

Code TIPE

Dossier Python :

Dossier Langues :

Module fr.json :

```
{
  "title": "Plane Simulation",
  "menu": {
    "display": {
      "title": "Affichage",
      "detach": "Détacher les contrôles",
      "attach": "Attacher les contrôles"
    },
    "settings": {
      "title": "Configuration",
      "speak": "Langues",
      "theme": {
        "title": "Thèmes",
        "light": "Clair",
        "dark": "Sombre",
        "uwu": "UwU",
        "system": "Système"
      }
    },
    "help": {
      "title": "Aide",
      "license": "Lire la licence"
    }
  },
  "control": {
    "title": "Contrôle des aéroports",
    "simple_add": {
      "title": "Des trucs"
    },
    "add_settings": {
      "runway" : "Nombre de piste",
      "airport" : "Nombre d'aéroport",
      "title": "Ajouter aéroport",
      "valid": "Ajouter"
    },
    "help": {
      "title": "Aide",
      "license": "Lire la licence"
    }
  },
  "file": {
    "license": "LICENCE_FR"
  },
  "warnings": {
    "64-bits": "Warning: attention c'est pas bien ! "
  },
}
```

```
"404": "Texte introuvable : regardez les logs."
}
```

Module aeroport.py :

Code sous licence GPL3+. Plus de détail a <<https://www.gnu.org/licenses/>> ou dans le fichier LICENSE

encoding = utf8

référencement de différents aéroports (sommets du graphe) représenté par des classes

```
import sys
import traceback
from typing import *
from math import *
from numpy import *
```

```
from settings import *
```

```
class Airport:
# to do : ajout des poids entres différents sommets.
    """Classe régissant les aéroports (noeuds)"""
    def __init__(self, nb_runway, id):
        self.nb_runway = nb_runway
        self.f_runway = self.nb_runway
        self.link = []
        self.id = id
    def is_free (self):
        """Vérifie si une piste est libre.
```

Returns:

```
bool: True s'il reste une piste de libre, False sinon.
    """
```

```
if self.f_runway > 0:
    return True
    return False
```

```
def take_runway(self):
    """Réserve une piste.
    """
```

```
self.f_runway -= 1
def free_runway(self):
    """Libère une piste réservé.
    """
```

```
self.f_runway +=1
def add_link (self, other_airport):
    """Rajoute un aeroport à la liste des aéroport avec une connection direct
```

Args:

```
other_airport (aeroport.Airport): Autre aéroport de la classe Airport du même fichier.
    """
```

```
self.link.append(other_airport)
def link_with (self):
```

""renvoie la liste des aéroports avec une connection directe.

Returns:

list: liste python contenant les aéroports liés.

""

return self.link

Module affichage.py :

Code sous liscence GPL3+. Plus de détail a <<https://www.gnu.org/licenses/>> ou dans le fichier LICENCE

encoding = utf8

import sys

import os

import json

from sys import *

from math import *

from functools import *

import des différentes librairies non-standard avec debug en cas de librairie manquante

try:

from PySide6.QtCore import *

from PySide6.QtWidgets import *

from PySide6.QtGui import *

Signal()

except ModuleNotFoundError as e:

print("le module PySide6 devrait être installé pour que ce programme puisse fonctionner, lisez README.md pour plus de détails", file=stderr)

raise e

import Python.settings as settings

import Python.langue as langue

app: QApplication = QApplication(sys.argv)

class Main_window(QMainWindow):

""Cette classe définit la fenêtre principale du programme, à partir d'un QWidget.""

changeLangSignal : Signal = Signal()

def __init__(self) -> None:

super().__init__()

self.setFocusPolicy(Qt.ClickFocus)

self.affichage_controles: bool = True

self.setWindowTitle(langue.get("title"))

#style

self.change_theme()

self.layout: QLayout = QVBoxLayout()

self.setLayout(self.layout)

self.layout.addSpacing(10)

Création des actions utilisables dans les menus

self.attach_detachAction: QAction = QAction(langue.get("menu.display.detach"), self)

```

self.attach_detachAction.triggered.connect(self.attach_detach_controles)
self.changeLangSignal.connect(self.attach_detach_texte)
self.langAction: list(QAction) = []
for speak in (("Français", "fr")):
self.langAction.append(QAction(speak[0], self))
self.langAction[-1].triggered.connect(partial(self.change_lang,speak[1]))
self.themeAction: list(QAction) = []
for theme in ("light", "dark", "system", "uwu"):
self.themeAction.append(QAction(langue.get("menu.settings.theme."+theme), self))
self.changeLangSignal.connect(langue.lazyEval(self.themeAction[-
1].setText,"menu.settings.theme."+theme))
self.themeAction[-1].triggered.connect(partial(self.change_theme,theme))

self.licenseAction: QAction = QAction(langue.get("menu.help.license"), self)
self.changeLangSignal.connect(langue.lazyEval(self.licenseAction.setText,"menu.help.license"))
self.licenseAction.triggered.connect(self.affich_licence)

# Création des menus
self.menuBar: QWidget = QMenuBar(self)
self.menuBar.setFixedWidth(self.size().width())

self.affichageMenu: QMenu = QMenu(langue.get("menu.display.title"), self.menuBar)
self.changeLangSignal.connect(langue.lazyEval(self.affichageMenu.setTitle,"menu.display.title"))
self.menuBar.addMenu(self.affichageMenu)
self.affichageMenu.addAction(self.attach_detachAction)
self.configMenu : QMenu = QMenu(langue.get("menu.settings.title"), self.menuBar)
self.changeLangSignal.connect(langue.lazyEval(self.configMenu.setTitle,"menu.settings.title"))
self.menuBar.addMenu(self.configMenu)
self.langMenu : QMenu = QMenu(langue.get("menu.settings.speak"), self.configMenu)
self.changeLangSignal.connect(langue.lazyEval(self.langMenu.setTitle,"menu.settings.speak"))
self.configMenu.addMenu(self.langMenu)
self.langMenu.addAction(self.langAction)
self.themeMenu : QMenu = QMenu(langue.get("menu.settings.theme.title"), self.configMenu)
self.changeLangSignal.connect(langue.lazyEval(self.themeMenu.setTitle,"menu.settings.theme.title
"))
self.configMenu.addMenu(self.themeMenu)
self.themeMenu.addAction(self.themeAction)

self.helpMenu: QMenu = QMenu(langue.get("menu.help.title"), self.menuBar)
self.changeLangSignal.connect(langue.lazyEval(self.helpMenu.setTitle,"menu.help.title"))
self.menuBar.addMenu(self.helpMenu)
self.helpMenu.addAction(self.licenseAction)
# Fenêtre détacheable de controle
self.affichage_controles: bool = True
self.controles = Controles()
self.layout.addWidget(self.controles)
self.controles.setFixedHeight(self.controles.minimumSizeHint().height())
self.changeLangSignal.connect(langue.lazyEval(self.controles.boutton1.setText,"control.simple_ad
d.title"))
self.changeLangSignal.connect(langue.lazyEval(self.controles.boutt_show_aj_air.setText,"control.a
dd_settings.title"))

```

```

def closeEvent(self, event) -> None:
    """Permet de fermer toutes les fenêtres lors de la fermeture de la fenêtre principale, et de terminer le
    programme"""
    app.exit(0)

def attach_detach_controles(self) -> None:
    """Permet d'afficher les controles dans une fenêtre séparée de la principale."""
    if self.affichage_controles:
        self.controles.hide()
        controles_graphiques.show()
        self.attach_detachAction.setText(langue.get("menu.display.attach"))
        self.affichage_controles = False

    else:
        self.controles.show()
        controles_graphiques.hide()
        self.attach_detachAction.setText(langue.get("menu.display.detach"))
        self.affichage_controles = True
        def attach_detach_texte(self):
            if self.affichage_controles:
                self.attach_detachAction.setText(langue.get("menu.display.detach"))
            else :
                self.attach_detachAction.setText(langue.get("menu.display.attach"))
        def change_lang(self, lang):
            settings.set("affichage.langue",lang)
            settings.save()
            langue.reload()
            self.changeLangSignal.emit()
        def change_theme(self, theme=None):
            style = ""
            if theme!=None:
                settings.set("affichage.theme",theme)
                settings.save()
                if settings.get("affichage.theme")=="dark":
                    style = ""
                    background-color: #262626;
                    color: #FFFFFF;
                    ""
                elif settings.get("affichage.theme")=="light":
                    style = ""
                    background-color: #DADADA;
                    color: #000000;
                    ""
                elif settings.get("affichage.theme")=="system":
                    style = ""
                    background-color: #1AFFE7;
                    color: #A100DE;
                    ""
                elif settings.get("affichage.theme")=="uwu":
                    style = ""

```

```

background-color: #DE00B6;
color: #36F5C6;
"""

self.setStyleSheet(style)
controles_graphiques.setStyleSheet(style)
Controles.fenetre_ajoute.setStyleSheet(style)

def affich_licence(self) -> None:
    """Cette fonction permet d'afficher la licence du projet"""
    self.fenetre_license: QWidget = QScrollArea()
    self.fenetre_license.setWindowTitle(langue.get("file.license"))
    try:
        path: str = os.path.abspath(os.path.dirname(__file__))
        path = os.path.join(path, langue.get("file.license"))
        with open(path, encoding="utf-8") as file:
            self.licenseTextlabel: QWidget = QLabel(file.read())
    except:
        print("The requested licence file was not found at", path, file=stderr)
        self.licenseTextlabel: QWidget = QLabel("Fichier manquant ou chemin cassé.")
    self.fenetre_license.setWidget(self.licenseTextlabel)
    self.fenetre_license.show()

class Controles(QWidget):
    """Classe permettant de gérer et afficher les contrôles d'ajout ainsi que leurs effets.

    Args:
        QWidget (class 'Shiboken.ObjectType'): permet l'utilisation de Widgets.
    """
    """Ce QWidget permet de gérer les différents contrôles."""

    fenetre_ajoute: QWidget = QScrollArea()
    fenetre_ajoute.setWindowTitle(langue.get("control.add_settings.title"))
    layout_aj_air: QLayout = QGridLayout()
    fenetre_ajoute.setLayout(layout_aj_air)

    amount = QSpinBox(minimum=0, maximum=10000, value=0)
    amountl = QLabel(langue.get("control.add_settings.runway"))
    layout_aj_air.addWidget(amountl, 0,0)
    layout_aj_air.addWidget(amount,0,1)

    nb_airport = QDoubleSpinBox(minimum=1, maximum=1, value=0,decimals=0)
    airport_label = QLabel(langue.get("control.add_settings.airport"))
    for i,widget in enumerate((airport_label, nb_airport, QLabel('+'),nb_airport)):
        layout_aj_air.addWidget(widget,1,i)
    nb_runway = QDoubleSpinBox(minimum=1, maximum=200, value=0,decimals=0)
    runway_label = QLabel(langue.get("control.add_settings.runway"))

    def ajouter_airport(*_) -> None:
        """Permet d'ajouter un aéroport.
        """

```

```

#Fenetre_principale.ajouter_airport(variable)

bouton_val_aj: QAbstractButton = QPushButton(langue.get("control.add_settings.valid"))
layout_aj_air.addWidget(bouton_val_aj)
bouton_val_aj.clicked.connect(ajouter_airport)

def __init__(self) -> None:
    """Méthode constructeur, permet de créer les boutons cliquables d'ajout d'aéroports.
    """
    super().__init__()
    self.setWindowTitle(langue.get("control.title"))
    # layout des controles widget
    self.layout = QHBoxLayout()
    self.setLayout(self.layout)

    self.bouton1: QAbstractButton = QPushButton(langue.get("control.simple_add.title"))
    self.bouton1.clicked.connect(Controles.ajouter_airport)
    self.layout.addWidget(self.bouton1)

    self.boutt_show_aj_air: QAbstractButton = QPushButton(langue.get("control.add_settings.title"))
    self.layout.addWidget(self.boutt_show_aj_air)
    self.boutt_show_aj_air.clicked.connect(self.fenetre_ajoute.show)

    controles_graphiques: QWidget = Controles()
    Fenetre_principale: QWidget = Main_window()

    Fenetre_principale.changeLangSignal.connect(langue.lazyEval(controles_graphiques.bouton_val_aj
    .setText,"control.add_settings.valid"))
    Fenetre_principale.changeLangSignal.connect(langue.lazyEval(controles_graphiques.fenetre_ajoute
    .setTitle,"control.add_settings.title"))
    for label,setloc in ((Controles.amountl,"control.add_settings.nb"),
    (Controles.airport_label,"control.add_settings.x"),
    (Controles.runway_label,"control.add_settings.y")):
        Fenetre_principale.changeLangSignal.connect(langue.lazyEval(label.setText,setloc))
    Fenetre_principale.changeLangSignal.connect(langue.lazyEval(controles_graphiques.bouton1.setT
    ext,"control.simple_add.title"))
    Fenetre_principale.changeLangSignal.connect(langue.lazyEval(controles_graphiques.boutt_show_a
    j_air.setText,"control.add_settings.title"))

    Fenetre_principale.showMaximized()

```

Module avion.py :

```

# Code sous liscence GPL3+. Plus de détail a <https://www.gnu.org/licenses/> ou dans le fichier
LICENSE
# encoding = utf8

# référencement de différents types d'avions représenté par des classes

import sys
import traceback

```

```

from typing import *
from math import *
from numpy import *

from settings import *
from graph import *
ledictionnairedesetops = { }

class Plane :
    """Classe régissant les différents types d'avions simulable."""
    def __init__(self, fuel_capacity, fuel_consumption, etops, position, speed) :
        """initializes

        Args : | fuel_Capacity of type float (kg)
        | fuel_consumption of type float (kg.s^-1)
        | etops of type int
        | position of type Airport*Airport*x∈[0;1]
        | speed of type int (m.s^-1)"""
        self.fuel_cap = fuel_capacity
        self.fuel = fuel_capacity
        self.fuel_cons = fuel_consumption
        self.etops = etops
        self.pos = position
        self.speed = speed
        self.dest = None
        self.chemin = []
        def new_destination(self, dest) :
            """Sets a new destination for the plane

            Args:
            dest (Airport): nouvelle destination
            """
            self.dest = dest
        def flight_time_left(self) :
            """returns flight capacity of self with current_fuel

            Returns:
            float: flight time capacity with current_fuel
            """
            return self.fuel/self.fuel_cons
        def etops_assert(self, route) :
            """checks if the plane has required etops for route

            Args:
            route (Airport*Airport): route trying to be taken

            Returns:
            bool: True if self has clearance for route, False otherwise

```



```

"""
if self.etops >= ledictionnairedesetops[route] : #to do : créer ledictionnairedesetops
return True
return False
def remove_s_fuel(self) :
"""removes 1s worth of fuel
"""
self.fuel = self.fuel-self.fuel_cons
def remove_min_fuel(self) :
"""removes 1min worth of fuel
"""
self.fuel = self.fuel-60*self.fuel_cons
def remove_h_fuel(self) :
"""removes 1h worth of fuel
"""
self.fuel = self.fuel-3600*self.fuel_cons
def ptick(self) :
"""simulates one tick for the plane
"""
newtruc = self.speed*60/graph[self.pos[0]][self.pos[1]]
if self.pos[2] + newtruc < 1 :
self.pos[2] += newtruc
else :
self.pos[0] = self.pos[1]
self.pos[1] = self.chemin[0]
self.chemin = self.chemin[1, :]
self.pos[2] = self.pos[2] + newtruc - 1

```

Module graph.py :

Code sous liscence GPL3+. Plus de détail a <<https://www.gnu.org/licenses/>> ou dans le fichier LICENSE

encoding = utf8

```

import sys
import traceback
from typing import *
from math import *
from numpy import *

```

```

from settings import *
from aeroport import *
from way import *

```

```

class Graph :
"""Airports link"""
def __init__(self, width) :
"""init de la class Graph

```

Args:

width (int): nombre de sommets (aéroports)

```

"""
self.n_airports = width # to do ajouter self.link un dictionnaire de dictionnaire contenant pour
chaque aeroport les liens direct avec les aeroports et leur distance (distance par rapport à soi-même
= 0).
self.values = [[inf for j in range(width)] for i in range(width)]
for i in range(self.n_airports) :
self.values[i][i] = 0
def new_airport(self) :
"""None --> Graph
Adds a new Airport to the Graph"""
self.n_airports += 1
self.values.append([-1 for i in range(self.n_airports)])
for i in range(self.n_airports-1) :
self.values[i].append(inf)
self.values[self.n_airports-1][self.n_airports-1] = 0
def add_link(self, airport1, airport2, cost) :
"""Airport_Id, Airport_Id, Int --> Graph
If no links exist between airport1 and airport2, adds a new link of weight cost between Airport1 and
Airport2"""
if self.values[airport1][airport2] == inf :
self.values[airport1][airport2] = cost
self.values[airport2][airport1] = cost
def __str__(self) :
"""affiche le nom des aéroports

```

Returns:

str: string représentant le graphe à afficher

```

"""
string = "\n"
for i in self.values :
string += "|"
for j in i :
string += str(j)
string += ", "
string = string[:-1]
string = string[:-1]
string += "\n"
return string

```

Module langue.py :

Code sous liscence GPL3+. Plus de détail a <<https://www.gnu.org/licenses/>> ou dans le fichier
LICENCE

encoding = utf8

import json

import os

from Python.settings import *

from functools import partial

lang: dict = { }

path: str = os.path.abspath(os.path.dirname(__file__))

```
path = os.path.join(path, "Langues/fr.json")
with open(path, 'r', encoding="utf-8") as setfile:
    lang = json.load(setfile)
```

```
def get(setloc: str) -> str:
    path: list[str]=setloc.split('.')
    try:
        temp = lang
        for key in path:
            temp = temp[key]
        return temp
    except (KeyError):
        print("Il semblerait que cette clef n'existe pas dans le fr.json vous allez devoir la rajouter ou requérir l'assistance d'une personne compétente.")
        return lang["404"]
```

```
def reload() -> None :
    global lang
    path: str = os.path.abspath(os.path.dirname(__file__))
    path = os.path.join(path, "Langues/" + settings.get("affichage.langue") + ".json")
    with open(path, 'r', encoding="utf-8") as setfile:
        lang = json.load(setfile)
    def __lazyInternal(function : callable, setloc : str) -> None:
        function(get(setloc))
```

```
def lazyEval(function : callable, setloc : str) -> callable:
    return partial(__lazyInternal,function,setloc)
```

Module LICENCE_FR :

Traduction en Français de la GNU GPL v3
Attention, traduction non officielle

This is an unofficial translation of the GNU General Public License into French. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL—only the original English text of the GNU GPL does that. However, we hope that this translation will help French speakers understand the GNU GPL better.

Ceci est une traduction non officielle de la GNU GPL (licence publique générale GNU) en français. Elle n'a pas été publiée par la Free Software Foundation, et n'établit pas juridiquement les termes de distribution des logiciels qui utilisent la GNU GPL – seul le texte anglais original de la GNU GPL le fait. Cependant, nous espérons que cette traduction aidera les francophones à mieux comprendre la GNU GPL.

LICENCE PUBLIQUE GÉNÉRALE GNU

Version 3, du 29 juin 2007.

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Chacun est autorisé à copier et distribuer des copies conformes de ce document de licence, mais toute modification en est proscrite.

Traduction française par Philippe Verdy
<verdy_p (à) wanadoo (point) fr>, le 30 juin 2007.

Avertissement important au sujet de cette traduction française.

Ceci est une traduction en français de la licence “GNU General Public License” (GPL). Cette traduction est fournie ici dans l’espoir qu’elle facilitera sa compréhension, mais elle ne constitue pas une traduction officielle ou approuvée d’un point de vue juridique.

La Free Software Foundation (FSF) ne publie pas cette traduction et ne l’a pas approuvée en tant que substitut valide au plan légal pour la licence authentique “GNU General Public License”. Cette traduction n’a pas encore été passée en revue attentivement par un juriste et donc le traducteur ne peut garantir avec certitude qu’elle représente avec exactitude la signification légale des termes de la licence authentique “GNU General Public License” publiée en anglais. Cette traduction n’établit donc légalement aucun des termes et conditions d’utilisation d’un logiciel sous licence GNU GPL — seul le texte original en anglais le fait. Si vous souhaitez être sûr que les activités que vous projetez seront autorisées par la GNU General Public License, veuillez vous référer à sa seule version anglaise authentique.

La FSF vous recommande fermement de ne pas utiliser cette traduction en tant que termes officiels pour vos propres programmes ; veuillez plutôt utiliser la version anglaise authentique telle que publiée par la FSF. Si vous choisissez d’acheminer cette traduction en même temps qu’un Programme sous licence GNU GPL, cela ne vous dispense pas de l’obligation d’acheminer en même temps une copie de la licence authentique en anglais, et de conserver dans la traduction cet avertissement important en français et son équivalent en anglais ci-dessous.

Important Warning About This French Translation.

This is a translation of the GNU General Public License (GPL) into French. This translation is distributed in the hope that it will facilitate understanding, but it is not an official or legally approved translation.

The Free Software Foundation (FSF) is not the publisher of this translation and has not approved it as a legal substitute for the

authentic GNU General Public License. The translation has not been reviewed carefully by lawyers, and therefore the translator cannot be sure that it exactly represents the legal meaning of the authentic GNU General Public License published in English. This translation does not legally state the terms and conditions of use of any Program licenced under GNU GPL — only the original English text of the GNU LGPL does that. If you wish to be sure whether your planned activities are permitted by the GNU General Public License, please refer to its sole authentic English version.

The FSF strongly urges you not to use this translation as the official distribution terms for your programs; instead, please use the authentic English version published by the FSF. If you choose to convey this translation along with a Program covered by the GPL Licence, this does not remove your obligation to convey at the same time a copy of the authentic GNU GPL License in English, and you must keep in this translation this important warning in English and its equivalent in French above.

Préambule

La Licence Publique Générale GNU (“GNU General Public License”) est une licence libre, en “copyleft”, destinée aux œuvres logicielles et d’autres types de travaux.

Les licences de la plupart des œuvres logicielles et autres travaux de pratique sont conçues pour ôter votre liberté de partager et modifier ces travaux. En contraste, la Licence Publique Générale GNU a pour but de garantir votre liberté de partager et changer toutes les versions d’un programme — afin d’assurer qu’il restera libre pour tous les utilisateurs. Nous, la Free Software Foundation, utilisons la Licence Publique Générale GNU pour la plupart de nos logiciels ; cela s’applique aussi à tout autre travail édité de cette façon par ses auteurs. Vous pouvez, vous aussi, l’appliquer à vos propres programmes.

Quand nous parlons de logiciel libre (“free”), nous nous référons à la liberté (“freedom”), pas au prix. Nos Licences Publiques Générales sont conçues pour assurer que vous ayez la liberté de distribuer des copies de logiciel libre (et le facturer si vous le souhaitez), que vous receviez le code source ou puissiez l’obtenir si vous le voulez, que vous puissiez modifier le logiciel ou en utiliser toute partie dans de nouveaux logiciels libres, et que vous sachiez que vous avez le droit de faire tout ceci.

Pour protéger vos droits, nous avons besoin d’empêcher que d’autres vous restreignent ces droits ou vous demande de leur abandonner ces droits. En conséquence, vous avez certaines responsabilités si vous distribuez des copies d’un tel programme ou si vous le modifiez : les responsabilités de respecter la liberté des autres.

Par exemple, si vous distribuez des copies d'un tel programme, que ce soit gratuit ou contre un paiement, vous devez accorder aux Destinataires les mêmes libertés que vous avez reçues. Vous devez aussi vous assurer qu'eux aussi reçoivent ou peuvent recevoir son code source. Et vous devez leur montrer les termes de cette licence afin qu'ils connaissent leurs droits.

Les développeurs qui utilisent la GPL GNU protègent vos droits en deux étapes : (1) ils affirment leur droits d'auteur ("copyright") sur le logiciel, et (2) vous accordent cette Licence qui vous donne la permission légale de le copier, le distribuer et/ou le modifier.

Pour la protection des développeurs et auteurs, la GPL stipule clairement qu'il n'y a pas de garantie pour ce logiciel libre. Aux fins à la fois des utilisateurs et auteurs, la GPL requière que les versions modifiées soient marquées comme changées, afin que leurs problèmes ne soient pas attribués de façon erronée aux auteurs des versions précédentes.

Certains dispositifs sont conçus pour empêcher l'accès des utilisateurs à l'installation ou l'exécution de versions modifiées du logiciel à l'intérieur de ces dispositifs, alors que les fabricants le peuvent. Ceci est fondamentalement incompatible avec le but de protéger la liberté des utilisateurs de modifier le logiciel. L'aspect systématique de tels abus se produit dans le secteur des produits destinés aux utilisateurs individuels, ce qui est précisément ce qui est le plus inacceptable. Aussi, nous avons conçu cette version de la GPL pour prohiber cette pratique pour ces produits. Si de tels problèmes surviennent dans d'autres domaines, nous nous tenons prêt à étendre cette restriction à ces domaines dans de futures versions de la GPL, autant qu'il sera nécessaire pour protéger la liberté des utilisateurs.

Finalement, chaque programme est constamment menacé par les brevets logiciels. Les États ne devraient pas autoriser de tels brevets à restreindre le développement et l'utilisation de logiciels libres sur des ordinateurs d'usage général ; mais dans ceux qui le font, nous voulons spécialement éviter le danger que les brevets appliqués à un programme libre puisse le rendre effectivement propriétaire. Pour empêcher ceci, la GPL assure que les brevets ne peuvent être utilisés pour rendre le programme non-libre.

Les termes précis et conditions concernant la copie, la distribution et la modification suivent.

TERMES ET CONDITIONS

Article 0. Définitions.

« Cette Licence » se réfère à la version 3 de la "GNU General Public License" (le texte original en anglais).

« Droit d'Auteur » signifie aussi les droits du "copyright" ou voisins qui s'appliquent à d'autres types de travaux, tels que ceux sur les

masques de semi-conducteurs.

« Le Programme » se réfère à tout travail qui peut être sujet au Droit d'Auteur ("copyright") et dont les droits d'utilisation sont concédés en vertu de cette Licence. Chacun des Licenciés, à qui cette Licence est concédée, est désigné par « vous. » Les « Licenciés » et les « Destinataires » peuvent être des personnes physiques ou morales (individus ou organisations).

« Modifier » un travail signifie en obtenir une copie et adapter tout ou partie du travail d'une façon nécessitant une autorisation d'un titulaire de Droit d'Auteur, autre que celle permettant d'en produire une copie conforme. Le travail résultant est appelé une « version modifiée » du précédent travail, ou un travail « basé sur » le précédent travail.

Un « Travail Couvert » signifie soit le Programme non modifié soit un travail basé sur le Programme.

« Propager » un travail signifie faire quoi que ce soit avec lui qui, sans permission, vous rendrait directement ou indirectement responsable d'un délit de contrefaçon suivant les lois relatives au Droit d'Auteur, à l'exception de son exécution sur un ordinateur ou de la modification d'une copie privée. La propagation inclue la copie, la distribution (avec ou sans modification), la mise à disposition envers le public, et aussi d'autres activités dans certains pays.

« Acheminer » un travail signifie tout moyen de propagation de celui-ci qui permet à d'autres parties de réaliser ou recevoir des copies. La simple interaction d'un utilisateur à travers un réseau informatique, sans transfert effectif d'une copie, ne constitue pas un acheminement.

Une interface utilisateur interactive affiche des « Notices Légales Appropriées » quand elle comprend un dispositif convenable, bien visible et évident qui (1) affiche une notice appropriée sur les droits d'auteur et (2) informe l'utilisateur qu'il n'y a pas de garantie pour le travail (sauf si des garanties ont été fournies hors du cadre de cette Licence), que les licenciés peuvent acheminer le travail sous cette Licence, et comment voir une copie de cette Licence. Si l'interface présente une liste de commandes utilisateur ou d'options, tel qu'un menu, un élément évident dans la liste présentée remplit ce critère.

Article 1. Code source.

Le « code source » d'un travail signifie la forme préférée du travail permettant ou facilitant les modifications de celui-ci. Le « code objet » d'un travail signifie toute forme du travail qui n'en est pas le code source.

Une « Interface Standard » signifie une interface qui est soit celle d'une norme officielle définie par un organisme de normalisation

reconnu ou, dans le cas des interfaces spécifiées pour un langage de programmation particulier, une interface largement utilisée parmi les développeurs travaillant dans ce langage.

Les « Bibliothèques Système » d'un travail exécutable incluent tout ce qui, en dehors du travail dans son ensemble, (a) est inclus dans la forme usuelle de packaging d'un Composant Majeur mais ne fait pas partie de ce Composant Majeur et (b) sert seulement à permettre l'utilisation du travail avec ce Composant Majeur ou à implémenter une Interface Standard pour laquelle une implémentation est disponible au public sous forme de code source ; un « Composant Majeur » signifie, dans ce contexte, un composant majeur essentiel (noyau, système de fenêtrage, etc.) du système d'exploitation (le cas échéant) d'un système sur lequel le travail exécutable fonctionne, ou bien un compilateur utilisé pour produire le code objet du travail, ou un interprète de code objet utilisé pour exécuter celui-ci.

Le « Source Correspondant » d'un travail sous forme de code objet signifie l'ensemble des codes sources nécessaires pour générer, installer et (dans le cas d'un travail exécutable) exécuter le code objet et modifier le travail, y compris les scripts pour contrôler ces activités. Cependant, cela n'inclue pas les Bibliothèques Système du travail, ni les outils d'usage général ou les programmes libres généralement disponibles qui peuvent être utilisés sans modification pour achever ces activités mais ne sont pas partie de ce travail. Par exemple le Source Correspondant inclut les fichiers de définition d'interfaces associés aux fichiers sources du travail, et le code source des bibliothèques partagées et des sous-routines liées dynamiquement, pour lesquelles le travail est spécifiquement conçu pour les requérir via, par exemple, des communications de données ou contrôles de flux internes entre ces sous-programmes et d'autres parties du travail.

Le Source Correspondant n'a pas besoin d'inclure tout ce que les utilisateurs peuvent régénérer automatiquement à partir d'autres parties du Source Correspondant.

Le Source Correspondant pour un travail sous forme de code source est ce même travail.

Article 2. Permissions de base.

Tous les droits accordés suivant cette Licence le sont jusqu'au terme des Droits d'Auteur ("copyright") sur le Programme, et sont irrévocables pourvu que les conditions établies soient remplies. Cette Licence affirme explicitement votre permission illimitée d'exécuter le Programme non modifié. La sortie produite par l'exécution d'un Travail Couvert n'est couverte par cette Licence que si cette sortie, étant donné leur contenu, constitue un Travail Couvert. Cette Licence reconnaît vos propres droits d'usage raisonnable ("fair use" en législation des États-Unis d'Amérique) ou autres équivalents, tels qu'ils sont pourvus par la loi applicable sur le Droit d'Auteur

(“copyright”).

Vous pouvez créer, exécuter et propager sans condition des Travaux Couverts que vous n’acheminez pas, aussi longtemps que votre licence demeure en vigueur. Vous pouvez acheminer des Travaux Couverts à d’autres personnes dans le seul but de leur faire réaliser des modifications à votre usage exclusif, ou pour qu’ils vous fournissent des facilités vous permettant d’exécuter ces travaux, pourvu que vous vous conformiez aux termes de cette Licence lors de l’acheminement de tout matériel dont vous ne contrôlez pas le Droit d’Auteur (“copyright”). Ceux qui, dès lors, réalisent ou exécutent pour vous les Travaux Couverts ne doivent alors le faire qu’exclusivement pour votre propre compte, sous votre direction et votre contrôle, suivant des termes qui leur interdisent de réaliser, en dehors de leurs relations avec vous, toute copie de votre matériel soumis au Droit d’Auteur.

L’acheminement dans toutes les autres circonstances n’est permis que selon les conditions établies ci-dessous. La concession de sous-licences n’est pas autorisé ; l’article 10 rend cet usage non nécessaire.

Article 3. Protection des droits légaux des utilisateurs envers les lois anti-contournement.

Aucun Travail Couvert ne doit être vu comme faisant partie d’une mesure technologique effective selon toute loi applicable remplissant les obligations prévues à l’article 11 du traité international sur le droit d’auteur adopté à l’OMPI le 20 décembre 1996, ou toutes lois similaires qui prohibent ou restreignent le contournement de telles mesures.

Si vous acheminez un Travail Couvert, vous renoncez à tout pouvoir légal d’interdire le contournement des mesures technologiques dans tous les cas où un tel contournement serait effectué en exerçant les droits prévus dans cette Licence pour ce Travail Couvert, et vous déclarez rejeter toute intention de limiter l’opération ou la modification du Travail, en tant que moyens de renforcer, à l’encontre des utilisateurs de ce Travail, vos droits légaux ou ceux de tierces parties d’interdire le contournement des mesures technologiques.

Article 4. Acheminement des copies conformes.

Vous pouvez acheminer des copies conformes du code source du Programme tel que vous l’avez reçu, sur n’importe quel support, pourvu que vous publiiez scrupuleusement et de façon appropriée sur chaque copie une notice de Droit d’Auteur appropriée ; gardez intactes toutes les notices établissant que cette Licence et tous les termes additionnels non permissifs ajoutés en accord avec l’article 7 s’appliquent à ce code ; et donnez à chacun des Destinataires une copie de cette Licence en même temps que le Programme.

Vous pouvez facturer à un prix quelconque, y compris gratuit, chacune des copies que vous acheminez, et vous pouvez offrir une protection additionnelle de support ou de garantie en échange d’un paiement.

Article 5. Acheminement des versions sources modifiées.

Vous pouvez acheminer un travail basé sur le Programme, ou bien les modifications pour le produire à partir du Programme, sous la forme de code source suivant les termes de l'article 4, pourvu que vous satisfassiez aussi à chacune des conditions requises suivantes :

- a) Le travail doit comporter des notices évidentes établissant que vous l'avez modifié et donnant la date correspondante.
- b) Le travail doit comporter des notices évidentes établissant qu'il est édité selon cette Licence et les conditions ajoutées d'après l'article 7. Cette obligation vient modifier l'obligation de l'article 4 de « garder intactes toutes les notices. »
- c) Vous devez licencier le travail entier, comme un tout, suivant cette Licence à quiconque entre en possession d'une copie. Cette Licence s'appliquera en conséquence, avec les termes additionnels applicables prévus par l'article 7, à la totalité du travail et chacune de ses parties, indépendamment de la façon dont ils sont emballés. Cette licence ne donne aucune permission de licencier le travail d'une autre façon, mais elle n'invalide pas une telle permission si vous l'avez reçue séparément.
- d) Si le travail a des interfaces utilisateurs interactives, chacune doit afficher les Notices Légales Appropriées ; cependant si le Programme a des interfaces qui n'affichent pas les Notices Légales Appropriées, votre travail n'a pas à les modifier pour qu'elles les affichent.

Une compilation d'un Travail Couvert avec d'autres travaux séparés et indépendants, qui ne sont pas par leur nature des extensions du Travail Couvert, et qui ne sont pas combinés avec lui de façon à former un programme plus large, dans ou sur un volume de stockage ou un support de distribution, est appelé un « agrégat » si la compilation et son Droit d'Auteur résultant ne sont pas utilisés pour limiter l'accès ou les droits légaux des utilisateurs de la compilation en deça de ce que permettent les travaux individuels. L'inclusion d'un Travail Couvert dans un agrégat ne cause pas l'application de cette Licence aux autres parties de l'agrégat.

Article 6. Acheminement des formes non sources.

Vous pouvez acheminer sous forme de code objet un Travail Couvert suivant les termes des articles 4 et 5, pourvu que vous acheminiez également suivant les termes de cette Licence le Source Correspondant lisible par une machine, d'une des façons suivantes :

- a) Acheminer le code objet sur, ou inclus dans, un produit physique (y compris un support de distribution physique), accompagné par le Source Correspondant fixé sur un support physique durable habituellement utilisé pour les échanges de logiciels.

b) Acheminer le code objet sur, ou inclus dans, un produit physique (y compris un support de distribution physique), accompagné d'une offre écrite, valide pour au moins trois années et valide pour aussi longtemps que vous fournissez des pièces de rechange ou un support client pour ce modèle de produit, afin de donner à quiconque possède le code objet soit (1) une copie du Source Correspondant à tout logiciel dans ce produit qui est couvert par cette Licence, sur un support physique durable habituellement utilisé pour les échanges de logiciels, pour un prix non supérieur au coût raisonnable de la réalisation physique de l'acheminement de la source, ou soit (2) un accès permettant de copier le Source Correspondant depuis un serveur réseau sans frais.

c) Acheminer des copies individuelles du code objet avec une copie de l'offre écrite de fournir le Source Correspondant. Cette alternative est permise seulement occasionnellement et non commercialement, et seulement si vous avez reçu le code objet avec une telle offre, en accord avec l'article 6 alinéa b.

d) Acheminer le code objet en offrant un accès depuis un emplacement désigné (gratuit ou contre facturation) et offrir un accès équivalent au Source Correspondant de la même façon via le même emplacement et sans facturation supplémentaire. Vous n'avez pas besoin d'obliger les Destinataires à copier le Source Correspondant en même temps que le code objet. Si l'emplacement pour copier le code objet est un serveur réseau, le Source Correspondant peut être sur un serveur différent (opéré par vous ou par un tiers) qui supporte des facilités équivalentes de copie, pourvu que vous mainteniez des directions claires à proximité du code objet indiquant où trouver le Source Correspondant. Indépendamment de quel serveur héberge le Source Correspondant, vous restez obligé de vous assurer qu'il reste disponible aussi longtemps que nécessaire pour satisfaire à ces obligations.

e) Acheminer le code objet en utilisant une transmission d'égal-à-égal, pourvu que vous informiez les autres participants sur où le code objet et le Source Correspondant du travail sont offerts sans frais au public général suivant l'article 6 alinéa d. Une portion séparable du code objet, dont le code source est exclu du Source Correspondant en tant que Bibliothèque Système, n'a pas besoin d'être inclu dans l'acheminement du travail sous forme de code objet.

Un « Produit Utilisateur » est soit (1) un « Produit de Consommation, » ce qui signifie toute propriété personnelle tangible normalement utilisée à des fins personnelles, familiales ou relatives au foyer, soit (2) toute chose conçue ou vendue pour l'incorporation dans un lieu d'habitation. Pour déterminer si un produit constitue un Produit de Consommation, les cas ambigus sont résolus en fonction de la

couverture. Pour un produit particulier reçu par un utilisateur particulier, l'expression « normalement utilisée » ci-avant se réfère à une utilisation typique ou l'usage commun de produits de même catégorie, indépendamment du statut de cet utilisateur particulier ou de la façon spécifique dont cet utilisateur particulier utilise effectivement ou s'attend lui-même ou est attendu à utiliser ce produit. Un produit est un Produit de Consommation indépendamment du fait que ce produit a ou n'a pas d'utilisations substantielles commerciales, industrielles ou hors Consommation, à moins que de telles utilisations représentent le seul mode significatif d'utilisation du produit.

Les « Informations d'Installation » d'un Produit Utilisateur signifient toutes les méthodes, procédures, clés d'autorisation ou autres informations requises pour installer et exécuter des versions modifiées d'un Travail Couvert dans ce Produit Utilisateur à partir d'une version modifiée de son Source Correspondant. Les informations qui suffisent à assurer la continuité de fonctionnement du code objet modifié ne doivent en aucun cas être empêchées ou interférées du seul fait qu'une modification a été effectuée.

Si vous acheminez le code objet d'un Travail Couvert dans, ou avec, ou spécifiquement pour l'utilisation dans, un Produit Utilisateur et l'acheminement se produit en tant qu'élément d'une transaction dans laquelle le droit de possession et d'utilisation du Produit Utilisateur est transféré au Destinataire définitivement ou pour un terme fixé (indépendamment de la façon dont la transaction est caractérisée), le Source Correspondant acheminé selon cet article-ci doit être accompagné des Informations d'Installation. Mais cette obligation ne s'applique pas si ni vous ni aucune tierce partie ne détient la possibilité d'installer un code objet modifié sur le Produit Utilisateur (par exemple, le travail a été installé en mémoire morte).

L'obligation de fournir les Informations d'Installation n'inclue pas celle de continuer à fournir un service de support, une garantie ou des mises à jour pour un travail qui a été modifié ou installé par le Destinataire, ou pour le Produit Utilisateur dans lequel il a été modifié ou installé. L'accès à un réseau peut être rejeté quand la modification elle-même affecte matériellement et défavorablement les opérations du réseau ou viole les règles et protocoles de communication au travers du réseau.

Le Source Correspondant acheminé et les Informations d'Installation fournies, en accord avec cet article, doivent être dans un format publiquement documenté (et dont une implémentation est disponible auprès du public sous forme de code source) et ne doit nécessiter aucune clé ou mot de passe spécial pour le dépaquetage, la lecture ou la copie.

Article 7. Termes additionnels.

Les « permissions additionnelles » désignent les termes qui

supplémentent ceux de cette Licence en émettant des exceptions à l'une ou plusieurs de ses conditions. Les permissions additionnelles qui sont applicables au Programme entier doivent être traitées comme si elles étaient inclues dans cette Licence, dans les limites de leur validité suivant la loi applicable. Si des permissions additionnelles s'appliquent seulement à une partie du Programme, cette partie peut être utilisée séparément suivant ces permissions, mais le Programme tout entier reste gouverné par cette Licence sans regard aux permissions additionnelles.

Quand vous acheminez une copie d'un Travail Couvert, vous pouvez à votre convenance ôter toute permission additionnelle de cette copie, ou de n'importe quelle partie de celui-ci. (Des permissions additionnelles peuvent être rédigées de façon à requérir leur propre suppression dans certains cas où vous modifiez le travail.) Vous pouvez placer les permissions additionnelles sur le matériel acheminé, ajoutées par vous à un Travail Couvert pour lequel vous avez ou pouvez donner les permissions de Droit d'Auteur ("copyright") appropriées.

Nonobstant toute autre clause de cette Licence, pour tout constituant que vous ajoutez à un Travail Couvert, vous pouvez (si autorisé par les titulaires de Droit d'Auteur pour ce constituant) compléter les termes de cette Licence avec des termes :

- a) qui rejettent la garantie ou limitent la responsabilité de façon différente des termes des articles 15 et 16 de cette Licence ; ou
- b) qui requièrent la préservation de notices légales raisonnables spécifiées ou les attributions d'auteur dans ce constituant ou dans les Notices Légales Appropriées affichées par les travaux qui le contiennent ; ou
- c) qui prohibent la représentation incorrecte de l'origine de ce constituant, ou qui requièrent que les versions modifiées d'un tel constituant soit marquées par des moyens raisonnables comme différentes de la version originale ; ou
- d) qui limitent l'usage à but publicitaire des noms des concédants de licence et des auteurs du constituant ; ou
- e) qui refusent à accorder des droits selon la législation relative aux marques commerciales, pour l'utilisation dans des noms commerciaux, marques commerciales ou marques de services ; ou
- f) qui requièrent l'indemnisation des concédants de licences et auteurs du constituant par quiconque achemine ce constituant (ou des versions modifiées de celui-ci) en assumant contractuellement la responsabilité envers le Destinataire, pour toute responsabilité que ces engagements contractuels imposent directement à ces octroyants de licences et auteurs.

Tous les autres termes additionnels non permissifs sont considérés comme des « restrictions avancées » dans le sens de l'article 10. Si le Programme tel que vous l'avez reçu, ou toute partie de celui-ci, contient une notice établissant qu'il est gouverné par cette Licence en même temps qu'un terme qui est une restriction avancée, vous pouvez ôter ce terme. Si un document de licence contient une restriction avancée mais permet la reconcession de licence ou l'acheminement suivant cette Licence, vous pouvez ajouter un Travail Couvert constituant gouverné par les termes de ce document de licence, pourvu que la restriction avancée ne survive pas à une telle cession de licence ou acheminement.

Si vous ajoutez des termes à un Travail Couvert en accord avec cet article, vous devez placer, dans les fichiers sources appropriés, une déclaration des termes additionnels qui s'appliquent à ces fichiers, ou une notice indiquant où trouver les termes applicables.

Les termes additionnels, qu'ils soient permissifs ou non permissifs, peuvent être établis sous la forme d'une licence écrite séparément, ou établis comme des exceptions ; les obligations ci-dessus s'appliquent dans chacun de ces cas.

Article 8. Terminaison.

Vous ne pouvez ni propager ni modifier un Travail Couvert autrement que suivant les termes de cette Licence. Toute autre tentative de le propager ou le modifier est nulle et terminera automatiquement vos droits selon cette Licence (y compris toute licence de brevet accordée selon le troisième paragraphe de l'article 11).

Cependant, si vous cessez toute violation de cette Licence, alors votre licence depuis un titulaire de Droit d'Auteur ("copyright") est réinstaurée (a) à titre provisoire à moins que et jusqu'à ce que le titulaire de Droit d'Auteur termine finalement et explicitement votre licence, et (b) de façon permanente si le titulaire de Droit d'Auteur ne parvient pas à vous notifier de la violation par quelque moyen raisonnable dans les soixante (60) jours après la cessation.

De plus, votre licence depuis un titulaire particulier de Droit d'Auteur est réinstaurée de façon permanente si ce titulaire vous notifie de la violation par quelque moyen raisonnable, c'est la première fois que vous avez reçu une notification de violation de cette Licence (pour un travail quelconque) depuis ce titulaire de Droit d'Auteur, et vous résolvez la violation dans les trente (30) jours qui suivent votre réception de la notification.

La terminaison de vos droits suivant cette section ne terminera pas les licences des parties qui ont reçu des copies ou droits de votre part suivant cette Licence. Si vos droits ont été terminés et non réinstaurés de façon permanente, vous n'êtes plus qualifié à recevoir de nouvelles licences pour les mêmes constituants selon l'article 10.

Article 9. Acceptation non requise pour obtenir des copies.

Vous n'êtes pas obligé d'accepter cette licence afin de recevoir ou exécuter une copie du Programme. La propagation asservie d'un Travail Couvert qui se produit simplement en conséquence d'une transmission d'égal-à-égal pour recevoir une copie ne nécessite pas l'acceptation. Cependant, rien d'autre que cette Licence ne vous accorde la permission de propager ou modifier un quelconque Travail Couvert. Ces actions enfreignent le Droit d'Auteur si vous n'acceptez pas cette Licence. Par conséquent, en modifiant ou propageant un Travail Couvert, vous indiquez votre acceptation de cette Licence pour agir ainsi.

Article 10. Cession automatique de Licence aux Destinataires et intermédiaires.

Chaque fois que vous acheminez un Travail Couvert, le Destinataire reçoit automatiquement une licence depuis les concédants originaux, pour exécuter, modifier et propager ce travail, suivant les termes de cette Licence. Vous n'êtes pas responsable du renforcement de la conformation des tierces parties avec cette Licence.

Une « transaction d'entité » désigne une transaction qui transfère le contrôle d'une organisation, ou de substantiellement tous ses actifs, ou la subdivision d'une organisation, ou la fusion de plusieurs organisations. Si la propagation d'un Travail Couvert résulte d'une transaction d'entité, chaque partie à cette transaction qui reçoit une copie du travail reçoit aussi les licences pour le travail que le prédécesseur intéressé à cette partie avait ou pourrait donner selon le paragraphe précédent, plus un droit de possession du Source Correspondant de ce travail depuis le prédécesseur intéressé si ce prédécesseur en dispose ou peut l'obtenir par des efforts raisonnables.

Vous ne pouvez imposer aucune restriction avancée dans l'exercice des droits accordés ou affirmés selon cette Licence. Par exemple, vous ne pouvez imposer aucun paiement pour la licence, aucune royauté, ni aucune autre charge pour l'exercice des droits accordés selon cette Licence ; et vous ne pouvez amorcer aucun litige judiciaire (y compris une réclamation croisée ou contre-réclamation dans un procès) sur l'allégation qu'une revendication de brevet est enfreinte par la réalisation, l'utilisation, la vente, l'offre de vente, ou l'importation du Programme ou d'une quelconque portion de celui-ci.

Article 11. Brevets.

Un « contributeur » est un titulaire de Droit d'Auteur ("copyright") qui autorise l'utilisation selon cette Licence du Programme ou du travail sur lequel le Programme est basé. Le travail ainsi soumis à licence est appelé la « version contributive » de ce contributeur.

Les « revendications de brevet essentielles » sont toutes les revendications de brevets détenues ou contrôlées par le contributeur, qu'elles soient déjà acquises par lui ou acquises subséquemment, qui pourraient être enfreintes de quelque manière, permises par cette Licence, sur la réalisation, l'utilisation ou la vente de la version

contributive de celui-ci. Aux fins de cette définition, le « contrôle » inclue le droit de concéder des sous-licences de brevets d'une manière consistante, nécessaire et suffisante, avec les obligations de cette Licence.

Chaque contributeur vous accorde une licence de brevet non exclusive, mondiale et libre de toute royauté, selon les revendications de brevet essentielles, pour réaliser, utiliser, vendre, offrir à la vente, importer et autrement exécuter, modifier et propager les contenus de sa version contributive.

Dans les trois paragraphes suivants, une « licence de brevet » désigne tous les accords ou engagements exprimés, quel que soit le nom que vous lui donnez, de ne pas mettre en vigueur un brevet (telle qu'une permission explicite pour mettre en pratique un brevet, ou un accord pour ne pas poursuivre un Destinataire pour cause de violation de brevet). « Accorder » une telle licence de brevet à une partie signifie conclure un tel accord ou engagement à ne pas faire appliquer le brevet à cette partie.

Si vous acheminez un Travail Couvert, dépendant en connaissance d'une licence de brevet, et si le Source Correspondant du travail n'est pas disponible à quiconque copie, sans frais et suivant les termes de cette Licence, à travers un serveur réseau publiquement accessible ou tout autre moyen immédiatement accessible, alors vous devez soit (1) rendre la Source Correspondante ainsi disponible, soit (2) vous engager à vous priver pour vous-même du bénéfice de la licence de brevet pour ce travail particulier, soit (3) vous engager, d'une façon consistante avec les obligations de cette Licence, à étendre la licence de brevet aux Destinataires de ce travail. « Dépendant en connaissance » signifie que vous avez effectivement connaissance que, selon la licence de brevet, votre acheminement du Travail Couvert dans un pays, ou l'utilisation du Travail Couvert par votre Destinataire dans un pays, infreindrait un ou plusieurs brevets identifiables dans ce pays où vous avez des raisons de penser qu'ils sont valides.

Si, conformément à ou en liaison avec une même transaction ou un même arrangement, vous acheminez, ou propagez en procurant un acheminement de, un Travail Couvert et accordez une licence de brevet à l'une des parties recevant le Travail Couvert pour lui permettre d'utiliser, propager, modifier ou acheminer une copie spécifique du Travail Couvert, alors votre accord est automatiquement étendu à tous les Destinataires du Travail Couvert et des travaux basés sur celui-ci.

Une licence de brevet est « discriminatoire » si, dans le champ de sa couverture, elle n'inclut pas un ou plusieurs des droits qui sont spécifiquement accordés selon cette Licence, ou en prohibe l'exercice, ou est conditionnée par le non-exercice d'un ou plusieurs de ces droits. Vous ne pouvez pas acheminer un Travail Couvert si vous êtes partie à un arrangement selon lequel une partie tierce exerçant son activité dans la distribution de logiciels et à laquelle vous effectuez

un paiement fondé sur l'étendue de votre activité d'acheminement du travail, et selon lequel la partie tierce accorde, à une quelconque partie qui recevrait depuis vous le Travail Couvert, une licence de brevet discriminatoire (a) en relation avec les copies du Travail Couvert acheminées par vous (ou les copies réalisées à partir de ces copies), ou (b) avant tout destinée et en relation avec des produits spécifiques ou compilations contenant le Travail Couvert, à moins que vous ayez conclu cet arrangement ou que la licence de brevet ait été accordée avant le 28 mars 2007.

Rien dans cette Licence ne devrait être interprété comme devant exclure ou limiter toute licence implicite ou d'autres moyens de défense à une infraction qui vous seraient autrement disponible selon la loi applicable relative aux brevets.

Article 12. Non abandon de la liberté des autres.

Si des conditions vous sont imposées (que ce soit par décision judiciaire, par un accord ou autrement) qui contredisent les conditions de cette Licence, elles ne vous excusent pas des conditions de cette Licence. Si vous ne pouvez pas acheminer un Travail Couvert de façon à satisfaire simultanément vos obligations suivant cette Licence et toutes autres obligations pertinentes, alors en conséquence vous ne pouvez pas du tout l'acheminer. Par exemple, si vous avez un accord sur des termes qui vous obligent à collecter pour le réacheminement des royalties depuis ceux à qui vous acheminez le Programme, la seule façon qui puisse vous permettre de satisfaire à la fois à ces termes et ceux de cette Licence sera de vous abstenir entièrement d'acheminer le Programme.

Article 13. Utilisation avec la Licence Générale Publique Affero GNU.

Nonobstant toute autre clause de cette Licence, vous avez la permission de lier ou combiner tout Travail Couvert avec un travail placé sous la version 3 de la Licence Générale Publique GNU Affero ("GNU Affero General Public License") en un seul travail combiné, et d'acheminer le travail résultant. Les termes de cette Licence continueront à s'appliquer à la partie formant un Travail Couvert, mais les obligations spéciales de la Licence Générale Publique GNU Affero, article 13, concernant l'interaction à travers un réseau s'appliqueront à la combinaison en tant que telle.

Article 14. Versions révisées de cette License.

La Free Software Foundation peut publier des versions révisées et/ou nouvelles de la Licence Publique Générale GNU ("GNU General Public License") de temps en temps. De telles version nouvelles resteront similaires dans l'esprit avec la présente version, mais peuvent différer dans le détail afin de traiter de nouveaux problèmes ou préoccupations.

Chaque version reçoit un numéro de version distinctif. Si le Programme indique qu'une version spécifique de la Licence Publique Générale GNU « ou toute version ultérieure » ("or any later version") s'applique à

celui-ci, vous avez le choix de suivre soit les termes et conditions de cette version numérotée, soit ceux de n'importe quelle version publiée ultérieurement par la Free Software Foundation. Si le Programme n'indique pas une version spécifique de la Licence Publique Générale GNU, vous pouvez choisir l'une quelconque des versions qui ont été publiées par la Free Software Foundation.

Si le Programme spécifie qu'un intermédiaire peut décider quelles versions futures de la Licence Générale Publique GNU peut être utilisée, la déclaration publique d'acceptation d'une version par cet intermédiaire vous autorise à choisir cette version pour le Programme.

Des versions ultérieures de la licence peuvent vous donner des permissions additionnelles ou différentes. Cependant aucune obligation additionnelle n'est imposée à l'un des auteurs ou titulaires de Droit d'Auteur du fait de votre choix de suivre une version ultérieure.
Article 15. Déclaration d'absence de garantie.

IL N'Y A AUCUNE GARANTIE POUR LE PROGRAMME, DANS LES LIMITES PERMISES PAR LA LOI APPLICABLE. À MOINS QUE CELA NE SOIT ÉTABLI DIFFÉREMMENT PAR ÉCRIT, LES PROPRIÉTAIRES DE DROITS ET/OU LES AUTRES PARTIES FOURNISSENT LE PROGRAMME « EN L'ÉTAT » SANS GARANTIE D'AUCUNE SORTE, QU'ELLE SOIT EXPRIMÉE OU IMPLICITE, CECI COMPRENANT, SANS SE LIMITER À CELLES-CI, LES GARANTIES IMPLICITES DE COMMERCIALISABILITÉ ET D'ADÉQUATION À UN OBJECTIF PARTICULIER. VOUS ASSUMEZ LE RISQUE ENTIER CONCERNANT LA QUALITÉ ET LES PERFORMANCES DU PROGRAMME. DANS L'ÉVENTUALITÉ OÙ LE PROGRAMME S'AVÉRERAIT DÉFECTUEUX, VOUS ASSUMEZ LES COÛTS DE TOUS LES SERVICES, RÉPARATIONS OU CORRECTIONS NÉCESSAIRES.
Article 16. Limitation de responsabilité.

EN AUCUNE AUTRE CIRCONSTANCE QUE CELLES REQUISES PAR LA LOI APPLICABLE OU ACCORDÉES PAR ÉCRIT, UN TITULAIRE DE DROITS SUR LE PROGRAMME, OU TOUT AUTRE PARTIE QUI MODIFIE OU ACHEMINE LE PROGRAMME COMME PERMIS CI-DESSUS, NE PEUT ÊTRE TENU POUR RESPONSABLE ENVERS VOUS POUR LES DOMMAGES, INCLUANT TOUT DOMMAGE GÉNÉRAL, SPÉCIAL, ACCIDENTEL OU INDUIT SURVENANT PAR SUITE DE L'UTILISATION OU DE L'INCAPACITÉ D'UTILISER LE PROGRAMME (Y COMPRIS, SANS SE LIMITER À CELLES-CI, LA PERTE DE DONNÉES OU L'INEXACTITUDE DES DONNÉES RETOURNÉES OU LES PERTES SUBIES PAR VOUS OU DES PARTIES TIERCES OU L'INCAPACITÉ DU PROGRAMME À FONCTIONNER AVEC TOUT AUTRE PROGRAMME), MÊME SI UN TEL TITULAIRE OU TOUTE AUTRE PARTIE A ÉTÉ AVISÉ DE LA POSSIBILITÉ DE TELS DOMMAGES.
Article 17. Interprétation des sections 15 et 16.

Si la déclaration d'absence de garantie et la limitation de responsabilité fournies ci-dessus ne peuvent prendre effet localement selon leurs termes, les cours de justice qui les examinent doivent appliquer la législation locale qui approche au plus près possible une levée absolue de toute responsabilité civile liée au Programme, à moins qu'une garantie ou assumption de responsabilité accompagne une copie du Programme en échange d'un paiement.

FIN DES TERMES ET CONDITIONS.

Comment appliquer ces termes à vos nouveaux programmes

Si vous développez un nouveau programme et voulez qu'il soit le plus possible utilisable par le public, la meilleure façon d'y parvenir et d'en faire un logiciel libre que chacun peut redistribuer et changer suivant ces termes-ci.

Pour appliquer ces termes, attachez les notices suivantes au programme. Il est plus sûr de les attacher au début de chacun des fichiers sources afin de transporter de façon la plus effective possible l'exclusion de garantie ; et chaque fichier devrait comporter au moins la ligne de réservation de droit ("copyright") et une indication permettant de savoir où la notice complète peut être trouvée :

<une ligne donnant le nom du programme et une brève idée de ce qu'il fait.>
Copyright (C) <année> <nom de l'auteur> — Tous droits réservés.

Ce programme est un logiciel libre ; vous pouvez le redistribuer ou le modifier suivant les termes de la "GNU General Public License" telle que publiée par la Free Software Foundation : soit la version 3 de cette licence, soit (à votre gré) toute version ultérieure.

Ce programme est distribué dans l'espoir qu'il vous sera utile, mais SANS AUCUNE GARANTIE : sans même la garantie implicite de COMMERCIALISABILITÉ ni d'ADÉQUATION À UN OBJECTIF PARTICULIER. Consultez la Licence Générale Publique GNU pour plus de détails.

Vous devriez avoir reçu une copie de la Licence Générale Publique GNU avec ce programme ; si ce n'est pas le cas, consultez :
<<http://www.gnu.org/licenses/>>.

Ajoutez également les informations permettant de vous contacter par courrier électronique ou postal.

Si le programme produit une interaction sur un terminal, faites lui afficher une courte notice comme celle-ci lors de son démarrage en mode interactif :

<programme> Copyright (C) <année> <nom de l'auteur>
Ce programme vient SANS ABSOLUMENT AUCUNE GARANTIE ; taper "affiche g" pour les détails. Ceci est un logiciel libre et vous êtes invité à le redistribuer suivant certaines conditions ; taper "affiche c" pour les détails.

Les commandes hypothétiques "affiche g" and "affiche c" devrait afficher les parties appropriées de la Licence Générale Publique. Bien sûr, les commandes de votre programme peuvent être différentes ; pour une interface graphique, vous pourriez utiliser une « boîte À propos. »

Vous devriez également obtenir de votre employeur (si vous travaillez en tant que programmeur) ou de votre école un « renoncement aux droits de propriété » pour ce programme, si nécessaire. Pour plus d'informations à ce sujet, et comment appliquer la GPL GNU, consultez <http://www.gnu.org/licenses/>.

La Licence Générale Publique GNU ne permet pas d'incorporer votre programme dans des programmes propriétaires. Si votre programme est une bibliothèque de sous-routines, vous pourriez considérer qu'il serait plus utile de permettre de lier des applications propriétaires avec la bibliothèque. Si c'est ce que vous voulez faire, utilisez la Licence Générale Publique Limitée GNU au lieu de cette Licence ; mais d'abord, veuillez lire <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Module main.py :

```
# Code sous liscence GPL3+. Plus de détail a <https://www.gnu.org/licenses/> ou dans le fichier
LICENSE
# encoding=utf8

import sys
import os
from typing import NoReturn
import importlib
import Python.affichage

def launch_app() -> NoReturn:
sys.exit(Python.affichage.app.exec())
```

Module settings.py :

```
# Code sous liscence GPL3+. Plus de détail a <https://www.gnu.org/licenses/> ou dans le fichier
LICENSE
# encoding = utf8

import json
import os
from typing import *
from sys import *

settings: dict = { }
defaults: dict = { }

try:
path: str = os.path.abspath(os.path.dirname(__file__))
path = os.path.join(path, "../vscode/default_settings.json")
with open(path, 'r', encoding="utf-8") as setfile:
defaults = json.load(setfile)
except Exception as e:
print("no default_settings.json file, package should be réinstallled or permission checked, aborting",
file=stderr)
```

```

raise FileNotFoundError('default_settings.json')

if "--no-settings" in argv:
    print("Using only default settings")
else:
    path: str = os.path.abspath(os.path.dirname(__file__))
    path = os.path.join(path, "../vscode/settings.json")
    try:
        with open(path, 'r', encoding="utf-8") as setfile:
            settings = json.load(setfile)
    except:
        print("inexistent or ill-formated settings.json file, using defaults settings only")

```

```

def get(setloc: str) -> Any:
    path: list[str]=setloc.split('.')
    try:
        temp = settings
        for key in path:
            temp = temp[key]
        return temp
    except (KeyError, TypeError):
        try:
            temp = defaults
            for key in path:
                temp = temp[key]
            return temp
        except:
            print("No setting found for ", setloc, file=stderr)
            return None

```

```

def set(setloc: str, value) -> bool:
    """Définit les paramètres 'setloc' à la valeur value.

```

Args:

setloc (str): Paramètre à mettre à jour, un fichier. Permet de séparer par groupe (par exemple "group.sub.setting")

value (any): La nouvelle valeur du paramètre, doit être représentable en json

Returns:

bool: if the update was successful
 """

```

temp = settings
path: list[str]=setloc.split('.')
for key in path[:-1]:
    if type(temp) is not dict:
        print("setting path to non dict, aborting", file=stderr)
        return False
    if key not in temp:
        temp[key] = { }
    temp = temp[key]

```

```
temp[path[-1]] = value
return True
```

```
def save() -> bool:
    """Sauvegarde les changer dans settings.json
```

Returns:

bool: Si les changement on pu se faire
"""

```
try:
    with open(path, 'w', encoding="utf-8") as setfile:
        json.dump(settings, setfile, indent=2)
    return True
except Exception as e:
    print("Illegal setting for json representation or no write access:", e, file=stderr)
    return False
```

Module simulation.py :

```
import time
import numpy as np
```

```
from way import *
from avion import *
from aeroport import *
from graph import *
#from affichage import *
```

```
class Sim :
    def __init__(self, G, tab_navions) :
        """initiates the simulation
```

Args:

G (Graph (see Graph.py)): the graph in which the simulation takes place, fixed
tab_navions (Plane array): lists all the planes currently running through the simulation
"""

```
if not isinstance(tab_navions, list) :
    raise TypeError("tab_navions must be of type list")
if not isinstance(G, Graph) :
    raise TypeError("G must be of type Graph")
self.tick = 0 #the current tick of the simulation 1 TICK = 1 MINUTE INGAME
self.graph = G #graph of the sim
self.planes = tab_navions #list of the planes in the sim
self.tickspeed = 1 #number of ticks per second
def set_tickspeed(self, ts) :
    """changes the tickspeed of the simulation
```

Args:

ts (int): neww tickspeed
"""

```
self.tickspeed = ts
```

```

def new_plane(self, plane) :
    """adds a new plane in the simulation

    Args:
    plane (Plane): the plane to be added to the sim
    """
    self.planes.append(plane)
    def new_tick(self) :
        """simulates a tick
        """
        self.tick += 1
        for p in self.planes :
            p.ptick()

```

Module way.py :

```

# Code sous liscence GPL3+. Plus de détail a <https://www.gnu.org/licenses/> ou dans le fichier
LICENSE
# encoding = utf8

```

```

from math import *
from numpy import *
from random import *
from time import *
#from settings import *
#from aeroport import *
#from avion import *

```

```

from collections import defaultdict
import sys

```

```

nombre_sommet = 5
class Heap():
    def __init__(self):
        self.array = []
        self.size = 0
        self.pos = []
        def newMinHeapNode(self, v, dist):
            minHeapNode = [v, dist]
            return minHeapNode
        # A utility function to swap two nodes
        # of min heap. Needed for min heapify
        def swapMinHeapNode(self, a, b):
            t = self.array[a]
            self.array[a] = self.array[b]
            self.array[b] = t
        # A standard function to heapify at given idx
        # This function also updates position of nodes
        # when they are swapped.Position is needed
        # for decreaseKey()
        def minHeapify(self, idx):

```

```

smallest = idx
left = 2*idx + 1
right = 2*idx + 2
if (left < self.size and
self.array[left][1]
< self.array[smallest][1]):
smallest = left
if (right < self.size and
self.array[right][1]
< self.array[smallest][1]):
smallest = right
# The nodes to be swapped in min
# heap if idx is not smallest
if smallest != idx:
# Swap positions
self.pos[self.array[smallest][0]] = idx
self.pos[self.array[idx][0]] = smallest
# Swap nodes
self.swapMinHeapNode(smallest, idx)
self.minHeapify(smallest)
# Standard function to extract minimum
# node from heap
def extractMin(self):
# Return NULL wif heap is empty
if self.isEmpty() == True:
return
# Store the root node
root = self.array[0]
# Replace root node with last node
lastNode = self.array[self.size - 1]
self.array[0] = lastNode
# Update position of last node
self.pos[lastNode[0]] = 0
self.pos[root[0]] = self.size - 1
# Reduce heap size and heapify root
self.size -= 1
self.minHeapify(0)
return root
def isEmpty(self):
return True if self.size == 0 else False
def decreaseKey(self, v, dist):
# Get the index of v in heap array
i = self.pos[v]
# Get the node and update its dist value
self.array[i][1] = dist
# Travel up while the complete tree is
# not heapified. This is a O(Logn) loop
while (i > 0 and self.array[i][1] <
self.array[(i - 1) // 2][1]):
# Swap this node with its parent
self.pos[ self.array[i][0] ] = (i-1)//2

```



```

self.pos[ self.array[(i-1)//2][0] ] = i
self.swapMinHeapNode(i, (i - 1)//2 )
# move to parent index
i = (i - 1) // 2;
# A utility function to check if a given
# vertex 'v' is in min heap or not
def isInMinHeap(self, v):
if self.pos[v] < self.size:
return True
return False
def printArr(dist, n):
print ("Vertex\tDistance from source")
for i in range(n):
print ("%d\t\t%d" % (i,dist[i]))
class Graph():
def __init__(self, V):
self.V = V
self.graph = defaultdict(list)
# Adds an edge to an undirected graph
def addEdge(self, src, dest, weight):
# Add an edge from src to dest. A new node
# is added to the adjacency list of src. The
# node is added at the beginning. The first
# element of the node has the destination
# and the second elements has the weight
newNode = [dest, weight]
self.graph[src].insert(0, newNode)
# Since graph is undirected, add an edge
# from dest to src also
newNode = [src, weight]
self.graph[dest].insert(0, newNode)
# The main function that calculates distances
# of shortest paths from src to all vertices.
# It is a O(ELogV) function
def dijkstra(self, src):
V = self.V # Get the number of vertices in graph
dist = [] # dist values used to pick minimum
# weight edge in cut
# minHeap represents set E
minHeap = Heap()
# Initialize min heap with all vertices.
# dist value of all vertices
for v in range(V):
dist.append(1e7)
minHeap.array.append( minHeap.
newMinHeapNode(v, dist[v]))
minHeap.pos.append(v)
# Make dist value of src vertex as 0 so
# that it is extracted first
minHeap.pos[src] = src
dist[src] = 0

```

```

minHeap.decreaseKey(src, dist[src])
# Initially size of min heap is equal to V
minHeap.size = V
# In the following loop,
# min heap contains all nodes
# whose shortest distance is not yet finalized.
while minHeap.isEmpty() == False:
# Extract the vertex
# with minimum distance value
newHeapNode = minHeap.extractMin()
u = newHeapNode[0]
# Traverse through all adjacent vertices of
# u (the extracted vertex) and update their
# distance values
for pCrawl in self.graph[u]:
v = pCrawl[0]
# If shortest distance to v is not finalized
# yet, and distance to v through u is less
# than its previously calculated distance
if (minHeap.isInMinHeap(v) and
dist[u] != 1e7 and \
pCrawl[1] + dist[u] < dist[v]):
dist[v] = pCrawl[1] + dist[u]
# update distance value
# in min heap also
minHeap.decreaseKey(v, dist[v])
# printArr(dist,V)

class Graph_list:
def __init__(self, size):
self.adj_matrix = [[0] * size for _ in range(size)]
self.size = size
self.vertex_data = [''] * size

def add_edge(self, u, v, weight):
if 0 <= u < self.size and 0 <= v < self.size:
self.adj_matrix[u][v] = weight
self.adj_matrix[v][u] = weight # For undirected graph

def add_vertex_data(self, vertex, data):
if 0 <= vertex < self.size:
self.vertex_data[vertex] = data

def dijkstra(self, start_vertex_data):
start_vertex = self.vertex_data.index(start_vertex_data)
distances = [float('inf')] * self.size
distances[start_vertex] = 0
visited = [False] * self.size

for _ in range(self.size):
min_distance = float('inf')

```

```

u = None
for i in range(self.size):
    if not visited[i] and distances[i] < min_distance:
        min_distance = distances[i]
        u = i

```

```

if u is None:
    break

```

```

visited[u] = True

```

```

for v in range(self.size):
    if self.adj_matrix[u][v] != 0 and not visited[v]:
        alt = distances[u] + self.adj_matrix[u][v]
        if alt < distances[v]:
            distances[v] = alt

```

```

return distances

```

```

def taille (graph):
    t = 0
    for i in graph:
        if i[0] > t:
            t = i[0]
        if i[1] > t:
            t = i[1]
    return t+1
totalstart = time()
for i in range (5, 11):
    def BellmanFord(graph, source):
        """Algorithme de Bellman-Ford

```

Args:

graph (Array): tableau de triplet, chacun représentant une arête de la forme (source, destination, taille)

source (int): Origine du chemin rechercher.

Returns:

Array: tableau des distances

```

"""

```

```

distance = [float("Inf")] * (nombre_sommet+1)
distance[source] = 0

```

```

for _ in range(nombre_sommet - 1):
    for u, v, w in graph:
        if distance[u] != float("Inf") and distance[u] + w < distance[v]:
            distance[v] = distance[u] + w

```

```

for u, v, w in graph:
    if distance[u] != float("Inf") and distance[u] + w < distance[v]:
        print("Graphe avec un cycle négatif")

```

```

return
del(distance[0])
return distance

def est_connexe (graph):
t = BellmanFord(graph, 1)
for i in t:
if i == float("Inf"):
return False
return True

print("Temps pour ", nombre_sommet, "sommets")
def random_graph (p):
tab_b = []
g = Graph(nombre_sommet+1)
gl = Graph_list(nombre_sommet+1)
letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
for i in range (nombre_sommet):
gl.add_vertex_data(i, letters[i])
for i in range (1,nombre_sommet+1):
for j in range (1, nombre_sommet+1):
if (i != j):
a = random()
if a >= p:
c = randint(60, 60*8)
tab_b.append((i, j, c))
tab_b.append((j, i, c))
g.addEdge(i, j, c)
g.addEdge(j, i, c)
gl.add_edge(i, j, c)
if (not est_connexe(tab_b)):
return random_graph (p)
return (tab_b, g, gl)

def test_des_algo ():
p = randint(300, 900)/1000
graph = random_graph (p)
return graph

start = time()
n = 100000
graph_alea_b = []
graph_alea_d = []
graph_alea_dl = []
for i in range (n):
a, b, c = test_des_algo()
graph_alea_b.append(a)
graph_alea_d.append(b)
graph_alea_dl.append(c)
print("Temps de génération aléatoire ", time() - start)

```

```

def test_simulation_speed():
s1 = time()
print("starting update")
for i in range (n):
BellmanFord (graph_alea_b[i], 1)
print("graph time Bellman-Ford :", time()-s1)
s2 = time()
for i in range (n):
graph_alea_d[i].dijkstra (1)
print("graph time Dijkstra adj_list :", time()-s2)
s3 = time()
for i in range (n):
graph_alea_dl[i].dijkstra('B')
print("graph time Dijkstra fib_heap :", time()-s3)
test_simulation_speed()
nombre_sommet += 1
print("total:", time()-totalstart)

```

Module launch.py :

```

#!/usr/bin/env python3
# Code sous liscence GPL3+. Plus de détail a <https://www.gnu.org/licenses/> ou dans le fichier
LICENCE_FR ou LICENSE
# encoding=utf8

```

Fichier utilisé pour lancer le code depuis une compilation du code source

```

import Python.main
Python.main.launch_app()

```

Module LICENSE :

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<https://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other

than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts,

regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the

Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the

licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version,

but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily

for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<https://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<https://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read

<<https://www.gnu.org/licenses/why-not-lgpl.html>>.