Rodolphe SAVOURET 2A ISE

Nicolas LEMOINE

TP C Embarqué

Objectif :

Le but du TP de C embarqué est de développer une application sur la platine MSP430FG169. Pour cela, nous avons décidé de développer un jeu du genre « Space Invaders ». En effet, le jeu consiste à détruire des astéroïdes avec les armes disponibles, soit des missiles, soit au canon. En plus, la difficulté augmente avec le temps et les météores avancent de plus en plus vite.

Fonctionnalités pour le joueur :

-Déplacer le vaisseau avec le joystick

- Tirer des missiles ou des obus pour détruire les météores avec les boutons A et B

- Visualiser le score à la fin de la partie

Fonctionnalités du jeu :

-Affichage des différents des « sprites » des éléments du jeu (vaisseau, météores, missiles, balles, explosion)

-Création pseudo-aléatoire des météores

-Détection des collisions

-Ajout et Suppression des missiles et balles

-Ajout et Suppression des météores

-Déplacement des différents éléments du jeu (vaisseau, météores, missiles, balles)

-Vitesse du jeu cadencé par TIMER

-Gestion des vies

## DIAGRAMME DU JEU (modules) :



# Fonctions :

**Driver Ecran :**

Le driver de l’écran LCD étant fourni, il est inutile de le documenter ici.

---------------------------------------------------------------------------------------------------------------------------

**Driver Joystick :**

-**void** *InitADC12(****void****) : Initialisation du convertisseur Analogique-Numérique*

-**unsigned int** *GetADC12\_A7(****void****): Récupération de l’état du port d’entrée du joystick*

-**void** *InitJoystick(****void****): Initialisation du Joystick*

**- JOYSTICK\_POS** *GetJoystickPosition(****void****) :* Récupérer l’état du joystick (S : Haut / Bas / Gauche / Droite / Centre)

---------------------------------------------------------------------------------------------------------------------------

**Driver Bouton :**

-**void** *InitBoutons()* : Configuration des boutons en entrée et sur interruption

---------------------------------------------------------------------------------------------------------------------------

**Gestion de l’avion :**

-**void** *afficher\_avion(****void****)* : Permet d'afficher le sprite avion sur l'écran

-**void** *decaler\_avion(****JOYSTICK\_POS pos****) :*  Permet de décaler l'avion en fonction de l'état du joystick

---------------------------------------------------------------------------------------------------------------------------

**Gestion des obstacles : (asteroide)**

-**void** *initMeteor(****void****)* : Fonction initialisant le module météore

-**void** *addMeteore(****void****)* : Fonction permettant d'ajouter une météore sur x aléatoire

-**void** *avanceMeteore(****void****)* : Fonction permettant de calculer l'avancée de tous les météores

-**void** *detruireMeteore(****unsigned int i****)* : Fonction permettant de détruire le météore d'index i

-**void** *afficher\_meteore(****void****)* : Fonction permettant d'afficher les météores

---------------------------------------------------------------------------------------------------------------------------

**Gestion des SPRITES : (images)**

Sprites interchangeable via un fichier .h :

* Avion
* Obstacles
* Munitions

---------------------------------------------------------------------------------------------------------------------------

**Aviator :**

-\_\_**interrupt void** *Timer\_A* ***(void)*** : Fonction principale du jeu , sur timer : Calcul puis Affichage

-**void** *init\_aviator()* : Fonction d'initialisation du jeu et des variables du jeu

-**void** *initTimer()* : Fonction d'initialisation du timer A

-**void** *checkCollision()* : Fonction permettant de détecter des collision entre les éléments

-***void*** *itoa(char\* dest, uint32\_t val, int digits, int base)* : Fonction permettant la conversion d'entier en ascii