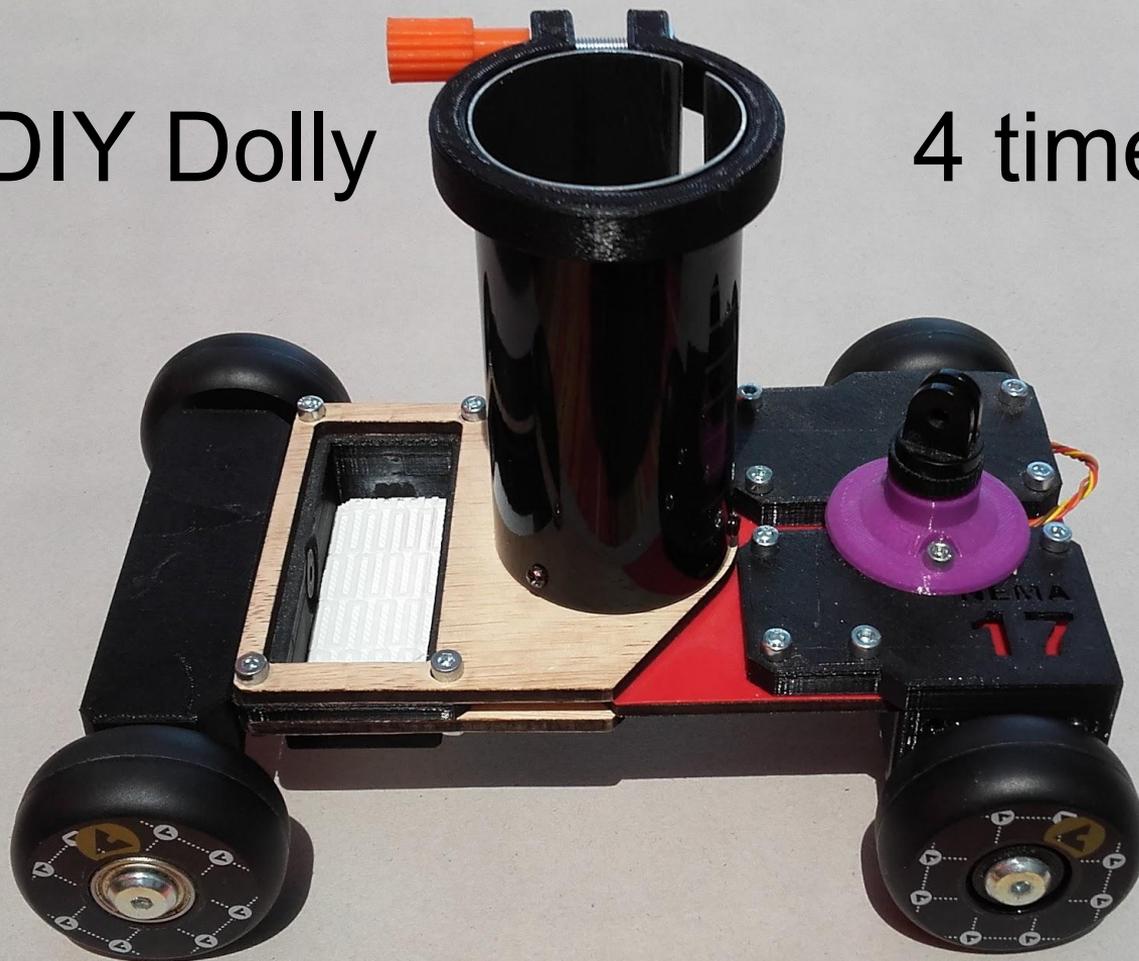


DIY Dolly

4 timelapse\*



\* (en bon Français: Traineau fait par soi même pour réaliser des "Laps de temps")

Une Dolly de cinéma pro permet de faire des travelings, qui donnent du mouvement à une séquence.



J'ai déjà fait quelques timelapses fixes, une mini-Dolly leur donnerait du mouvement.

# Apparté: les étapes pour créer un timelapse

Une fois l'ensemble de prise de vues réalisé (x photos).....

.....( c'est là qu'entre en jeu Dolly et APN ) .....

..... il faut transformer le tout sous une forme "vidéo".

## Pour cela, il faut:

- transférer les photos sur un ordinateur
- supprimer les photos floues - sur ou sous exposées ...etc
- tout redimensionner au format vidéo final
- faire l'assemblage final

# Cahier des charges

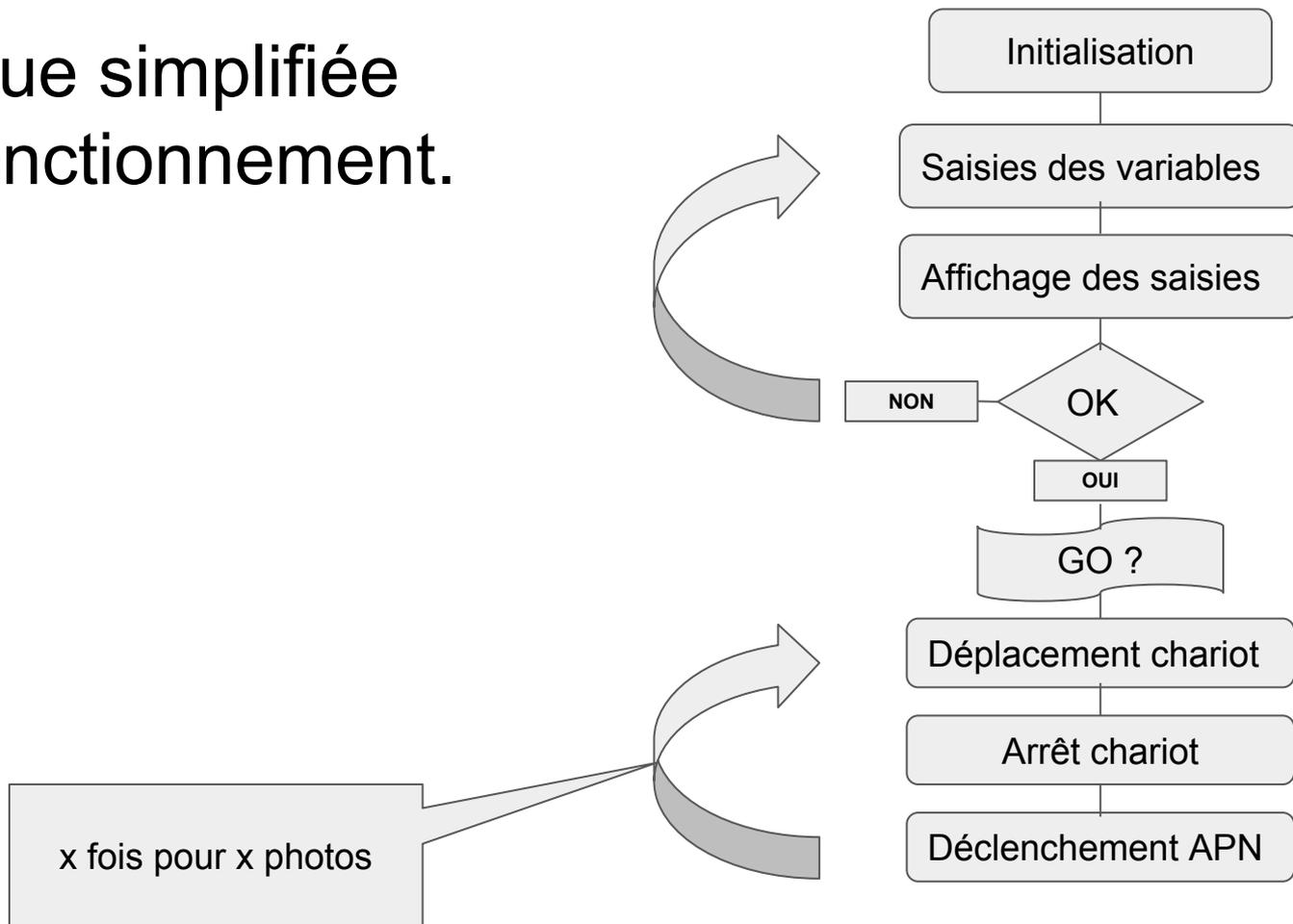
- capable de **faire des travelings** sans rails, donc doit pouvoir rouler sur un sol +/- plat (pas de rail) malgré qq irrégularités du sol ( env. 1 cm)
- capable de **commander un système de prise de vue**, donc doit supporter 2 kg de matériel (APN - batteries - moteur - rotule photo)
- **stable** avec APN à 40 cm du sol
- **programmable**: en distance, durée et nombre de photos
- doit être **autonome**
- **transportable** dans malette alu standard
- j'éviterai de m'y asseoir dessus

# Les éléments nécessaires au chariot

- 1 châssis avec 4 roues
- 1 pack batterie
- 1 moteur d'entraînement et son driver
- 1 micro-contrôleur
- 1 écran pour visu et saisie des paramètres
- 1 APN et sa colonne-support



# Logique simplifiée de fonctionnement.



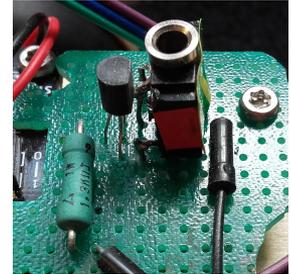
# Supports de la logique

- 1 Arduino Uno
- 1 écran LCD et ses boutons de saisie



# Exploitation

- 1 Moteur pas à pas et 1 driver
- 1 switch déclenchement APN



# Alimentation

Pack “batteries” 12 v (5 v pour Arduino)

Nb: tous les composants sont embarqués.

La conception initiale ne prévoit pas de télécommande.



# C'est à ce moment là que je découvre l'OFS !!

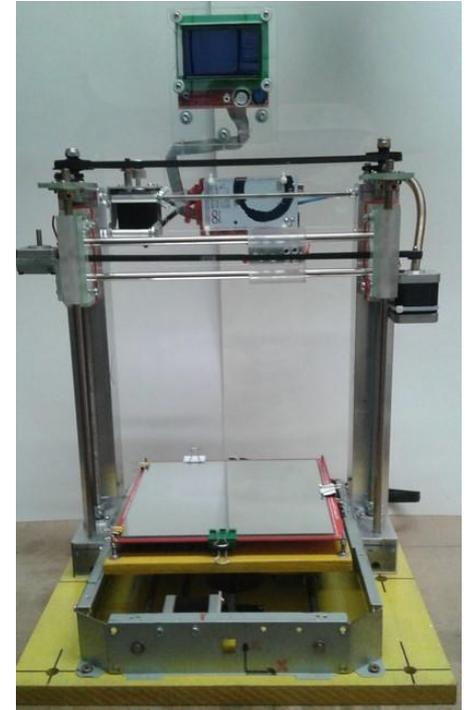


Courant 2016, je dispose enfin de beaucoup de temps libre. Je connais l'existence des FabLab et fais une recherche pour voir ce qui existe sur St Etienne: OpenFactory !!!!!

**L'OFS (Open Factory Sainté) c'est quoi:**

- des gens accueillants, pleins de compétences,
- du matériel à disposition
- des réponses à pratiquement toutes les questions
- des débuts de solutions à beaucoup de problèmes.

**Du coup, j'en profite pour finir mon imprimante 3D commencée depuis longtemps.**



# Grâce à l'OFS, mon projet Dolly devient réalité.

J'avais globalement bien défini les divers besoins et éléments à associer pour la réalisation de ce projet. Par contre je n'avais pas mesuré du tout l'ensemble des embûches et des connaissances nécessaires à l'aboutissement du projet.

Heureusement, j'ai eu de l'aide: David, puis Hubert, mes nouveaux copains Roger et Bruno, pour m'aider dans cette tâche ardue !

Ce projet (complexe pour moi) va finalement aboutir grâce à eux.

En voici la suite .....



# La partie programmation Arduino 1/2



La logique de fonctionnement étant définie, il faut écrire le programme Arduino.

N'ayant aucun talent dans ce domaine, une recherche sur le net avec les mots clé: "Timelapse arduino dolly" renvoie beaucoup d'infos et après pas mal de recherche je trouve une solution qui me semble toute prête: "Slider cam par Rob Taylor".

Ce programme gère tout : l'affichage et saisie via le LCD - le pilotage moteur - la commande APN.

Il y aura les paramètres propres au projet à ajuster dans ce programme à savoir:

les seuils de sensibilité des boutons du LCD, le ratio "1 pas / distance parcourue" et ajuster des tempo "delay" (durée de l'impulsion pour déclencher l'APN (variable selon les modèles)).

# La partie programmation Arduino 2/2 (refonte totale)



Le programme “tout prêt” est fonctionnel, mais il s’avère que la manipulation du mini clavier de 6 boutons installé à bord du chariot est inconmode.

**Mais à l’OFS on discute, on montre ce que l’on fait et on demande conseil.**

C’est ainsi que apprenons l’existence de **Appinventor** et que Bruno entrevoit toutes les possibilités qui s’offrent à nous:

- pilotage possible avec un Smartphone (nécessite l’ajout d’un module Bluetooth sur le chariot, mais permet de se débarrasser de l’écran LCD et de la complexité de la gestion de son affichage dans l’Arduino (  $\frac{2}{3}$  du programme))
- Nous écrivons un nouveau programme (Bruno toujours) qui permettra de rajouter une fonction “pause” en cas de problème durant la prise de vue.

Le programme complet: >>>> voir fichier joint: pilot\_dolly\_smartphone\_evolution110\_final.ino

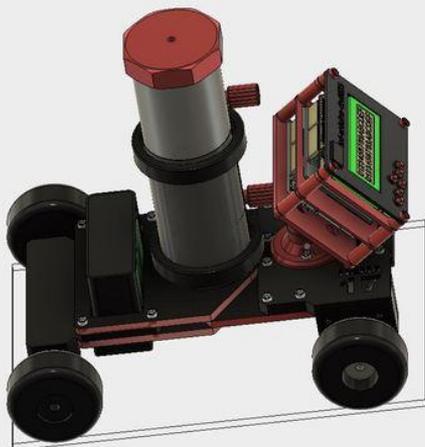
# CAO du chariot (sur Fusion 360)



AUTODESK®  
FUSION 360

- dessin de tous les composants en ma possession:  
roues - moteur - batteries - Arduino - driver - LCD ..... etc
- ensuite concevoir les éléments manquants en tenant compte des moyens de fabrication à disposition : bois - métal - impression 3D

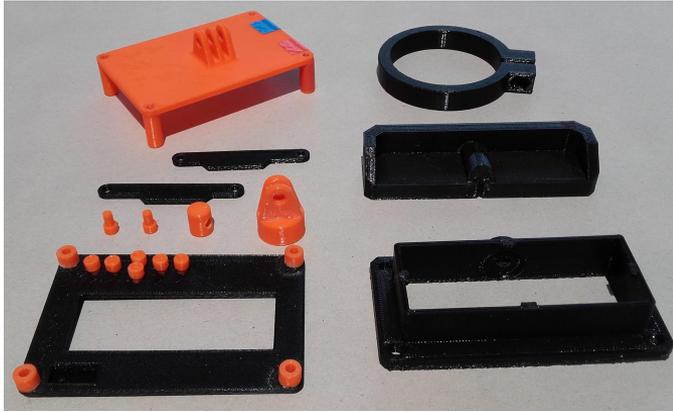
Version  
avec  
LCD



Version  
avec  
Smart  
phone



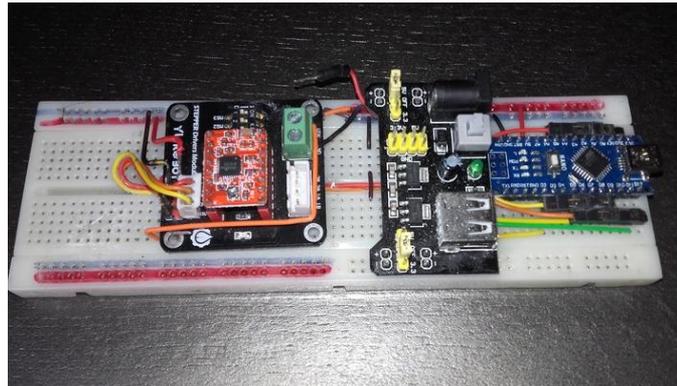
# Impression 3D



# Découpe Laser



# Electronique



# Evolution avec Smartphone et Appinventor

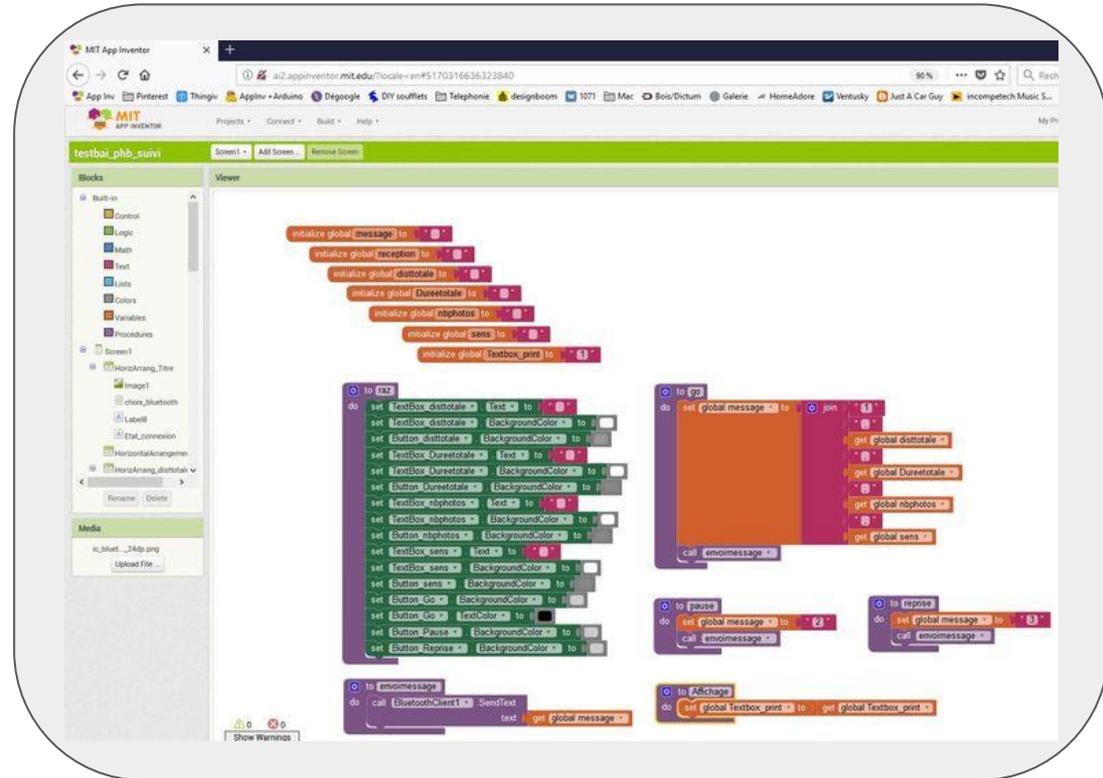


- Exit le LCD et ses boutons de saisies très peu pratiques.
- Le logiciel en ligne **Appinventor** permet de concevoir une appli Android via un Smartphone et un module Bluetooth installé sur le chariot.
  - La programmation comprend 2 volets :
    - 1 volet de conception de l'écran
    - 1 volet de gestion de l'affichage et des communications Bluetooth avec un module appairé distant.

# Les 2 volets de Appinventor:

L'écran du Smartphone

La programmation



initialize global message to " "

initialize global reception to " "

initialize global disttotale to " "

initialize global Dureetotale to " "

initialize global nbphotos to " "

initialize global sens to " "

initialize global Textbox\_print to " 1 "

```
to raz
do
  set TextBox_disttotale . Text to " "
  set TextBox_disttotale . BackgroundColor to 
  set Button_disttotale . BackgroundColor to 
  set TextBox_Dureetotale . Text to " "
  set TextBox_Dureetotale . BackgroundColor to 
  set Button_Dureetotale . BackgroundColor to 
  set TextBox_nbphotos . Text to " "
  set TextBox_nbphotos . BackgroundColor to 
  set Button_nbphotos . BackgroundColor to 
  set TextBox_sens . Text to " "
  set TextBox_sens . BackgroundColor to 
  set Button_sens . BackgroundColor to 
  set Button_Go . BackgroundColor to 
  set Button_Go . TextColor to 
  set Button_Pause . BackgroundColor to 
  set Button_Reprise . BackgroundColor to
```

0 0  
Show Warnings

```
to go
do
  set global message to join " 1 "
  get global disttotale
  get global Dureetotale
  get global nbphotos
  get global sens
  call envoimessage
```

```
to envoimessage
do
  call BluetoothClient1 .SendText
  text get global message
```

```
to pause
do
  set global message to " 2 "
  call envoimessage
```

```
to Affichage
do
  set global Textbox_print to get global Textbox_print
```

```
to reprise
do
  set global message to " 3 "
  call envoimessage
```

```
when Clock1 .Timer
do
  if BluetoothClient1 .IsConnected
  then
    set TextBox_print .Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
    set global Textbox_print to get global Textbox_print
    call Affichage
```

```
when Screen1 .Initialize
do
  set Clock1 .TimerEnabled to false
  call raz
```

```
when choix_bluetooth .BeforePicking
do
  set choix_bluetooth .Elements to BluetoothClient1 .AddressesAndNames
```

```
when choix_bluetooth .AfterPicking
do
  set choix_bluetooth . Selection to call BluetoothClient1 .Connect
  address choix_bluetooth . Selection
  if BluetoothClient1 . IsConnected
  then
    set Etat_connexion . Text to " Connection BT OK "
    set Etat_connexion . TextColor to red
    set Clock1 . TimerEnabled to true
    set Clock1 . TimerInterval to 800
    call raz
  else
    set Etat_connexion . Text to " Bluetooth Déconnecté "
    set Etat_connexion . TextColor to red
    call raz
```

```
when Button_disttotale .Click
do
  set Button_disttotale . BackgroundColor to green
  set global disttotale to TextBox_disttotale . Text
  set TextBox_disttotale . BackgroundColor to green
  call TextBox_Dureetotale .RequestFocus
```

```
when Button_Dureetotale .Click
do
  set Button_Dureetotale . BackgroundColor to green
  set global Dureetotale to TextBox_Dureetotale . Text
  set TextBox_Dureetotale . BackgroundColor to green
  call TextBox_nbphotos .RequestFocus
```

```
when Button_nbphotos .Click
do
  set Button_nbphotos . BackgroundColor to 
  set global nbphotos to TextBox_nbphotos . Text
  set TextBox_nbphotos . BackgroundColor to 
  call TextBox_sens .RequestFocus
```

```
when Button_Go .Click
do
  set Button_Go . BackgroundColor to 
  set Button_Go . BackgroundColor to 
  set Button_Go . TextColor to 
  call go
```

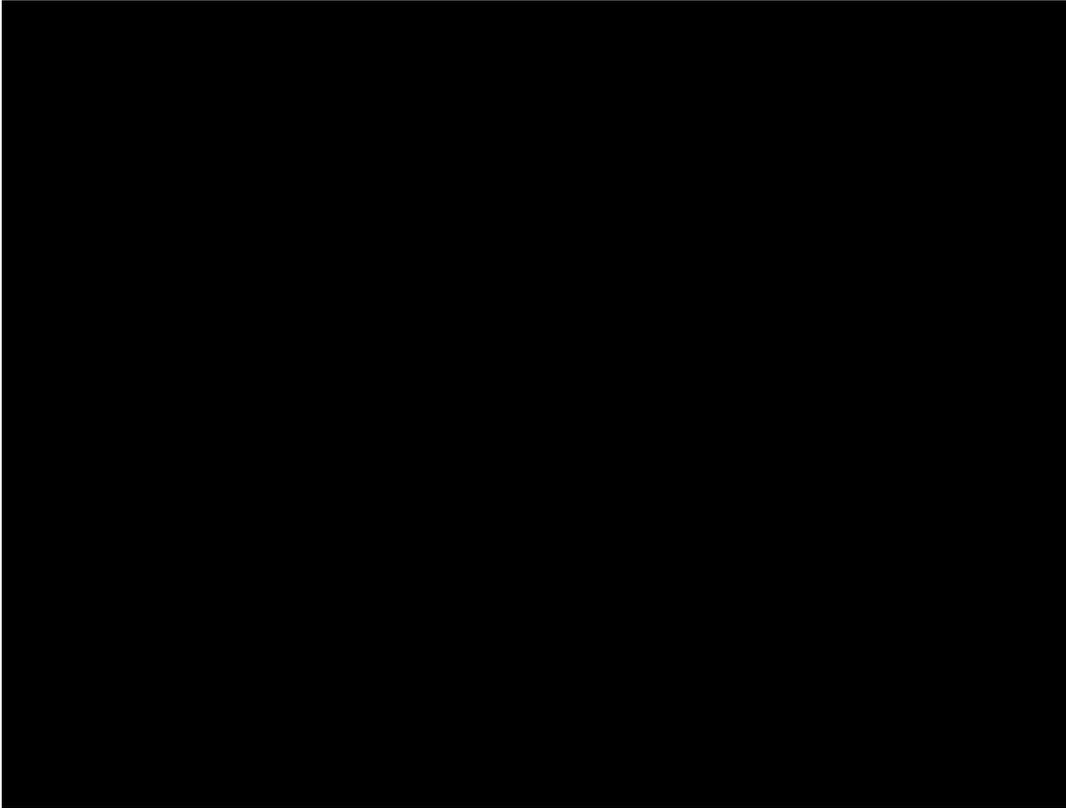
```
when Button_Reprise .Click
do
  set Button_Reprise . BackgroundColor to 
  set Button_Pause . BackgroundColor to 
  call reprise
```

```
when Button_sens .Click
do
  set Button_sens . BackgroundColor to 
  set global sens to
    if TextBox_sens . Text = "g"
    then TextBox_sens . Text = "1"
    else TextBox_sens . Text = "0"
  set TextBox_sens . BackgroundColor to 
  call TextBox_Dureetotale .HideKeyboard
```

```
when Button_Pause .Click
do
  set Button_Pause . BackgroundColor to 
  set Button_Reprise . BackgroundColor to 
  call pause
```

```
when Button_RAZ .Click
do
  call raz
```

# Mon premier timelapse avec traveling



.....il y a de la  
marge de  
progression.....

# Pour + de détails:

**AppInventor:** <http://ai2.appinventor.mit.edu/>

La création d'une application, se fait entièrement en ligne.

**Arduino:** <https://www.arduino.cc/>

**Instructables:** <http://www.instructables.com/id/Course-on-MIT-App-Inventor-and-Arduino/>

**Open Factory:** <http://www.openfactory42.org/>

**Github:** en préparation