



PROJET ARCHITECTURE DES ORDINATEURS

Synthétiseur MIPS



QUENTIN DESCHAMPS

MAJDA EL ATIA

Mai 2020

Table des matières

1	Introduction	2
2	Réalisation	2
2.1	Organisation du clavier	2
2.2	Principe	3
2.3	Sortie audio	3
3	Utilisation	5
4	Conclusion	5
5	Annexes	6

1 Introduction

Le but de ce projet est de réaliser un synthétiseur musical en langage assembleur MIPS. Ce rapport explique comment le programme a été réalisé et comment l'utiliser. Vous trouverez aussi en annexe la liste des commandes.

2 Réalisation

Avant de programmer, nous avons d'abord analysé le sujet afin de décider comment structurer notre projet.

2.1 Organisation du clavier

Un synthétiseur musical est composé d'un clavier pour jouer et de boutons réalisant différentes fonctions, notamment pour changer d'instrument. C'est pourquoi, nous avons d'abord réfléchi à l'organisation d'un clavier AZERTY pour simuler celui d'un synthétiseur. L'illustration ci-dessous montre cette organisation.



FIGURE 1 – Organisation du clavier

Ce schéma utilise la notation anglo-saxonne des notes. Vous pouvez vous reporter au tableau ci-dessous si vous n'y êtes pas familier.

Notation française	Do	Ré	Mi	Fa	Sol	La	Si
Notation anglo-saxonne	C	D	E	F	G	A	B

TABLE 1 – Notation

Cette organisation est très inspirée d'un piano virtuel en ligne ¹. Vous trouvez

1. Lien [ici](#)

sur le schéma :

- **En bleu** : les notes non altérées qui correspondent aux touches blanches d'un piano
- **En noir** : les notes altérées qui correspondent aux touches noires d'un piano
- **En vert** : les touches permettant de changer d'instrument
- **En gris** : les touches pour modifier la durée des sons
- **En orange** : la touche de sortie du programme

Les indices associés aux notes correspondent à leur numéro d'octave. Le clavier AZERTY est ici très bien optimisé car ce synthétiseur peut parcourir presque 3 octaves (35 touches au total). À chaque touche est donc associée un caractère ASCII qui correspond à une commande.

2.2 Principe

Le principe du programme est très simple. Lorsque celui-ci est lancé, il rentre dans une boucle. Celle-ci se répète en réalisant les trois étapes suivantes :

1. **Lecture du caractère** : l'utilisateur appuie sur une touche. Le programme lit alors le caractère ASCII associé à la touche.
2. **Décodage de la demande** : le programme cherche à quelle commande est associée le caractère qu'il vient de recevoir². Il y a 5 actions différentes :
 - Quitter le programme (caractère "SPACE")
 - Changer d'instrument en tapant son numéro (caractère "&")
 - Changer d'instrument par le suivant dans la liste (caractère "=")
 - Diminuer (caractère "<") ou augmenter (caractère ">") la durée des sons
 - Jouer un son (caractère parmi les bleus ou noirs du schéma)
3. **Exécution de la demande** : le programme exécute la demande de l'utilisateur.

Ce cycle recommence jusqu'à ce que l'utilisateur demande de quitter le programme. Si la touche appuyée ne correspond à aucune fonction ("q" par exemple), alors le programme ne fait rien et recommence la boucle.

2.3 Sortie audio

Pour produire des sons, le programme utilise le *syscall* 31 du langage MIPS. Celui-ci utilise le service MIDI de l'ordinateur. À son appel, le programme génère un son et continue. Plusieurs paramètres doivent être mentionnés :

- La **hauteur** du son (*pitch* en anglais) dans le registre **\$a0**. Cela correspond à un entier entre 0 et 127. La valeur 60 correspond à un C4 (do médium).

2. Voir la table des commandes 3 en annexe

Par exemple, en partant de ce C, on peut construire la gamme chromatique suivante :

Valeur	60	61	62	63	64	65	66	67	68	69	70	71
Note	C	C#	D	D#	E	F	F#	G	G#	A	A#	B

TABLE 2 – Correspondance entre valeur et note

La note la plus basse qui peut être produite par le synthétiseur MIPS est le C3, correspondant au caractère "a". Il a pour valeur 48. Dans la fonction **fctJoueSon**, le programme va donc chercher le caractère dans un tableau. Le tableau est ordonné du caractère de la note la plus grave à celui de la note la plus aiguë. Ainsi, le programme parcourt le tableau en recherchant la valeur du caractère demandé par l'utilisateur. Une fois celle-ci trouvée, la note correspondante est donnée par la formule :

$$pitch = 48 + i$$

Où i est l'indice du caractère dans le tableau "tabNotes".

- La **durée** du son dans le registre **\$a1**. Celle-ci correspond à une valeur en millisecondes. Cette durée est initialisée à 200 ms. L'utilisateur peut la diminuer avec la touche "<" ou l'augmenter avec ">" par pas de 20 ms grâce aux fonctions **fctDiminueDuree** et **fctAugmenteDuree**. La ligne 175 fait en sorte de ne pas descendre en dessous de 0 dans la première fonction.
- L'**instrument** dans le registre **\$a2**. Celui-ci est représenté par une valeur entre 0 et 127. La valeur 0 est celle par défaut et correspond à un piano. Vous pouvez trouver ci-dessous l'organisation des instruments tirée de la documentation MIPS :

0-7	Piano	64-71	Reed
8-15	Chromatic Percussion	72-79	Pipe
16-23	Organ	80-87	Synth Lead
24-31	Guitar	88-95	Synth Pad
32-39	Bass	96-103	Synth Effects
40-47	Strings	104-111	Ethnic
48-55	Ensemble	112-119	Percussion
56-63	Brass	120-127	Sound Effects

FIGURE 2 – Liste des instruments

- Le **volume** du son dans le registre **\$a3**. Cela correspond encore à une valeur entre 0 et 127, où 0 est le silence et 127 est le plus bruyant. Le volume étant assez faible, nous avons décidé de fixer cette valeur à 127. L'utilisateur peut toujours modifier le volume sonore de son ordinateur.

3 Utilisation

Voici maintenant comment utiliser le synthétiseur MIPS :

1. **Lancer le programme** : réalisez l'assemblage puis lancez le programme. Un message de bienvenue s'affiche.
2. **Jouer** : appuyez sur les touches de votre clavier pour jouer. Si vous n'entendez rien, montez le volume !
3. **Changer d'instrument** : appuyez sur la touche "&". Le programme vous demande le numéro de l'instrument que vous souhaitez (se reporter à la liste). Entrez le numéro et appuyez sur la touche ENTER.
4. **Instrument suivant** : appuyez sur la touche "=" et le programme change l'instrument actuel par le suivant dans la liste. Il vous affiche son numéro.
5. **Modifier la durée des sons** : appuyez sur la touche "<" pour réduire et sur ">" pour augmenter la durée des sons produits. Le programme affiche la durée en millisecondes.
6. **Quitter le programme** : appuyez sur la touche SPACE.

4 Conclusion

Le synthétiseur MIPS est fonctionnel et simple d'utilisation. Le clavier est bien optimisé et peut parcourir presque 3 octaves. De plus, la possibilité de changer d'instrument en fait un vrai synthétiseur.

Cependant, la gestion de la durée des sons ne repose pas sur le vrai principe d'un piano. En effet, l'utilisateur ne peut pas moduler la durée des sons en appuyant plus ou moins longtemps sur une touche comme c'est le cas pour un piano. Cela vient du système de lecture des caractères qui est lent et bloque le programme. Cependant, les restrictions du langage MIPS rendent ce problème irrésoluble. De ce fait, il est aussi impossible de jouer des accords. La possibilité de changer la durée des sons avec les touches "<" et ">" est une solution alternative dans les limites de ce langage.

5 Annexes

Lien du dépôt GitHub : <https://github.com/Quentin18/Synthesizer-MIPS>

Table des figures

1	Organisation du clavier	2
2	Liste des instruments	4

Liste des tableaux

1	Notation	2
2	Correspondance entre valeur et note	4
3	Table des commandes	7

Caractère	Code ASCII	Commande
a	97	C3
é	233	C#3
z	122	D3
"	34	D#3
e	101	E3
r	114	F3
(40	F#3
t	116	G3
-	45	G#3
y	121	A3
è	232	A#3
u	117	B3
i	105	C4
ç	231	C#4
o	111	D4
à	224	D#4
p	112	E4
w	119	F4
s	115	F#4
x	120	G4
d	100	G#4
c	99	A4
f	102	A#4
v	118	B4
b	98	C5
h	104	C#5
n	110	D5
j	106	D#5
,	44	E5
;	59	F5
l	108	F#5
:	58	G5
m	109	G#5
!	33	A5
ù	249	A#5
=	61	Instrument suivant
&	38	Changement instrument
<	60	Diminue durée des sons
>	62	Augmente durée des sons
SPACE	32	Exit

TABLE 3 – Table des commandes