**SPACE INVADERS**

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

**Driver Ecran :**

Les éléments du driver Ecran :

* Afficher pixel ( E : X, Y, Couleur)
* Afficher fond ecran (E : Couleur)
* Ecrire du texte sur l’ecran (E : X, Y, Texte) Char ou string

void LCD\_Init(void);

void LCD\_Backlight(const uint8\_t state);

void LCD\_DefaultSettings(void);

void LCD\_Fill(const uint16\_t color);

void LCD\_SetPixel(const uint8\_t x, const uint8\_t y, const uint16\_t color);

void LCD\_DrawLine(uint8\_t x0, uint8\_t y0, const uint8\_t x1, const uint8\_t y1, const uint16\_t color);

void LCD\_DrawRect(const uint8\_t x0, const uint8\_t y0, const uint8\_t x1, const uint8\_t y1, const uint8\_t fill, const uint16\_t color);

void LCD\_PutChar(char c, int x, int y, int size, int fColor, int bColor);

void LCD\_PutStr(char \*pString, int x, int y, int Size, int fColor, int bColor);

void LCD\_Printf (const char \*fmt, ...);

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

**Driver Joystick :**

Les éléments du driver joystick :

* Récupérer l’état du joystick (S : Haut / Bas / Gauche / Droite / Centre)
* Timer pour mettre à jour régulièrement l’état du Joystick

void InitADC12(void);

unsigned int GetADC12\_A7(void);

void InitJoystick(void);

JOYSTICK\_POS GetJoystickPosition(void);

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

**Driver Bouton :**

Les éléments du driver Bouton :

* Récupérer l’état des boutons B1 et B2 (S : True / False)

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

**Gestion de l’avion :**

* Affichage de l’avion (E : X, Y)
* Déplacement de l’avion (E : sens )
* Destruction de l’avion

void afficher\_avion(unsigned int x, unsigned int y);

void decaler\_avion(JOYSTICK\_POS pos);

void initTimer();

void init\_SPACE\_INVADER();

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

**Gestion des munitions :**

* Tirer des munitions
  + Créer un éléments « munition » dans un tableau de munition
* Changer de munition
* Destruction de l’avion

Struct Munition

{

X, Y

Type (Type de munition)

Existe (booleen)

}

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

**Gestion des obstacles : (asteroide)**

* Génération aléatoire d’un obstacle
* Avancement des obstacles
* Destruction d’un obstacle

Struct Obstacle

{

X, Y

Existe (booleen)

}

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

**Gestion des SPRITES : (images)**

Sprites interchangeable via un fichier .h :

* Avion
* Obstacles
* Munitions

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

**Calcul Général :**

* Gestion de l’avance des différents éléments (Avion, munition, obstacles)
* Gestion des collisions (munition – obstacle / avion – obstacle)
* Gestion des points