ELE-792

Projet de fin d’études en génie électrique

**Rapport d’étape**

Remis à :

Jean-François Boland

Rédigé par :

Jimmy Vaillancourt

VAIJ21118302

François Gervais

GERF

Remis le XX mai 2010

Table des matières

Introduction

Chapitre 1 – Analyse de la problématique

1.1 Lacunes des consoles semblables actuelles

1.2 Division en sous-problèmes

Chapitre 2 – Choix d’une solution

2.1 Synthèse des besoins selon les sous-problèmes

2.2 Définition des barèmes menant à la prise de décision

2.3 Évaluations des solutions possibles

2.4 Prise de décision en fonction des barèmes établis

Chapitre 3 – Risques et opportunités

3.1 Risques

3.2 Opportunités

Chapitre 4 – Réalisation

Conclusion

Liste de référence

Bibliographie

# Description du problème

Si vous voulez développer un jeu vidéo, quelques possibilités s’offrent à vous. Vous pouvez bien sûr le faire sur ordinateur, mais il est souvent plus intéressant de jouer dans le confort de son salon. Dans ce cas, vous pouvez vous tourner vers les dernières consoles de jeux qui offrent souvent la possibilité faire son propre jeu. Cependant dans le dernier cas, vous êtes limité aux fonctionnalités que le fabricant veut bien vous donner.

Il vous reste donc les consoles ouvertes. Ces consoles sont faites dans le but de fournir une plateforme de base pour le développement de jeux vidéo tout en permettant à l’utilisateur de modifier tout le logiciel roulant sur celle-ci. De plus, dans la plupart des cas, ce genre de plateforme offre les plans du matériel afin de permette à quiconque ayant les habiletés nécessaires, de modifier la plateforme. Bref, tous les pouvoirs sont donnés à l’utilisateur.Ceci est très intéressant, mais les choix disponibles présentement comportent plusieurs lacunes.

Premièrement, les solutions existantes n’utilisent pas de contrôleur graphique matériel. Ceci limite grandement les performances, car il est très difficile pour le logiciel de faire le rafraichissement d’un écran. Comme un grand pourcentage du temps du processeur est utilisé pour la gestion de l’écran, il ne reste que peu de temps pour la gestion du jeu lui-même qui est une tâche assez difficile en soi.

Encore du coté du matériel, les solutions existantes on souvent trop peu de mémoire disponible ce qui rend impossible l’utilisation d’image de haute qualité visuelle à l’intérieur du jeu.

Une autre lacune des consoles ouvertes est le logiciel fourni. En effet, celui-ci est souvent très optimisé dû au manque de performance du microcontrôleur ce qui le rend souvent difficile à comprendre. De plus, la plupart de temps, aucun moteur de jeux n’est disponible ce qui rend la programmation de jeux assez ardus. Pour finir, le jeu et le noyau sont souvent indissociables.

Du côté de l’interface utilisateur, les consoles ouvertes actuelles utilisent souvent de simples contrôleurs désuets ce qui limite l’envergure des jeux développés.

# Objectifs

L’objectif de ce projet est de développer une console de jeux ouverte offrant une solution aux lacunes des autres consoles du même type.

Du coté logiciel nous allons commencez par faire une couche d’abstraction matérielle permettant à ceux qui ne sont pas familiers avec la programmation embarquée de tout de même modifier le noyau. De plus, nous allons développer un moteur de jeu facilitant la création de jeux vidéo. Le tout sera fait en utilisant la programmation orientée objet permettant de faciliter la réutilisation de code pour les sections génériques des jeux.

Ce moteur de jeux sera ensuite adapté pour les ordinateurs. Ceci permettra à l’usager de développer son jeu sur PC avant de l’essayer sur la console. De cette façon, nous pourrons accélérer de cycle de développement du jeu.

De plus, nous développerons quelques jeux de démonstration, ce qui facilitera l’apprentissage d’un nouvel utilisateur.

Une manette de jeux récente sera aussi utilisée afin d’élargir les possibilités lors du développement de jeux.

# Hypothèses et contraintes

## Hypothèses

Du côté logiciel, nous ferons l’hypothèse que le temps processeur requis pour le traitement de l’image est beaucoup plus grand que le traitement de toute autre section du jeu. En utilisant une intelligence artificielle simple pour les ennemis, ceci est assez près de la réalité.

Du côté matériel, nous démarrons avec l’hypothèse qu’il est possible de réaliser, avec une certaine facilité, un convertisseur numérique analogique de type R2R d’assez bonne qualité afin de pouvoir transmettre les images générées sur un écran avec entrée VGA. Dans le cas où ça ne fonctionnerait pas, il nous reste toujours l’option d’utiliser un circuit intégré dédié à cette tâche, mais à un coût plus élevé. Nous émettrons aussi l’hypothèse qu’il est faisable de réaliser une telle console sans dépasser un budget de 50$.

## Contraintes

Premièrement, la contrainte la plus importante est, comme indiqué plus haut, le coût de réalisation du prototype. Ce dernier ne devrait, idéalement, pas dépasser 50$. Il faudra donc être minutieux dans le choix des composants utilisés, sans trop influencer la qualité du produit final.

Cependant, du côté logiciel nous serons limités à la puissance de calcul du microprocesseur choisi ce qui ne sera sûrement pas très élevé. De plus, comme nous voulons produire du code facilement réutilisable, nous nous limiterons aux langages de programmation orientés objet. Ce langage devra aussi être assez populaire afin de simplifier l’apprentissage.

# Calculs et analyse

## Choix du microcontroleur

Le choix du microcontrôleur est un des choix les plus importants de ce projet, car c’est lui qui dictera les principales contraintes futures autant matérielles que logicielles. Le choix sera fait selon plusieurs critères, dont la puissance de calcul. Ceci est assez compliqué à estimer, mais doit être fait lors de l’étude préliminaire. Pour ce faire, fixons-nous d’abord des barèmes.

Nous voulons que la console permette la création de jeux du calibre du Super Nintendo des années 90. Celui-ci possède plusieurs modes vidéo, dont plusieurs résolutions possibles, plusieurs nombres d’images par seconde, etc. Nous allons essayer de faire un tableau récapitulatif en utilisant des valeurs les plus utilisées ou des valeurs moyennes selon le cas. Ceci va nous aider à fixer des valeurs nominales pour les performances que nous voulons atteindre avec notre console.

|  |  |
| --- | --- |
|  | Super Nintendo |
| Architecture | 16 bits |
| Fréquence | 3.58 Mhz (valeur effective max) |
| Résolution | 256x224 (plus utilisé) |
| Couleurs | 16 bits |
| Image par seconde | 10 (valeur moyenne) |
| Espace pour le jeu | 6 MB (max) |
| Nombre de couches d’affichage | 2-3 |

À partir de ces valeurs, faisons une série de choix qui serait des valeurs acceptables pour notre console.

|  |  |
| --- | --- |
|  | Console |
| Architecture | 32 bits |
| Résolution | 480x272 |
| Couleurs | 24 bits |
| Image par seconde | 10 |
| Nombre de couches d’affichage | 2 |

Nous faisons ces choix pour plusieurs raisons. Premièrement, les architectures 32 bits sont très populaires de nos jours. Deuxièmement, les images de 24 bits par pixel sont un format très répandu et 10 images par seconde, en moyenne, est un taux de rafraichissement encore utilisé aujourd’hui et assez facile à atteindre. Nous nous limiterons à deux couches d’affichage, car la plupart des jeux n’utilisent que 2 couches. Pour finir, la résolution choisie est un bon compromis entre la haute résolution et la résolution du Super Nintendo. C’est cette résolution qui est normalement utilisée pour les écrans 4.3 pouces, très populaire chez les consoles portables.

Avec ces informations et notre hypothèse de départ sur le temps de traitement de l’image par le processeur, nous pouvons estimer la puissance de calcul du processeur dont nous aurons besoin.



Le processeur devra donc être capable de faire en moyenne 2.6 millions de transferts par seconde pour l’affichage. Ceci peut être assez contraignant, mais il sera toujours possible de faire des choix matériels afin d’alléger la tâche comme avec l’utilisation de canaux DMA.

## Convertisseur numérique analogique

Pour ce qui est de la conception du convertisseur numérique analogique, nous voulons en réduire le coût au minimum. C’est la raison pour laquelle nous avons choisis de créer notre propre circuit de conversion à partir d’un modèle de type R2R. Nous aurons donc besoin de 3 circuits de conversion de 8 bits chacun afin d’alimenter les canaux R, G et B du port VGA.

Afin de faire un choix éclairé sur les valeurs de résistances et des autres composants à utiliser, nous avons dressé une liste de quelques paramètres d’une entrée VGA.

Tension analogique des signaux RGB : 0,7Vp-p

Impédence d’entrée des signaux RGB : 75ohms

Tension des signaux de synchronisation horizontaux et verticaux : 5V TTL

La tension produite par notre convertisseur R2R avec n=8 bits et la tension de sortie de chaque bit égale à 3,3V:



Il nous faudra donc trouver un moyen de convertir un signal variant entre 0 et 3,29V en un signal variant entre 0 et 0,7V. Pour l’instant, nous croyons utiliser, pour chaque canal de couleur, un simple convertisseur de tension avec un suiveur comme ci-dessous :

0-3,29V

R1

R2

0-0,7V

De plus, les signaux de synchronisation sont donnés par des pulses de niveau TTL de différentes largeurs. Le microcontrolleur pourra fournir directement les pulses nécessaires à 3,3V.