

# ELE3312 Projet Final:

## Jeux Tetris

Automne 2017

### Introduction

Tetris est un jeu vidéo de puzzle conçu par Alekseï Pajitnov à partir de juin 1984 sur Elektronika 60. Votre projet final consiste à implanter ce jeu sur votre système embarqué (microcontrôleur + écran LCD). En outre, le seul moyen de communication avec le microcontrôleur sera le sonar. Vous devrez donc l'utiliser pour indiquer les déplacements de vos pièces.

## 1 Présentation du jeu

Dans ce jeu, des briques de différentes formes (Tétriminos) tombent du sommet d'un puits jusqu'à toucher le sol ou un autre Tétrimino. Le joueur doit dévier la chute des pièces afin de créer des lignes horizontales complètes, qui s'élimineront. Chaque ligne éliminée augmente le score en fonction du nombre de lignes complétées simultanément. Plus le score est grand, plus la vitesse de chute des briques augmente. La difficulté s'en trouve accrue. Lorsqu'un Tétrimino est placé, un nouveau apparaît automatiquement en haut du puits. À tout moment, le nombre de lignes éliminées ainsi que la prochaine pièce à paraître sont indiqués à l'écran. Si un Tétrimino touche le sommet du puits par empilement, la partie s'arrête et le score est verrouillé : c'est la fin de la partie.

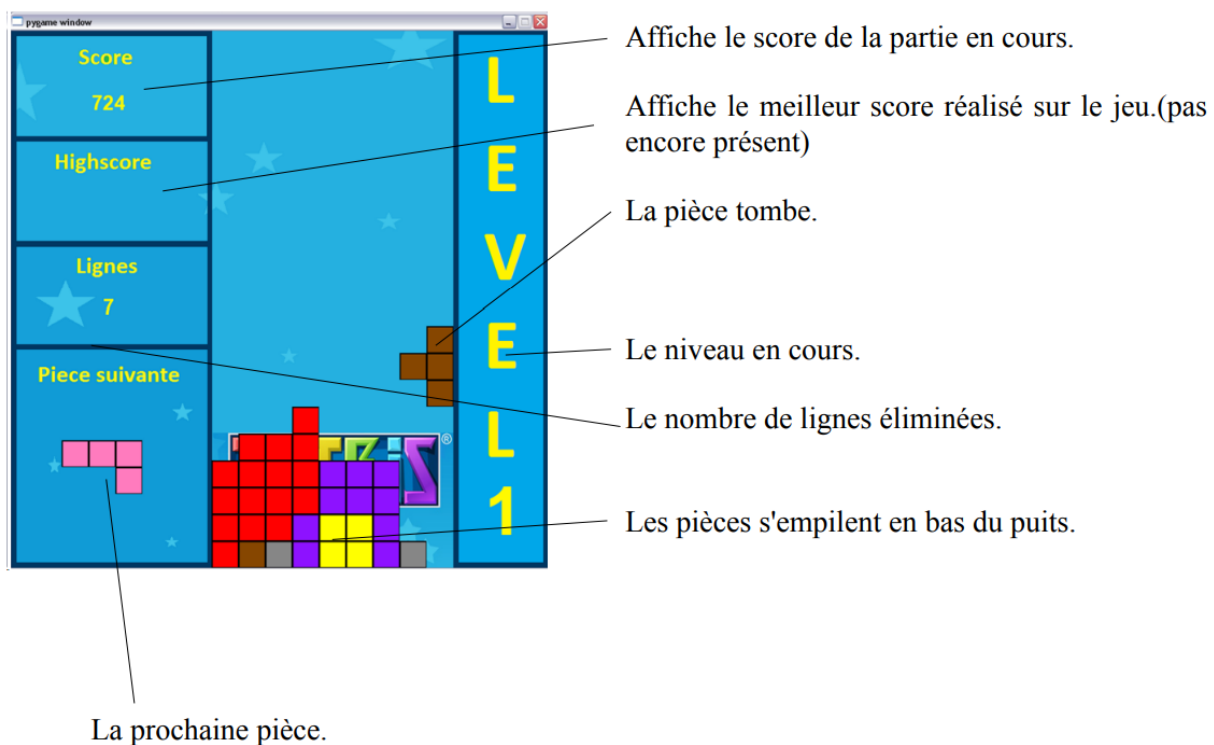


Figure 1: Tetris

Le joueur doit essayer de jouer le plus longtemps possible avant que le puits du jeu ne soit rempli. Le score est calculé comme suit en fonction du nombre de lignes supprimées :

- 1 ligne qui disparaît rapporte 40 points
- 2 lignes qui sont supprimées rapportent 100 points
- 3 lignes qui sont supprimées rapportent 300 points
- 4 lignes (on ne peut pas plus) rapportent 1200 points.

## 2 Utilisation du sonar

### 2.1 Principe de fonctionnement

Le principe du sonar est très simple : un haut-parleur produit une onde sonore dans la gamme des ultrasons. Cette bande de fréquence se situe entre 20 kHz et 1 MHz, c'est à dire juste au-dessus de la bande des sons audibles par l'oreille humaine (c'est d'ailleurs pourquoi on appelle cette bande de fréquence les ultrasons). Dans notre cas, la fréquence se situe autour de 40kHz.

Lorsqu'elle atteint un objet, l'onde va être en partie réfléchi et elle aboutira à un récepteur un certain temps après son émission initiale (Figure 2). Pendant ce temps, l'onde

aura effectué un aller-retour entre le sonar et l'objet. Etant donné la vitesse de propagation d'une onde sonore dans l'air (environ 340 m/s), on peut calculer la distance à laquelle se trouve l'objet :

$$d_{object} = v_{onde} \times \frac{t_{aller/retour}}{2} \quad (1)$$

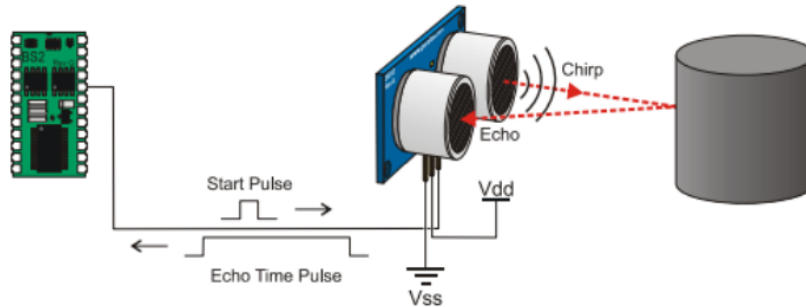


FIGURE 1 – Fonctionnement du Sonar

Figure 2: Fonctionnement du Sonar

Les sonars sont depuis longtemps utilisés par la marine, et en particulier dans les sous-marins. En effet, les radars ne sont pas utilisables sous l'eau car les ondes électromagnétiques s'y atténuent rapidement. Sinon, c'est le même principe de fonctionnement. Une autre application des sonars est ... le "radar" de recul ! En effet, contrairement à ce que leur nom laisse penser, les radars de recul de vos voitures utilisent des ultrasons pour détecter les obstacles.

## 2.2 Contrôle du sonar depuis le microcontrôleur

Le contrôle du sonar va se faire un peu de la même manière que celui du DAC. Chaque fois que vous voudrez lancer une détection sur le sonar, vous devrez lui envoyer un signal de trigger. Vous pouvez l'effectuer de manière occasionnelle par logiciel, ou régulièrement grâce à un timer. En retour, vous obtiendrez un signal de type créneau, de même durée que l'aller-retour de l'onde sonore (cf Figure 2). Afin de mesurer la durée de ce créneau, il vous faudra utiliser un timer en mode Capture. Vous trouverez plus de détails dans le livre de référence au chapitre 15. En résumé, ce mode fonctionne de la manière suivante :

1. Vous devez configurer le Timer de manière à générer une interruption alternativement sur un front montant puis sur un front descendant du signal d'entrée
2. Lorsqu'un front montant est détecté, la valeur du compteur est enregistrée dans le registre CCR. Vous devez lire et enregistrer cette valeur
3. Au prochain front descendant, allez lire la nouvelle valeur de CCR
4. La différence entre les deux mesures vous donnera la durée de votre créneau

5. Attention à la configuration de votre timer (prescaler et ARR). Cela influencera les caractéristiques de votre mesure.

## 2.3 Commandes à reconnaître

Vous utiliserez deux détecteur à ultrasons (un horizontal et un vertical). Un des défis sera de vous assurer qu'ils n'interfèrent pas entre eux. Les mouvements suivants doivent être reconnus :

1. Un mouvement du haut vers le bas détecté par le sonar vertical entrainera une descente rapide du Tetrimino;
2. Un bref passage de votre main dans le champ du sonar vertical entrainera une rotation du Tetrimino.
3. Un déplacement de gauche à droite ou de droite à gauche dans le champ du sonar horizontal guidera le déplacement de gauche à droite du Tetrimino. Par exemple, pour un sonar horizontal placé à droite, si votre main est loin du sonar, alors votre Tétrimino devra être à gauche dans votre jeu. Par contre, si celle-ci est proche de votre sonar, alors votre Tétrimino devra être à droite dans votre jeu.

## 3 Production de son

Il vous est demandé de produire du son pour agrémenter le jeu. Minimalelement, on veut entendre un bip lorsqu'une pièce se stabilise et un son spécial lorsqu'une ligne est complétée. La manière la plus simple de produire un son est d'utiliser le PWM. Toutefois, aucune restriction ne vous est imposée.

## 4 Contraintes

1. Respecter les règles du jeu Tetris.
2. Chaque Tétrimino doit être une des formes définies dans le fichier *shapes.h* et illustrées à la figure 3.
3. Utiliser les structures de données et les constantes qui seront mises à votre disposition. Si on change les valeurs des paramètres, votre code doit pouvoir s'adapter (après recompilation).
4. Les sonars doivent continuellement prendre des mesures (utiliser les interrupts) et les rendre disponibles dans deux variables (int sonar\_x, sonar\_y).
5. Le son doit être rendu disponible aux bornes d'une connecteur femelle jack stéréo de 1/8 de pouce.

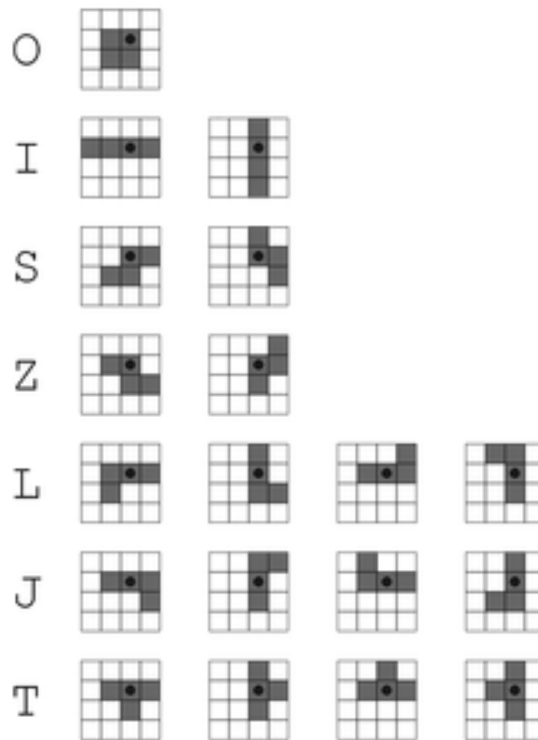


Figure 3: Tétrimino

## 5 Étapes suggérées

Le développement de l'application en tant que telle devrait se faire au cours des deux premières semaines. La troisième semaine devrait être consacrée à la finalisation du déverminage et l'intégration des entrées/sorties (capteurs ultrasons et production du son). Les trois séances en laboratoire devraient être utilisées pour vérifier les points suivants avec le chargé de laboratoire :

1. Mesure de la distance par ultrasons.
2. Production de son.
3. Intégration du projet final.

Pour rappel, la dernière séance de laboratoire est consacrée à l'évaluation détaillée de votre projet.

## 6 Barème

- *2pts* Le Tétrimino peut bouger de gauche à droite sans jamais dépasser la limite du champ de jeux.

- *1pts* Le Tétrimino se place correctement sur une ligne ou sur un autre Tétrimino.
- *1pts* Lorsqu'une ligne horizontale est complétée, elle disparaît et les blocs supérieurs tombent.
- *1pts* La pièce suivante et le nombre de lignes éliminées sont correctement annoncés.
- *1pts* Le score respecte les conditions décrites dans la section 1.1.
- *1pts* La vitesse de chute du Tétrimino augmente au fur et à mesure que le niveau augmente.
- *2pts* Les sonars sont capables de mesurer correctement la distance (erreur max : 2cm sur 1 mètre).
- *2pts* Les deux mouvements sur le sonar vertical sont bien reconnus.
- *2pts* Les sons sont produits au bon moment.
- *2pts* La qualité graphique et sonore.
- *3pts* La qualité de la démonstration lors de la compétition (évaluation par les pairs).
- *1pt* Bonus de la meilleure démonstration lors de la compétition.