Rapport de test
Synthesizer

Vincente NATTA
Dorian LEROY
Kheireddine BOURAHLI
Guillaume DARVES
Yves MOCQUARD
Caroline LAVAURE
Robin HACAULT
Anthony LHOMME

SOMMAIRE

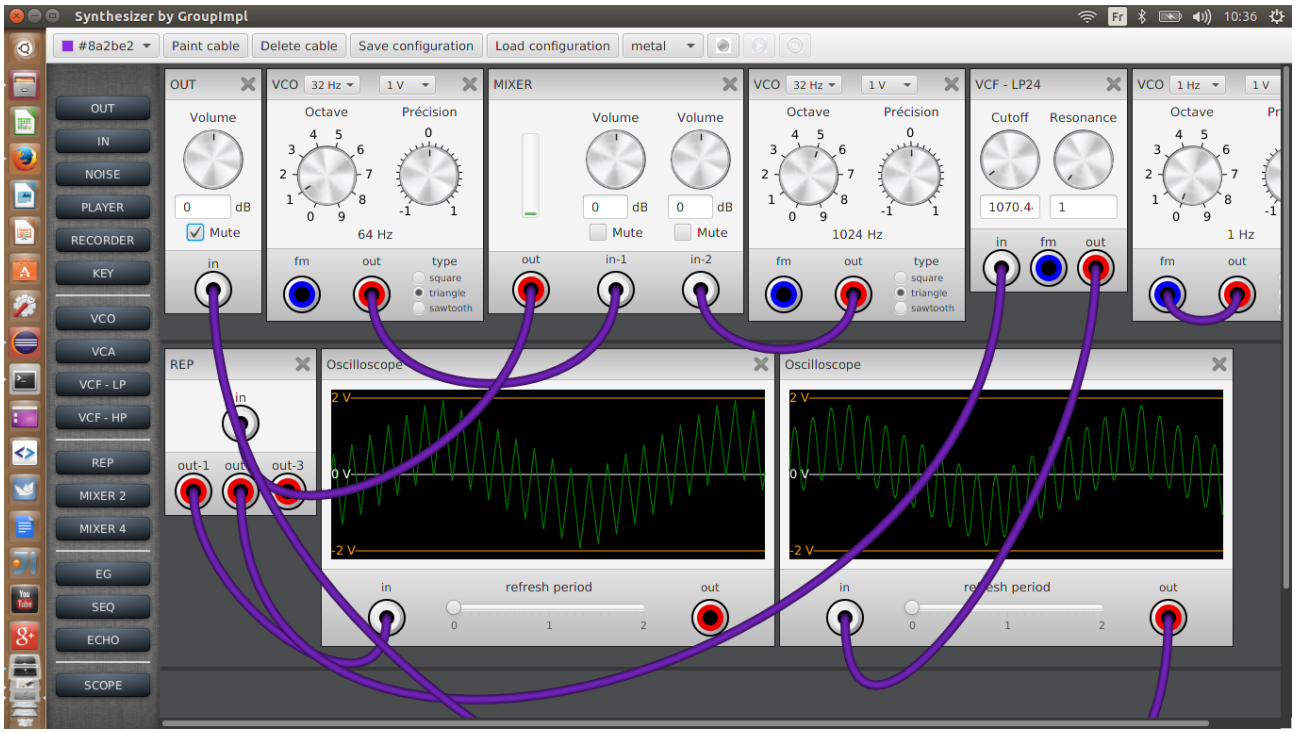
Pour assurer le bon fonctionnement d'un logiciel par rapport aux besoins et aux exigences recueillis par la spécification, les tests nous permettent d'assurer un niveau de qualité suffisant pendant le développement.
Les tests permettent notamment d'identifier les dysfonctionnements, les anomalies ou les régressions qui empêchent la couverture des exigences formulées.

Tests unitaires

Les tests unitaires sont partiels, comme le rapport de couverture de code le montre.
La validation a été faites par des vérifications méticuleuses de chaque module ainsi que par des tests fonctionnels.
Effectivement certaines classes ne sont pas ou très difficilement testables.

Tests fonctionnels

Voici un montage qui permet de tester le bon fonctionnement du filtre passe bas (LP24).
La sauvegarde de la configuration est le fichier :
Synthesizer/src/main/resources/configuration/test/test_VCF_LP24.sl



Sur l'oscilloscope de gauche on a un signal constitué de la somme de deux signaux en triangle : un de

fréquence 1024Hz et l'autre de fréquence 64Hz.
On peut visualiser les signaux séparément en cliquant sur les mutes du Mixer.
L'oscilloscope de droite représente le signal après être passé dans le composant LP24.
Tant que le Cutoff du LP24 est supérieur à 1024 le signal du passe-bas sera le même qu'à l'origine. Quand on le fait baisser progressivement on se rapproche du signal à 64Hz celui à 1024Hz étant éliminé par le filtre.

A l'aide du VCO qui est sur la droite en branchant le fm sur le fm du LP24, on peut faire varié le cutoff pris en compte. Pour 1 volt le cutoff sera multiplier par 2, pour -1 le cutoff sera divisé par 2.

Gestion des anomalies

La problématique de la gestion des anomalies ne nous est apparu qu'assez tardivement dans le projet si bien que nous n'avons rien mis en place de particulier.
Dans le cadre du projet, la systématisation de la vérification des modules juste après le développement et les tests unitaires nous a permis de pouvoir prendre conscience des dysfonctionnements et de les traiter assez rapidement.
On peut dire notre gestion des anomalies était basée sur la communication verbale, et sur la création de nouvelles users stories pour les dysfonctionnements les plus importants et les plus durables.
Nous sommes conscients que cette façon de faire n'était "jouable" que dans la mesure où le projet était de courte durée.

Couverture de code

La page html de la couverture de code est dans le fichier projectsynthesizer/jacoco/index.html.
En voici une capture d'écran partielle :

fr.istic.groupimpl.synthesizer.keyboard	0 %	0 %	22	22	58	58	18	18	3	3
fr.istic.groupimpl.synthesizer.cable	0 %	n/a	5	5	33	33	5	5	2	2
fr.istic.groupimpl.synthesizer.oscilloscope.jsyn	0 %	0 %	13	13	39	39	7	7	1	1
fr.istic.groupimpl.synthesizer.seq.jsyn	0 %	0 %	15	15	36	36	11	11	2	2
fr.istic.groupimpl.synthesizer.out	12 %	0 %	13	17	44	52	12	16	2	3
fr.istic.groupimpl.synthesizer.oscilloscope	0 %	n/a	16	16	41	41	16	16	3	3
fr.istic.groupimpl.synthesizer.echo.jsyn	0 %	0 %	8	8	31	31	6	6	1	1
fr.istic.groupimpl.synthesizer.player	32 %	0 %	18	26	43	65	16	24	2	3
fr.istic.groupimpl.synthesizer.vca	23 %	60 %	15	21	39	49	11	15	2	3
fr.istic.groupimpl.synthesizer.rep	0 %	0 %	17	17	36	36	14	14	3	3
fr.istic.groupimpl.synthesizer.eg.jsyn	79 %	72 %	14	38	27	124	3	13	0	2
fr.istic.groupimpl.synthesizer.io	0 %	n/a	3	3	22	22	3	3	1	1
fr.istic.groupimpl.synthesizer.whitenoise	0 %	n/a	12	12	24	24	12	12	3	3
fr.istic.groupimpl.synthesizer.linein	0 %	n/a	12	12	23	23	12	12	3	3
fr.istic.groupimpl.synthesizer.logger	30 %	n/a	10	15	20	28	10	15	0	1
fr.istic.groupimpl.synthesizer	0 %	n/a	3	3	14	14	3	3	1	1
fr.istic.groupimpl.synthesizer.player.jsyn	87 %	100 %	2	18	1	39	2	11	0	2
fr.istic.groupimpl.synthesizer.vcf.jsyn	98 %	67 %	2	9	1	31	1	7	0	1
fr.istic.groupimpl.synthesizer.rep.jsyn	97 %	88 %	1	8	1	23	0	4	0	1
fr.istic.groupimpl.synthesizer.vco.jsyn	100 %	100 %	0	23	0	67	0	19	0	2
fr.istic.groupimpl.synthesizer.mixer.jsyn	100 %	100 %	0	24	0	63	0	14	0	2
fr.istic.groupimpl.synthesizer.util.jsyn	100 %	88 %	1	19	0	43	0	15	0	2
fr.istic.groupimpl.synthesizer.vca.jsyn	100 %	100 %	0	9	0	23	0	7	0	1
fr.istic.groupimpl.synthesizer.recorder.jsyn	100 %	100 %	0	8	0	23	0	6	0	1
fr.istic.groupimpl.synthesizer.keyboard.jsyn	100 %	100 %	0	8	0	16	0	6	0	1
fr.istic.groupimpl.synthesizer.out.jsyn	100 %	n/a	0	4	0	13	0	4	0	1
Total	9 644 of 12 639	24 %	396 of 547	28 %	779	1 013	2 248	2 920	558	727	72	102

On peut voir que le nombre de lignes de code est 12639.
En utilisant une commande en ligne : wc -l `find . -name "*.java"` on trouve 14993, la différence est dû au fait que jacoco ne compte pas les commentaires.