

Cycle ingénieur
3A Challenge technique
13 au 17 janvier 2025

Challenge technique – Projet en C

-Back to the Past II- **« Sponsored by R. Zemeckis himself »**

Introduction :

Dans le cadre du **challenge technique** à réaliser en langage C il est proposé de modéliser, programmer et documenter une application de jeu d'aventure en mode texte.

Le jeu d'aventure textuel est une forme de divertissement interactif qui repose sur une narration immersive et des choix du joueur pour déterminer l'évolution de l'histoire.

Pour cela il est demandé de réaliser une application autonome programmée en langage C qui contiendra le moteur de jeu et sera capable de lancer une aventure.

Le travail sera modulaire afin de respecter les normes de programmation en C.

Ce travail sera réalisé en groupe et sera soutenu à la fin de sa réalisation par chaque membre du groupe.

Objectif du projet :

L'objectif principal du projet est de concevoir et de développer un jeu d'aventure textuel entièrement fonctionnel en C.

Une attention particulière sera demandée sur l'histoire du jeu, l'implémentation d'un moteur de jeu textuel, la gestion des choix utilisateurs, la gestion d'un inventaire et l'ensemble des composants auxquels vous penserez pour rendre l'aventure inoubliable.

Objectif 1 : Moteur de jeu textuel

La première chose à mettre en place sera le moteur de jeu responsable d'afficher l'histoire et de prendre en compte les interactions du joueur.

Le moteur devra être rigoureux et robuste afin de répondre systématiquement, quel que soit le contexte.

Objectif 2 : Narration interactive

Une partie de jeu se doit d'être inoubliable. Dans ce cadre, une attention toute particulière sera mise en œuvre pour rédiger une histoire captivante et qui sera plongé le joueur dans votre histoire. A vous l'imagination pour porter des personnages haut en couleurs et nous faire visiter des lieux iconiques de votre univers.

Objectif 3 : Choix et conséquences

Raconter une bonne histoire est déjà bien, mais il est l'heure d'y intégrer une offre de choix significatives pour l'utilisateur qui influencent l'évolution de l'histoire.

Objectif 4 : Inventaire et objets

Rien de plus irritant que de traverser des lieux sans jamais pouvoir y recueillir quoi que ce soit.

Tout bon jeu d'aventure possède son inventaire, et votre jeu ne dérogera pas à la règle, vous implémenterez un inventaire digne de ce nom et vous veillerez à ce que le joueur puisse utiliser ses objets.

L'inventaire et les objets stockables sont toujours de très bons mécanismes de jeu.

Objectif 5 : Système de points et fin

Chaque action, chaque choix peut servir un système de point transparent pour le joueur qui pourra ensuite servir à implémenter

un système de fins alternatives qui feront apparaître la re jouabilité du jeu.

Objectif 6 : Sauvegarde et chargement

Afin d'offrir un confort de jeu exemplaire, et éventuellement permettre au joueur d'explorer d'autres choix, un système de sauvegarde et de chargement sera mis en place au plus tôt dans le développement.

Conclusion :

Ce projet offre une opportunité aux étudiants de développer ses compétences en programmation C, en gestion de projet et en conception d'histoire immersive.

une grande liberté est laissé au groupe quant aux thèmes qu'il souhaite aborder dans son jeu.

Néanmoins, aucun thème, sujet à controverse ne sera admis.

Dans le doute, il sera préférable de demander avant de se lancer dans une écriture qui porterait atteinte aux valeurs de l'école.

La progression étape par étape permettra de mieux maîtriser chaque aspect de l'application afin de délivrer le produit final.

Tips :

Beaucoup de grands noms du jeu vidéo ont été programmé sur ce principe du jeu d'aventure en mode texte.

Peu enclin à afficher des graphismes au démarrage, et nos ordinateurs n'ayant pas été conçu pour ça initialement, le jeu d'aventure textuel a fait les beaux jours de nombreux joueurs initiaux.

C'est justement ce besoin d'avoir un détail graphique en plus de la description qui a poussé les programmeurs de Sierra en 1980 à demander à nos ordinateurs de faire les choses autrement afin d'afficher les prémices de ce qui deviendrait des mondes et des

univers de fictions.

Mais avant ça, l'âge d'or du jeu d'aventure se faisait en mode texte et proposait des histoires incroyables.

On doit donc retrouver cette exaltation en lançant une partie de votre jeu.

Dernier conseil : Avant de démarrer, poser vous la question de quel rendu vous souhaitez obtenir et éventuellement ajoutez des librairies qui peuvent vous aider à rendre votre titre plus conforme à vos attentes. Pour rappel, printf est capable d'afficher de petits graphismes et de la couleur avec les séquences d'échappements et l'utilisation de l'Unicode. La librairie « ncurses » peut aider à dessiner sur un terminal. Attention à votre terminal. Il est impératif de vérifier que ce dernier supporte l'UTF-8 et que vos fichiers sources soient eux aussi encodé en UTF-8 pour utiliser l'Unicode.

Barème de notation – livrable de conception

Item	Détail	Points
Diagramme de modélisation	Diagramme de conception avec structures, interactions, fonctions	8
Plan de test		8
Gestion de projet / tâches	Bonne répartition des tâches entre les membres du groupe	4

Barème de notation – exigences fonctionnelles

Item	Détail	Points
------	--------	--------

Tests	Deux fonctions pertinentes seront testées unitairement	1
Respect des exigences	Le moteur est capable de lancer une aventure/l'aventure n'est pas codé en dur dans le programme.	4
Respect des exigences	Le moteur gère les interactions avec le joueur.	3
Respect des exigences	Le moteur gère la gestion de l'inventaire	3
Respect des exigences	Le moteur permet la sauvegarde et le chargement de parties	3
Respect des exigences	La narration sera immersive, les choix et conséquences auront un impact et il sera possible de terminer et mourir lors de la partie pour proposer un tout.	6

Barème de notation – exigences techniques

Item	Détails	Points
Commentaires	Le code est commenté de manière propre et à bon escient	4

Règle de codage	Les règles du C sont appliqué avec pertinence	4
Structures	Les structures sont utilisées à bon escient	4
Code modulaire et généricité	Le découpage du code est pensé à haut niveau, rendant le moteur facile à maintenir et à faire évoluer	4
Gestion des erreurs	Une bonne gestion des erreurs rend l'application stable et robuste	4

La qualité du rendu de chaque item exercera une influence sur les points accordés.

Absence totale de rendu pour un item : 0 points

Rendu partiel ou bâclé d'un item : moitié des points

Rendu complet et de qualité de l'item : tous les points

Plagiat entre les équipes : note 0 au module Challenges Techniques

Un seul commit sur git : note 0 au module Challenges Techniques