"**Autoconf** est un outil permettant de rendre vos logiciels plus portables en exécutant une série de tests afin de découvrir les caractéristiques du système avant compilation."
"**Automake** est un outil qui génère, conformément à certains standards, les *Makefiles*. Il simplifie sensiblement le processus de description des règles de production du logiciel et exécute des tâches additionnelles telles que la gestion des dépendances inter fichiers sources."
"**Libtool** est une interface en ligne de commande pour la compilation et l'édition de liens. Il facilite, indépendamment de la plateforme, la production de bibliothèques statiques et partagées."
Lien utile : `http://sources.redhat.com/autobook/`

`Makefile.am`	`configure.in`	`main.c`

`## Commentaire`
`bin_PROGRAMS = hello`
`hello_SOURCES = main.c`
	`dnl utilise par autoconf`
	`dnl pour generer configure.`
	`AC_INIT(main.c)`
	`AM_INIT_AUTOMAKE(hello, 1.0)`
	`AC_PROG_CC`
	`AC_OUTPUT(Makefile)`
		`/* Hello World, amsi 2004 */`
		`#include <stdio.h>`
		`int main(int ac, char ** av) {`
		`  printf("Hello World\n") ;`
		`  return 0 ;`
		`}`

Makefile.am

- `bin_PROGRAMS` : Spécifie le nom des exécutables générés.
- `xxx_SOURCES` : Spécifie la liste des fichiers sources nécessaire à la création de l'exécutable `xxx` (ici `hello`).

configure.in

- `dnl` : *Do Not Link*, ligne de commentaire.
- `AC_*` : Macros spécifiques à `autoconf`.
- `AM_*` : Macros spécifiques à `automake`.
- `AC_INIT(fichier)` : Première macro appelée (obligatoire) ; vérifie que autoconf est lancé dans le bon répertoire en testant l'existance du fichier passé en argument.
- `AM_INIT_AUTOMAKE(package, version)` : Spécifie le nom et la version du package.
- `AC_PROG_CC` : Trouve le compilateur `C` approprié.
- `AC_OUTPUT(Makefile)` : Dernière macro appelée (obligatoire) ; spécifie dans le fichier script "`configure`" que le fichier généré sera "`Makefile`".

Création de l'archive du projet

```
bash$ aclocal
```

→ Génère un fichier "`aclocal.m4`" contenant les définitions des *macros* utilisées dans "`configure.in`".

```
bash$ autoconf
```

→ Génère le fichier script "`configure`".

```
bash$ automake -a
```

→ Génère le fichier "`Makefile.in`" et créé des liens vers les fichiers "`install-sh`", "`mkinstalldirs`", "`missing`", "`INSTALL`" et "`COPYING`". Les fichiers "`NEWS`", "`README`", "`AUTHORS`" et "`ChangeLog`" doivent être créés par l'utilisateur[1].

```
bash$ ./configure
```

→ Procède aux tests spécifiés dans "`configure.in`" et créé le fichier "`Makefile`".

```
bash$ make dist
```

→ Créé l'archive *prète à l'emploi* "`hello-1.0.tar.gz`" contenant tout le nécessaire pour compiler et installer le projet sur "toute architecture".

Devoir 02 (pour la semaine prochaine) : Écrire, en suivant le modèle donné en FIG1, les fichiers `Makefile.am` et `configure.in` du projet :
- `http://www.ai.univ-paris8.fr/~amsi/OD0405S1/dl/demoGL-1.0.0.tgz`
M'envoyer par email l'archive générée par `make dist` ainsi qu'un *screenshot* de l'exécution de l'application.

[1] Pour plus d'informations sur les formats des fichiers mentionnés, voir les *GNU coding standards* sur `http://www.gnu.org/prep/standards/`

```
## Makefile.am pour le projet demoGL

bin_PROGRAMS = demoGL
demoGL_SOURCES = main.c
CLEANFILES = *~
```

```
dnl Process this file with autoconf to produce a configure script.
AC_INIT(main.c)
dnl nom et version du package
AM_INIT_AUTOMAKE(demoGL, 1.0)
dnl Verifier le type de systeme
AC_CANONICAL_HOST
dnl Trouver le compilateur C approprie
AC_PROG_CC
dnl Verifie l'existance des headers C standards
AC_HEADER_STDC
dnl Verifie l'existance des headers passes en argument
dnl AC_MSG_ERROR affiche un message d'erreur \
dnl si les headers recherches n'existent pas
dnl AC_CHECK_HEADERS(GL/gl.h GL/glu.h GL/glut.h, , AC_MSG_ERROR(OpenGL headers not found))
CFLAGS="-Wall -O3"
LIB_GL_PATH=`locate libGL.so | sed -e "s/libGL.so.*//" -e "q"`
LIB_GLU_PATH=`locate libGLU.so | sed -e "s/libGLU.so.*//" -e "q"`
LIB_GLUT_PATH=`locate libglut.so | sed -e "s/libglut.so.*//" -e "q"`
dnl For each host type, set the appropriate options
dnl MACOSX
case "$host" in
*-*-darwin*)
        AC_DEFINE_UNQUOTED(MACOSX, 1, [If we are on macosx plateform.])
        LDFLAGS="$LDFLAGS -flat_namespace -undefined suppress -lz "
        LIBS="$LIBS -framework OpenGL -lobjc -framework GLUT"
dnl
;;
dnl CYGWIN
*-*-cygwin*)
        AC_DEFINE_UNQUOTED(CYGWIN,1, [If we are on cygwin plateform.])
        LIBS="$LIBS -lopengl32 -lglu32 -lglut32"
;;
dnl MINGW32
*-*-mingw32)
        AC_DEFINE_UNQUOTED(MINGW32,1, [If we are on mingw32 plateform.])
        LIBS="$LIBS -lopengl32 -lglu32 -lglut32"
;;
dnl Cross-compiling MINGW32
i386-pc-mingw32msvc)
        AC_DEFINE_UNQUOTED(MINGW32,1, [If we are on mingw32 plateform.])
        LIBS="$LIBS -lopengl32 -lglu32"
;;
dnl OTHER SYSTEMS
*)
        AC_CHECK_LIB(GL, glEnable, LIBS="$LIBS -L$LIB_GL_PATH -lGL")
        if test "$ac_cv_lib_GL_glEnable" = no ; then
                AC_CHECK_LIB(MesaGL, glEnable, LIBS="$LIBS -L$LIB_GL_PATH -lGL",
                        AC_MSG_ERROR(GL not found - please install GL or MesaGL))
        fi
        AC_CHECK_LIB(GLU, gluLookAt, LIBS="$LIBS -L$LIB_GLU_PATH -lGLU")
        if test "$ac_cv_lib_GLU_gluLookAt" = no ; then
                AC_CHECK_LIB(MesaGLU, gluLookAt, LIBS="$LIBS -L$LIB_GLU_PATH -lGLU",
                        AC_MSG_ERROR(GLU not found - please install GLU or MesaGLU))
        fi
        AC_CHECK_LIB(glut, glutInit, LIBS="$LIBS -L$LIB_GLUT_PATH -lglut",
                        AC_MSG_ERROR(glut not found - please install glut))
;;
esac
LIBS="-lm $LIBS"
AC_OUTPUT(Makefile)
```

FIG. 1 – Un exemple de fichier `Makefile.am` et `configure.in` pour un projet utilisant `OpenGL`.