle	le chemin d'accès est inaccessible	un message d'erreur est retourné au joueur

2.1.1.5 Théories des portes logiques

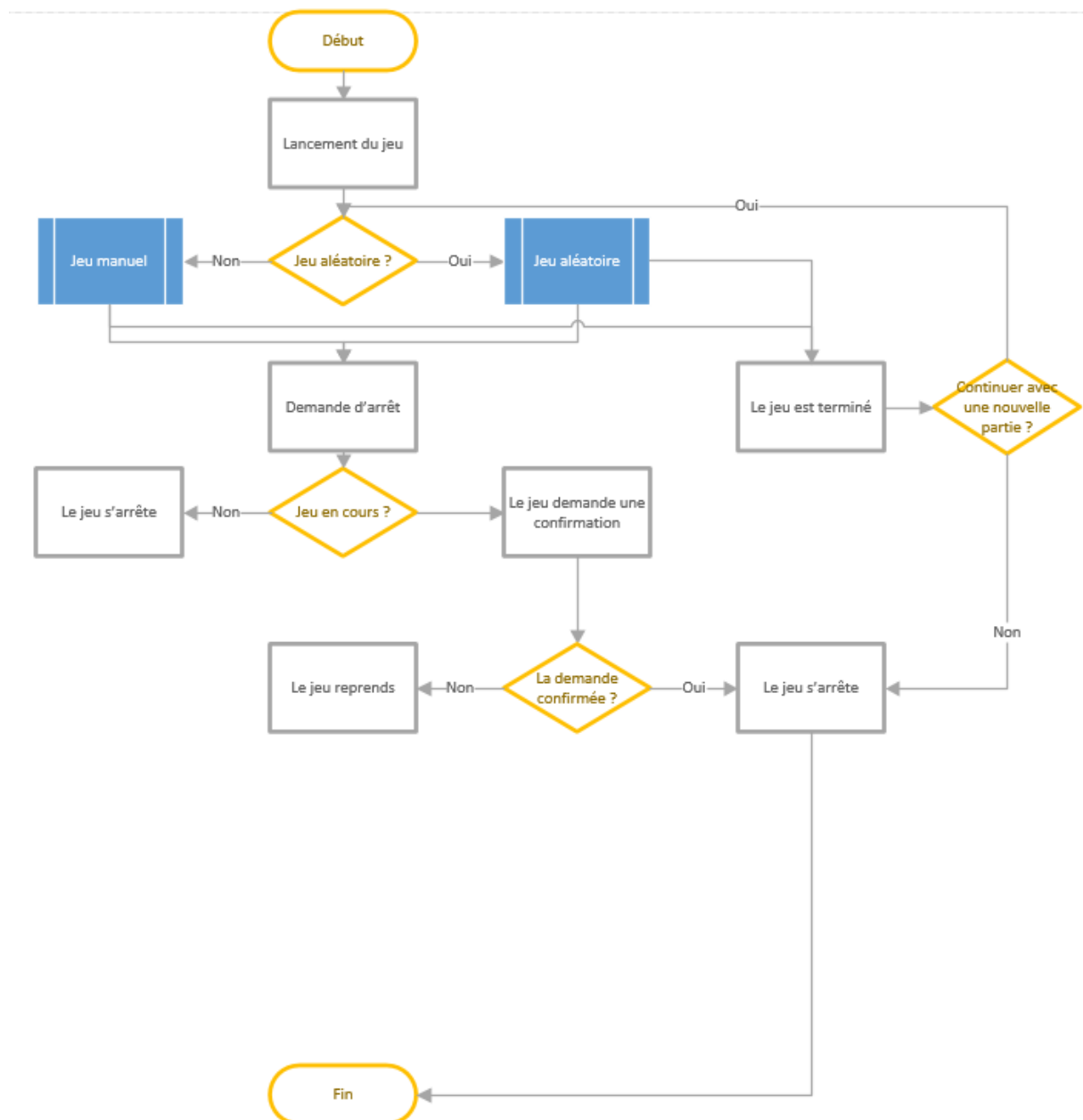
Nom	ThéoriePortLogiques-01	
en tant que	Joueur	
Je veux	Avoir afficher sur comment marche les portes logique	
Pour	connaître/être sur des fonctionnement des portes logique	
Priorité	M	
Action	Condition	Réaction
Lancement d'une partie		la fonctionnement sur les porte logique courante est affichée
en appuyant sur F1		la fonctionnement sur les porte logique courante est affichée

2.1.2 Diagramme de flux

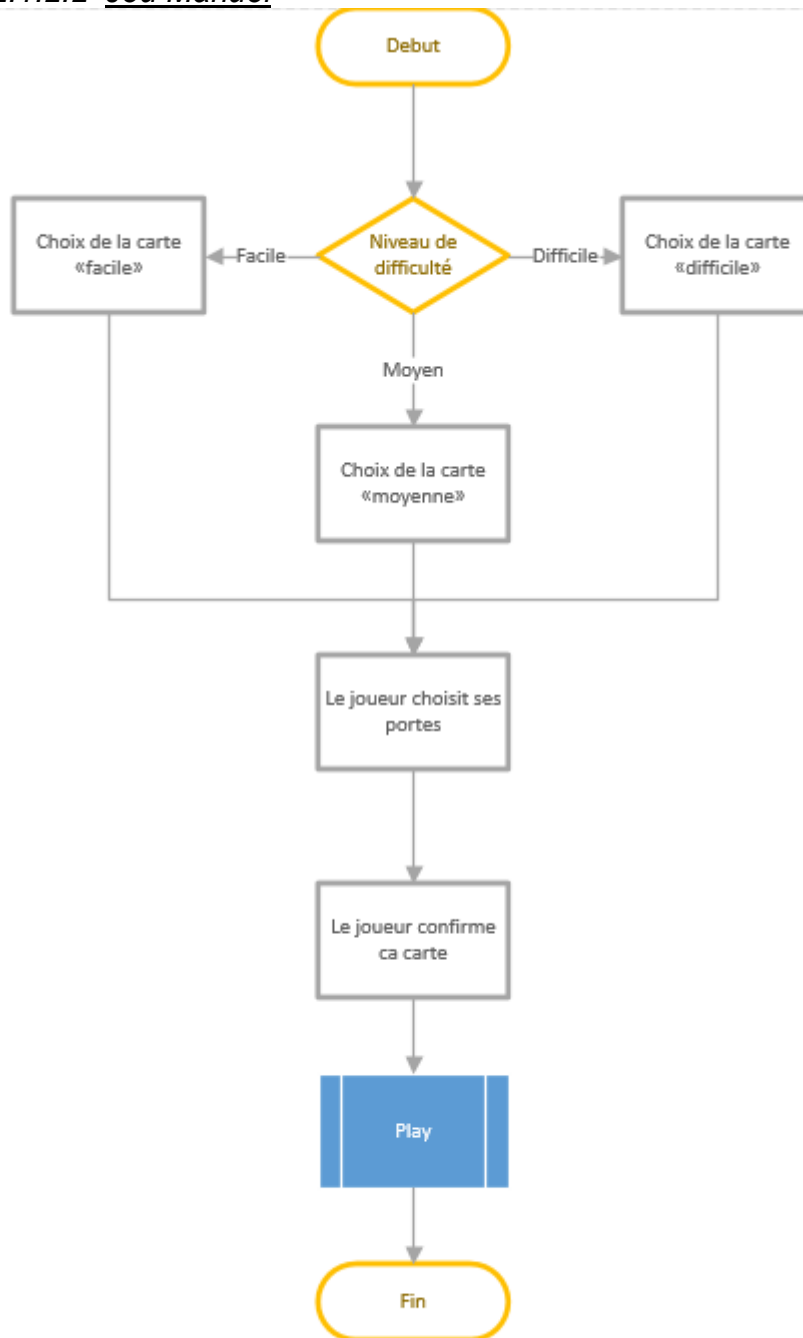
Pour avoir une idée plus précise du comment une fonctionnalité doit être implémentée, j'ai fait plusieurs diagrammes de flux, dont on peut voir comment les fonctions interagissent entre elles

2.1.2.1 Main

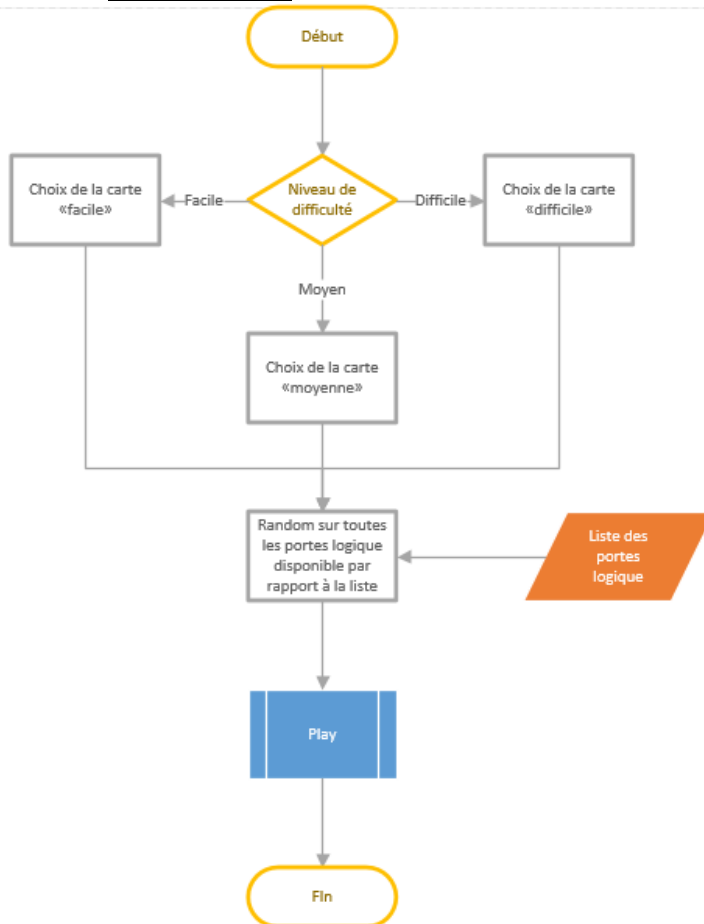
Le code main, est le script maître qui gère l'appelle des autres fonctions, et gère aussi la fermeture du programme



2.1.2.2 Jeu Manuel

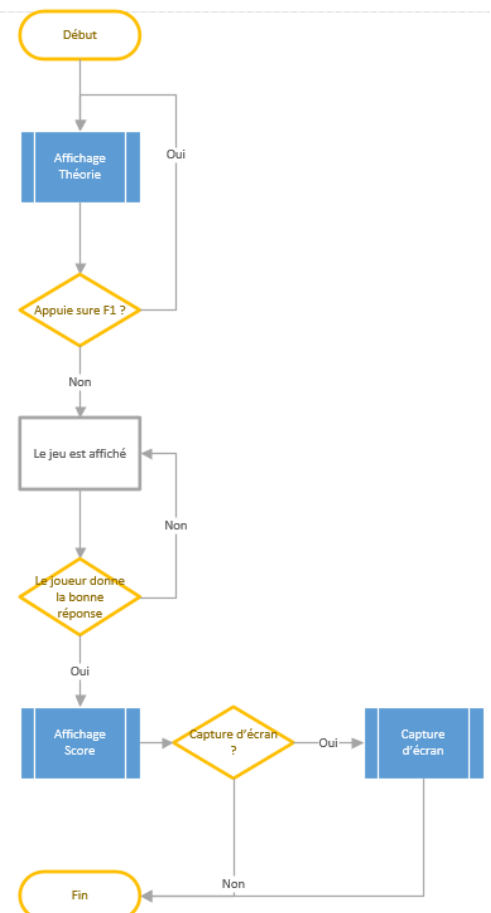


2.1.2.3 Jeu aléatoire



2.1.2.4 Play

Play, est la fonction qui gère la résolution du puzzle, et si le joueur trouve la bonne sortie, le jeu est considéré comme terminé



2.2 Stratégie de test

2.2.1 Test Unitaire

Un test Unitaire sera effectué sur la fonctionnalité de la résolution de puzzle, afin de bien vérifier le bon fonctionnement de cette fonctionnalité, avec le module de tests proposé par Visual Studio.

2.2.2 Tests d'acceptation

Plusieurs tests d'acceptation seront mis sur IceScrum, je demanderais à plusieurs camarades de classe et membres de ma famille pour tester le produit, et de vérifier que toutes les tests d'acceptation passent.

2.3 Risques techniques

- *risques techniques (complexité, manque de compétences, ...).*

Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, ...).

2.4 Planification

Révision de la planification initiale du projet :

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

2.5 Dossier de conception

Fournir tous les documents de conception :

- *le choix du matériel HW*
- *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
- *le choix des outils logiciels pour la réalisation et l'utilisation*
- *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...*
- *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champ) et les requêtes.*

- *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...*

Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !

3 Réalisation

3.1 Dossier de réalisation

Décrire la réalisation "physique" de votre projet

- *les répertoires où le logiciel est installé*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*
- *le numéro de version de votre produit !*
- *programmation et scripts: bibliothèques externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

NOTE : Évitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

3.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

3.3 Erreurs restantes

S'il reste encore des erreurs:

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

3.4 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

4 Conclusions

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

5.2 Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Archives du projet

Media, ... dans une fourre en plastique