

# GameSoc

## I / Présentation du projet

Ce projet est un jeu, implémenté sur un FPGA.

Le principe de ce jeu est simple : en bougeant la carte « Nexys A7 », vous bougez un petit carré à l'écran, affiché en noir sur fond jaune.

Sur l'écran, apparaît également un autre carré noir, qui clignote.

Le but est de faire bouger votre carré pour « manger » celui-ci, c'est à dire pour que les 2 carrés se superposent.

Si vous y arrivez, votre score augmente d'un point.

Mais attention, si votre carré touche le bord de l'écran votre score revient à 0 !

L'objectif est donc d'atteindre le score maximal pour gagner la partie.

Il est également possible de régler, à tout moment dans une partie :

- La difficulté du jeu (*cela va influencer sur la vitesse/accélération de déplacement de votre carré*) ;
- Le score à atteindre pour gagner (*5 ou 10*).

## II / Interface utilisateur

Pour fonctionner, le jeu a impérativement besoin :

- D'une carte FPGA Nexys – A7 avec le programme du jeu implémenté dedans ;
- D'un écran connecté via le port VGA de la carte ;
- [Optionnel] D'un pc sous Windows et de la carte FPGA reliée sur un port USB de ce pc .

### Listes des commandes :

- Bouton poussoir 1 : reset du jeu ;
- Interrupteur 1 (*premier en partant de la gauche*) : Affiche le logo « N7 » à l'écran lorsque le jeu est en attente de « Start » ;
- Interrupteur 2 (*second en partant de la gauche*): Gère, en jeu, la difficulté ;
- Interrupteur 3 (*troisième en partant de la gauche*): Gère, en jeu, le score à atteindre pour gagner (5 ou 10) ;

- Interrupteur 4 (*premier en partant de la droite*) : démarre le jeu.

#### Listes des états :

- Appui sur le bouton « reset » ;
- Le message « Hello » est affiché pendant 2 secondes ;
- Le message « Start ? » s'affiche ensuite. Pendant cette étape, l'interrupteur 1 permet d'afficher ou d'effacer le logo « N7 » à l'écran. Il faut actionner l'interrupteur 4 pour démarrer le jeu ;
- Une fois le jeu démarré, le score s'affiche sous la forme « SCORE : X ». Le bouton reset permet à tout moment de revenir au début ;
- Une fois le score maximal atteint, le jeu affiche « SUCCESS ! » et revient à la première étape.

Le jeu affiche les messages à la fois sur les afficheurs 7 segments (*8 caractères maximum*) de la carte FPGA, et sur le port série.

Une interface a également été développée pour permettre un meilleur affichage des données du jeu.



Cette interface a été développée par nos soins, en langage C++ avec la librairie SDL. Elle a été compilée pour Windows uniquement.

Un fichier de configuration (*config.txt*), fourni avec l'application, permet de régler le port COM sur lequel est connecté la carte, ainsi que le baudrate.

### **III / Ressources utilisées**

Ce projet utilise les ressources suivantes :

- Accéléromètre, via liaison SPI avec le FPGA, qui va permettre de détecter les mouvements de la carte ;

- Afficheur 7 segments, pour afficher des informations à l'utilisateur;
- Port VGA, pour afficher le jeu ;
- Interrupteurs, pour contrôler le jeu ;
- Bouton poussoir, pour remettre à zéro le jeu ;
- UART, pour transmettre des informations à l'interface sur le pc ;

## IV/ Conception

La plupart des blocs ont été conçus en VHDL, et sont pilotés par le MicroBlaze via des blocs d'interface entrée/sortie (AXI-GPIO).

- Le bloc « **7segments** » est piloté en entrée depuis le MicroBlaze, via 2 registres contenant les données à écrire sur les 7 segments (*1 registre = 4 caractères sur les 7 segments, et tout le traitement est effectué depuis le SDK*) ;
- Le bloc « **accelerometre** » donne l'accélération sur l'axe X et l'accélération sur l'axe Y via 2 registres, reliés en tant qu'en entrée sur le MicroBlaze ;
- Le bloc « **VGA** » est piloté en entrée depuis le Microblaze, via 2 registres : la position en X et en Y du carré à afficher sur l'écran.

Le développement sur le MicroBlaze en C, a pris une bonne part du projet (*environ 60 % du temps total*). Il s'agit en effet du cœur du jeu, donc ce n'était pas une partie à négliger.

La partie VHDL (hardware) a été réalisée en premier pour gagner du temps sur la partie SDK.