Rapport de DEEP

**Livraison intermédiaire.**

**Sineytin**

Ce projet vise à créer un jeu interactif inspiré de Space Invaders. Le jeu comprendra alors un personnage pouvant tirer des projectiles sur des ennemis en mouvements. Le Sineytin sera jouable à deux et comprendra plusieurs niveaux de difficulté.

**Cahier des charges :**

Les règles du jeu :

-Le joueur doit détruire 5 ennemis en mouvement avant que ceux-ci ne l’atteigne.

-Le score sera basé sur la vitesse que met le joueur pour éliminer les ennemis et le nombre d’ennemis éliminés.

-Le joueur peut tirer via le bouton (un tir par appuie sur le bouton)

-Le joueur peut se déplacer sur l’axe horizontal. (joystick gauche ou droite) (Écran tourné en sens portrait)

-Les ennemies se déplacent horizontalement de gauche à droite et descendent continuellement vers le bas de tel sorte que le joueur dispose d’environ une minute pour tous les éliminer.

Afin d’avoir accès au score et au choix, un menu sera affiché au démarrage du jeu et à chaque fin de partie.

Charte Graphique :

Joueur :

Vaisseau spatial => Carré bleu clair

Ennemis :

Planètes => Disque vert

Projectiles :

Balles => Rectangles rouge

Fond :

Espace => Noir avec des points blancs

Score :

Text Gris

Game Over :

Text Rouge

Menu :

Text non sélectionné Gris

Text sélectionné bleu

Titre en rouge

# Composants utilisés

|  |  |
| --- | --- |
| **Nom ou Référence du composant** | **Description/rôle dans l’application/Caractéristiques principales dans le projet.** |
| Bouton poussoir (x2) | Permet aux deux joueurs de tirer. Alimenté en 3,3V, le bouton ressort une tension de 3,3V quand il est relâché et 0V lorsqu’il est pressé. |
| Joystick (x2) | Permet aux deux joueurs de se déplacer de gauche à droite. Composé de deux potentiomètres et d’un bouton alimenté en 3,3V. Le bouton ressort une tension de 3,3V quand il est relâché et 0V lorsqu’il est pressé. Les potentiomètres ressortent une tension qui varie de 0V à 3,3V selon leur angle de rotation. |
| Ecran TFT ILI9341 240x320 | Affiche les données du jeu. Alimenté en 3,3V, l’écran affiche 240x320 pixels. Il dispose de 11 broches de commande et de 3 broches d’alimentation. Utilise la technologie LCD. |
| Buzzer SMA-13LT | Émet des sons : début, fin, sélection menu, tir, élimination. Emet un son plus ou moins aigue selon la tension d’alimentation de 0V à 3,3V. |

|  |  |
| --- | --- |
| **Pin** | **rôle** |
| **PA0** |  |
| **PA1** | Bouton joystick 2 |
| **PA2** | UART2 Tx => X joystick 2 |
| **PA3** | UART2 Rx => Y joystick 2 |
| **PA4** | écran |
| **PA5** | écran |
| **PA6** | écran |
| **PA7** | écran |
| **PA8** |  |
| **PA9** | Bouton tir 2 |
| **PA10** | Bouton tir 1 |
| **PA11** |  |
| **PA12** |  |
| **PA13** | (non dispo - SWDIO) |
| **PA14** | (non dispo - SWDCLK) |
| **PA15** | non dispo sur certaines bluepill |
| **PB0** | écran |
| **PB1** | écran |
| **PB2** | (non dispo) |
| **PB3** |  |
| **PB4** |  |
| **PB5** | Bouton joystick 1 |
| **PB6** | UART1 Tx => X joystick 1 |
| **PB7** | UART2 Rx => Y joystick 1 |
| **PB8** |  |
| **PB9** |  |
| **PB10** | écran |
| **PB11** | écran |
| **PB12** |  |
| **PB13** |  |
| **PB14** |  |
| **PB15** |  |
| **PC13** | (LED bluepill) |
| **PC14** | (Quartz 32kHz bluepill) |
| **PC15** | (Quartz 32kHz bluepill) |

# Cahier de suivi

Pour chaque date (chaque séance de travail, pendant ou en dehors des créneaux prévus à l’agenda), notez dans ce cahier de suivi les tâches réalisées, les réalisateurs, les difficultés rencontrées, l’état d’avancement de la réalisation…

|  |  |  |
| --- | --- | --- |
| Date | Tâches, réalisateurs, difficultés rencontrés. | A faire la prochaine fois |
| 8/11 PM | Simon : Définir le projet et répartir les tâches.  Vianney : Définir le projet et répartir les tâches.  Quentin : Définir le projet et répartir les tâches. | Simon : Souder l’alimentation.  Vianney : Configuration environnement STM32  Quentin : Configuration environnement STM32 |
| 10/11 AM | Simon : Souder l’alimentation.  Vianney : Configuration environnement STM32  Quentin : Configuration environnement STM32 | Simon : Souder les capteurs.  Vianney : Configurer pin bouton.  Quentin : initialisation de l’écran et mise a l’aise a son fonctionnement |
| 15/11 PM | Simon : Souder les capteurs.  Vianney : Configurer pins du bouton.  Quentin : initialisation de l’écran et mise a l’aise a son fonctionnement | Simon : Finaliser les connectiques et l’attribution des ports.  Vianney : Résoudre problème car appui non détecté.  Quentin :création faisant déplacer un carrée de part en art de l’écran |
| 17/11 AM | Simon : Finaliser les connectiques et l’attribution des ports.  Vianney : Résoudre problème car appui non détecté.  Quentin : création faisant déplacer un carrée de part en art de l’écran | Simon : Ajout d’un bouton et d’un joystick à la carte pour un deuxième joueur.  Vianney : Coder le déplacement du vaisseau.  Quentin : uttilisation d’un joystick pour faire bouger le carrée |
| 22/11 PM | Simon : Ajout d’un bouton et d’un joystick à la carte pour un deuxième joueur.  Vianney : Coder le déplacement du vaisseau.  Quentin :compréhension du fonctionnement du joystick | Simon : Modélisation de la boite contenant notre électronique.  Vianney : Faire partir un projectile du vaisseau lors de l’appui du bouton.  Quentin :régler les problèmes avec le joystick |
| 24/11 AM | Simon : Modélisation de la boite contenant notre électronique.  Vianney : Faire partir un projectile du vaisseau lors de l’appui du bouton.  Quentin :régler les problèmes avec le joystick | Simon : Mise en place du Github afin de partager et de structurer notre projet  Vianney : Générer le projectile à l’appui du bouton.  Quentin :réussir a faire bouger le bouger le carrée avec le joystick |
| 29/11 PM | Simon : Mise en place du Github afin de partager et de structurer notre projet  Vianney : Générer le projectile à l’appui du bouton.  Quentin :réussir a faire bouger le carrée avec le joystick | Simon : Rédaction du livrable et restructuration de nos attentes vis-à-vis du projet.  Vianney :  Quentin : améliorer le mouvement du carée avec le joystick |
| 1/12 AM | Simon : Rédaction du livrable et restructuration de nos attentes vis-à-vis du projet.  Vianney : génération du projectile à l’appui du bouton  Quentin : Mouvement du carrée améliorer | Simon : Documenter en doxygen le code source.  Vianney : Réussir la détection de collision.  Quentin : création d’un fond d’écran. |
| 6/12 PM | Simon : Documenter en doxygen le code source.  Vianney : Réussir la détection de collision.  Quentin : création d’un fond d’écran. | Simon : coder le déclenchement du buzzer  Vianney : compléter la détection de collision  Quentin : compléter le fond d’écran. |

# État d'avancement et analyse du projet réalisé

A l’heure actuelle, la partie hardware, l’affichage graphique et l’implémentation des capteurs est finie.

La seule chose que l’on aurait pu changer est la carte de commande pour pouvoir coder plus facilement un fond d’écran en mouvement. (on aurait pu prendre une carte raspberry)

# Software

Partie Capteur :

 Utilise les deux fonctions précédentes pour faire bouger un objet sur l’axe x.

 Associe le mouvement de droite sur x à un objet.

 Associe le mouvement de droite sur x à un objet.

 Créer le carré.

 Créer le projectile.

 Créer le fond.

# Compléments envisagés

|  |  |
| --- | --- |
| - routage de PCB (avec bluepill = 3 pts || CMS microcontrôleur nu = 4pts) |  |
| - utilisation d'un analyseur logique pour déchiffrer des trames (2 pts) |  |
| - mesure de conso selon scénarios (2 pts) |  |
| - enregistrement de paramètres en flash (1pt) |  |
| - design CAO d'un boîtier (2 pts) |  |
| - documentation doxygen du code source (1pt) |  |
| - jeu de tests pour valider une fonctionnalité software ou hardware (1pt) |  |
| - gestion de version du code source (1pt) |  |