HANI Pamella
HORLAVILLE Albin
EL KANBI Asmae

INFO-3

SPC : Rapport du projet

I. Description du projet

Nous avons réalisé une version simplifiée du jeu Simon édité par Hasbro et édité pour la première fois en 1978. Contrairement au jeu d'origine, notre programme ne dicte pas la couleur sur laquelle appuyée et n'agrandie pas la séquence au fur et à mesure de la réussite des joueurs. Ici le jeu propose une série de couleur que le joueur doit mémoriser puis rendre par l'intermédiaire de la ligne série reliant le clavier de l'ordinateur à la carte STM32. Appuyer sur le bouton bleu de la carte STM32 permet de lancer le jeu. Pour la couleur bleue par exemple, l'utilisateur doit saisir la lettre « b » correspondante à la première lettre du mot. Le jeu est composé des couleurs ROUGE, VERT, BLEU et MAGENTA. Donc le joueur dispose des touches r, v, b et m pour interagir avec le jeu. Si le joueur réussit, la LED tricolore s'illumine 3 fois en verte. Une autre séquence s'exécute alors. Cependant, si jamais le joueur se trompe à un quelconque moment ou qu'il met trop de temps à soumettre sa réponse, la LED rouge clignote 3 fois pour indiquer l'échec. Le jeu s'arrête alors, et le joueur doit relancer le programme pour essayer d'aller le plus loin possible.

II. Explication du rôle des interruptions

Les interruptions nous servent à allumer la séquence au début du jeu, sans arrêter le programme. Puis en moins de 15 secondes gérées par le traitant d'interruption de l'horloge système, le joueur doit appuyer sur les bonnes touches du clavier dans l'ordre. S'il se trompe ou qu'il arrive à 15 secondes sans avoir tout taper, la condition de défaite se déclenche. Le timer se réinitialise à chaque fois que le joueur remporte une manche. Une bonne idée d'utilisation des interruptions serait d'utiliser le buzzer pour jouer une musique pendant le jeu, en variant la fréquence des bits que l'on lui met à 1 ou 0. Mis à part ces deux utilisations, nous ne voyons pas d'autres façons d'utiliser les interruptions dans le cadre de notre jeu.

III. Description des principaux schémas algorithmiques mis en place

Le schéma algorithmique de notre projet est réparti entre le traitant et le « main ».

Dans le traitant se trouve le compteur pour attendre et déclencher la séquence de lumières au début de la manche, ainsi que la défaite. Celle-ci est déclenchée 15 secondes après la fin de la séquence de couleurs. Globalement, le « main » parcourt la liste de mots, ceux-ci étant rangés du mot le plus facile au mot le plus difficile. Ensuite nous utilisons la fonction « getc » car nous n'avons pas réussi à implémenter le fait d'entrer des caractères par

l'interruption. Ceci-dit cela ne gêne en rien le programme puisque si le joueur s'attarde trop à répondre, le traitant d'interruption de l'horloge système déclenchera la défaite. Également, si jamais le joueur se trompe de caractère, le programme le détecte et déclenche la défaite. Il peut donc y avoir autant de niveaux de difficulté que souhaité. Nous avons réfléchi et tenté d'incorporer de l'aléatoire dans le choix de la séquence, mais l'indisponibilité des librairies C nous aurait contraint à complexifier énormément notre programme. Le fait de choisir et conserver l'ordre d'apparition des séquences permet de faire évoluer la difficulté pour atteindre lentement un point si compliqué qu'il sera impossible de le résoudre. Ainsi le joueur aura pour but de faire le meilleur score possible, s'améliorant de partie en partie.

Le « main » quant à lui gère la ligne série, c'est-à-dire les caractères entrés par le joueur. Il choisit une séquence aléatoirement parmi un ensemble prédéfini. Tant que la lettre rentrée est correcte, le programme lit la suivante et dès que la fin de la séquence est atteinte, la victoire est déclenchée. Si une lettre ne correspond pas, cela déclenche instantanément la défaite.

Avec cette structure, le jeu est fluide et le programme prend fin quand l'un ou l'autre l'a décidé. Le traitant n'est pas long mais permet de servir de chronomètre, et d'afficher les couleurs de la séquence, et le code est clair et pourvu de sens.

IV. Description du travail réalisé

Concernant le TP, nous avons réalisé l'exercice 1.1, 1.2 et ceux-ci fonctionnent. Cependant, nous avons éprouvé des difficultés pour l'implémentation de la fonction _async_puts() (dernière question de l'exercice 3) : par manque de temps, nous n'avons pas continué à réfléchir là-dessus. Nous avons également eu du mal à faire fonctionner le traitant d'interruption de l'USART2 (avant-dernière question de l'exercice 3) : celui-ci est bien implémenté, mais pour des raisons obscures, cela ne fonctionne pas.

Globalement, le jeu fonctionne normalement, même s'il y a quelques anomalies (qui n'empêchent pas de jouer). En effet, les deux premières séquences sont identiques : même si nous n'avons pas eu le temps de trouver et corriger ce bug, nous pensons que cela est sûrement dû au fait que nous n'incrémentons pas l'indice de la liste au bon moment.

V. Explication de points importants

Notre traitant d'interruption se sert de la variable « incr » et du modulo pour éteindre et allumer la LED. Toutes les 500 ms, le programme va soit l'allumer, soit l'éteindre pour informer le joueur de la séquence de couleurs. Après cela, le traitant se transforme en chronomètre pour déclencher la défaite si le joueur ne répond pas assez rapidement. En cas de succès d'une séquence par le joueur, la variable « incr » est réinitialisée.

VI. Idées d'amélioration

Comme idée d'amélioration, nous avions pensé à ajouter une musique jouée par le buzzer afin de renforcer l'implication et l'immersion du joueur. En effet, cela aurait été possible en utilisant l'interruption de l'horloge système, et en variant la fréquence des bits que le

programme lui envoie. Ce projet semble extrêmement ambitieux, et consisterait un projet à part-entière. Un deuxième ajout, aussi centré sur le buzzer, pourrait être de produire un son (le plus satisfaisant à l'oreille possible) lorsque le joueur appuie sur la bonne touche du clavier ainsi qu'une musique dédiée en cas de défaite, et réciproquement en cas de victoire.

Pour aller plus loin, un choix de difficulté de base pourrait être proposé au début de chaque partie, pour éviter de refaire les premières séquences, devenant trop simple pour le joueur engrangeant de l'expérience.

Pour finir, pour rendre le jeu commercialisable, il serait bien d'ajouter un bouton par couleur à la carte, pour que celle-ci devienne indépendante de l'ordinateur.