**Document Synthèse**

**Par Jonathan Lavigne  
Jérémie Gladu et  
Marc Simard**

**Remis à  
Robert Turenne**

**Département d’informatique  
Cégep de Saint-Jérôme  
Mardi 12 mai 2015**

Table des matières

Table des matières

[Journal de bord 3](#_Toc419199797)

[Le gyroscope 8](#_Toc419199798)

[Définition 8](#_Toc419199799)

[Origine 8](#_Toc419199800)

[Différences entre le gyroscope et le gyromètre 8](#_Toc419199801)

[Expérimentation du gyroscope 9](#_Toc419199802)

[Fonctionnement d’un gyroscope électronique 9](#_Toc419199803)

[Applications 10](#_Toc419199804)

[Référence : 11](#_Toc419199805)

[Capteur de flexion 12](#_Toc419199806)

[Description et principes: 12](#_Toc419199807)

[Utilité dans notre projet et usages possibles : 12](#_Toc419199808)

[Comment l’utiliser adéquatement: 12](#_Toc419199809)

[Référence: 13](#_Toc419199810)

[Accéléromètre 14](#_Toc419199811)

[Définition et principes: 14](#_Toc419199812)

[Avantage: 14](#_Toc419199813)

[Application sportive: 14](#_Toc419199814)

[Info circulation: 14](#_Toc419199815)

[Rotation du téléphone: 14](#_Toc419199816)

[Inconvénients: 15](#_Toc419199817)

[Référence: 15](#_Toc419199818)

[Physique de l’application 16](#_Toc419199819)

[Formules utilisés: 17](#_Toc419199820)

[Référence: 17](#_Toc419199821)

[Fonctionnement du gant 18](#_Toc419199822)

[Le quaternion 18](#_Toc419199823)

[Calcul des angles 19](#_Toc419199824)

[Calcul de la position 19](#_Toc419199825)

[vitesse constante : 19](#_Toc419199826)

[Avec la variation d’angle : 20](#_Toc419199827)

[Capteur de contact (input pull-up) 21](#_Toc419199828)

[Propriétés des pins configurés en «INPUT\_PULLUP» 21](#_Toc419199829)

[Fonctionnement du circuit 21](#_Toc419199830)

[Référence 21](#_Toc419199831)

# Journal de bord

La présente section donne une approximation du temps passé sur le projet en général. Il est possible que certaines dates et que du temps de travail aient étés oubliés.

**Jonathan Lavigne :**

1. 22 au 28 mars
   1. Test\_Contact\_Et\_Flex (Processing) – 1h
   2. Cont\_Flex\_Test (Arduino) – 30mins
   3. Accel\_Test (Arduino) – 1h
   4. Accel\_Test (Processing) – 3h
   5. Total : 9h30
2. 29 mars au 4 avril
   1. Test\_Serial\_V0.75 (Accel, Processing + Arduino) – 3h
   2. Test\_Mouse\_Robot (Processing) – 2h
   3. Application d’arc (Processing) – 1h
   4. Total : 6h
3. 5 au 12 avril
   1. Gyroscope rectangle (Arduino + processing) – 5h
   2. Gyroscope avec souris – 1h
   3. Journal de bord + Table des matières livrables – 1h
   4. Total : 7h
4. 14 avril
   1. Gestion de tous les capteurs via arduino - 2h15
   2. Processing pour tous ces capteurs - 45m
   3. Total : 3h
5. 17 avril
   1. Finalisation du code processing
   2. intégration de la configuration
   3. Total : 2h
6. 20-21 avril (17h30 - 19h, 21h15 - 00h00)
   1. - Construction du gant - 4h15
7. 21 avril :
   1. finalisation du gant
   2. finalisation du fonctionnement du arduino
   3. finalisation du processing
   4. test avec application de l’arc
   5. Total : 3h
8. 24 avril :
   1. Intégration des applications Arduino
      1. Processing de l’arc
      2. Processing de la souris
      3. Menu principale pour son fonctionnement
   2. correctifs de bugs mineurs
   3. Total : 2h
9. 28 avril :
   1. Correction de bugs mineurs du programme processing
   2. intégration du drag’n drop avec la souris. - 2h
   3. Intégration de la documentation dans le document final - 1h
   4. Total : 3h
10. 1er mai et 4eme Mai :
    1. Amélioration de la documentation du gyroscope : 30 mins.
    2. Amélioration de la documentation du gyroscope : 45 mins.
    3. Total : 1h15
11. 5 mai :
    1. Amélioration des codes +
    2. finalisation de la documentation du gyroscope
    3. Section “les abandons”
    4. Total : 3h
12. 8 mai :
    1. Documentation du fonctionnement du gant : 2h15.
13. 12 Mai :
    1. Finalisation du document de synthèse
    2. document personnel
    3. test application final
    4. Finalisation du dossier de remise + affiche
    5. Total : 5h
14. Auparavant : Heures de cours normales + 10h dans la semaine de relâche.

**Jérémie Gladu :**

1. 22 au 28 février
   1. Petite recherche d’introduction à Processing (1h30)
   2. Début de la documentation du gyroscope (3h)
2. 1er au 7 mars
   1. Début de la documentation du capteur de flexion (3h)
   2. Réalisation d’un croquis de menu en processing (1h30)
3. 8 au 14 mars
   1. Croquis du manche de l’arc (1h30)
4. 15 au 21 mars
   1. Amélioration de l’arc (en tout: 4h30)
      1. Ajout de la corde
      2. Ajout d’une flèche
      3. Tentative d’y insérer de la physique
      4. Optimisation de ses fonctions
   2. Recherche sur la gestion des données du gyroscope pour aider Jonathan (1h)
5. 22 au 28 mars
   1. Documentation sur la physique derrière le tir à l’arc (2h).
   2. Documentation sur le capteur de contact (1h).
   3. Optimisation de l’arc avec l’aide de Jonathan pour ajouter la physique (30min)
6. 29 mars au 4 avril
   1. Programmation d’un menu en C# (4h30)
7. 5 au 12 avril
   1. Documentation du gyroscope (2h)
   2. Amélioration du menu en C# (1h30)
   3. Compréhension du code à Jonathan (30min)
8. 13 au 19 avril
   1. Définir mes tâches réalisées depuis le début (1h)
   2. Programmer le menu des paramètres (4h30)
9. 20 au 26 avril
   1. Programmer le menu des paramètres (4h)
   2. Optimisation/finalisation du menu (2h)
10. 26 avril au 3 mai
    1. Optimisation/finalisation du menu (2h) \*41h30 à ce jour.
    2. Lecture et compréhension des codes de mes coéquipiers (30min)
    3. Finalisation de la documentation sur le capteur de flexion (30min)
11. 4 au 10 mai
    1. Documentation et compréhension de code et concept arduino/processing (2h30)

**Marc Simard :**

1. 22 au 28 février
   1. Documentation accéléromètre - 3h
   2. Recherche arduino - 1h
2. 1 au 7 mars
   1. Recherche programmation processing en générale - 3h
   2. Document horaire du projet - 3h
3. 8 au 14 mars
   1. Application Scroll bar - 2h
4. 15 au 21 mars
   1. Recherche processing
      1. Couleurs - 1h
      2. Formes - 1h
      3. Souris - 30 min
   2. Début application pour le gant - 1h
5. 22 au 28 mars
   1. Application test pour le gant - 5h
6. 29 mars au 4 avril
   1. Tentative processing 3D - 2h
7. 12 au 18 avril
   1. Lecture fichier + arduino - 2h
   2. Application arc 1.0 - 7h
8. 19 au 25 avril
   1. Application arc 1.1 - 3h
   2. Application arc 1.2 - 5h
   3. Application arc 1.3 - 3h
9. 26 avril au 3 mai
   1. Application arc 1.4 - 3h
   2. Début application arc 1.5 - 3h
   3. Lecture des documents et codes - 1h
10. 4 au 10 mai
    1. Application arc 1.5 - 2h
    2. Définir mes tâches réalisées depuis le début - 1h
    3. Documentation physique de l’arc et accéléromètre - 2h

Les abandons

Cette sous-partie expliquera les éléments initiaux prévus qu’on a finalement décidé d’abandonner et la raison de l’abandon. Cette partie aura aussi pour but de donner le temps perdu et le temps gagné, où et comment ce temps a été utilisé.

Après avoir passé environ 20h à essayer de comprendre et de contrôler les données de l’accéléromètre, nous avons décidé d’abandonner ça. La raison en est fort simple : les données sont tellement instables et influencées par le monde extérieur qu’il s’avérait pratiquement impossible de contrôler un élément avec ces données. En effet, même lors de l’inertie, l’accélération que l’accéléromètre calculait, variait de 3 à 4 mètres par secondes. Le temps initial que l’on avait prévu sur la compréhension et l’analyse des données de l’accéléromètre pour les transformer en position était de 15 heures. L’abandon a donc causé un déficit de 5 heures sur l’horaire final.

Ensuite, initialement, on croyait qu’il était impossible de contrôler la souris avec « processing » en utilisant des données autres que celles venant de la souris. Alors on avait prévu construire plusieurs classes pour faire l’interface utilisateur. Dans ces classes on remarque notamment la classe bouton, curseur, « slider », et plusieurs autres. La classe bouton et la classe « slider » ont étés implémenté, les deux ensembles prenant 3h de développement. Au début, on avait prévu 1h pour chaque, causant un déficit d’une heure. Heureusement, le temps initial prévu pour toute l’interface utilisateur était d’environ 5h. On a donc gagné 2h de temps supplémentaire en abandonnant cette partie. La raison de l’abandon est simple : nous avons trouvé la classe «Robot» de java, qui permet de contrôler le clavier, la souris, la molette de la souris et les clics. Le 2h gagné dans l’abandon de cette partie a permis à la compréhension et au développement de la classe «Robot».

Par après, puisque l’oral n’est plus en classe devant tous, la version 1.0 du présent document n’aura pas de PowerPoint, nous donnant un avance d’environ 10h. Ce temps a été réutilisé pour peaufiner le menu principal et le rendre accessible et simple d’utilisation avec le gant.

Par la suite, puisque nous avions déjà perdu quelques heures à cause des imprévus concernant l’accéléromètre, nous avons dû abandonner la version 3D de la simulation du tir à l’arc. Nous avons cependant gardé l’application et ajouter quelques éléments intéressant pour la présentation du 22 mai. Malgré que le temps alloué ne soit pas présent dans la version initiale du Microsoft Project, nous avions pensé passer 6h sur le développement de l’application « processing ». Le temps du développement du menu a été plus long que prévu, il a donc empiété sur le temps initial du développement de l’arc. De plus, l’application finale de la simulation du tir à l’arc a prit environ 4h à développer. Nous donnant un déficit d’environ 3h à cause du développement du menu.

Finalement, nous avons décidé d’abandonner la mise à jour du Microsoft Project pour plutôt tenir un journal de bord contenant toutes les heures et les travaux des différents membres. Ce journal de bord est présent ci-haut dans le présent document. Étant donné qu’il y avait environ 30 minutes par cours réservé à la mise à jour du document Microsoft Project, nous avons donc gagné plusieurs heures suite au 22 mars. Ces heures ont été réutilisées dans la documentation et l’écriture du présent document en plus d’être utilisé pour la compréhension de chacune des applications des membres.

Temps initial prévu des abandons : 15 + 5 + 10 + 4 + 0.5\*16 = environ 42 heures

Temps passé sur les abandons : 20 + 3 + 0 + 7 + 0.5 \* 8 = environ 34 heures

Temps réutilisable réutilisé : 8 heures (Documentation + menu)

Documentation du projet

La présente partie présentera la documentation séparée des différents concepts scientifiques présents dans le projet final.

# Le gyroscope

## Définition

Un gyroscope est un instrument qui permet de mesurer un angle dans l’espace (Ou selon un ou deux axes) selon un référentiel inerte. Il exploite le principe de conservation du moment angulaire. En effet, le terme gyroscope provient du grec : le préfix “gyro” qui signifie “rotation” et du suffixe “scope” qui signifie “qui observe”.

## Origine

Il fut inventé par le physicien français [Léon Foucault](http://www.futura-sciences.com/magazines/matiere/infos/personnalites/d/physique-leon-foucault-1118/) en 1852.

À la suite d'une expérience basée sur les mouvements de la [Terre](http://www.futura-sciences.com/magazines/terre/infos/dico/d/structure-terre-terre-4725/), Foucault les met en évidence grâce au pendule (appelé pendule de Foucault, et toujours visible au Panthéon de Paris). Cependant, il se rend compte que le pendule effectue ses rotations à une vitesse inférieure à celle de la Terre d’un facteur d’environ 1/sin (latitude) où «latitude» est la latitude actuelle du gyroscope.

## Différences entre le gyroscope et le gyromètre

Il ne faut pas confondre gyroscope et gyromètre : un gyroscope est un capteur de **position** angulaire alors qu’un [gyromètre](http://fr.wikipedia.org/wiki/Gyrom%C3%A8tre) un capteur de **vitesse** angulaire. Chacun possédant leurs utilités et leurs fonctions bien précises.

Par exemple, le gyroscope peut servir à la stabilisation d’une direction ou d’un référentiel mécanique comme dans les missiles ou les drones. Il peut aussi servir avec l’accéléromètre pour déterminer la position et la vitesse d’un véhicule quelconque dans l’espace.

Pour sa part, le gyromètre peut servir pour pratiquement les mêmes choses que le gyroscope, sauf que ses données sont nettement plus précises. Contrairement au gyroscope, le gyromètre ne possède pas de pièce mobile et n’a pas d’inertie. Il est fortement utilisé dans la stabilisation des plateformes, des fusées ou des avions.

Dans le cas du projet de notre gant, le gyroscope sera utilisé pour sa simplicité et sa stabilité. De plus, c’est la puce électronique la moins coûteuse des deux.

## Expérimentation du gyroscope

Une façon simple d'expérimenter cet effet consiste à tenir à bout de bras une roue de vélo par les écrous du moyeu et de la faire faire tourner rapidement par une autre personne. Lorsque l'on tente de pencher sur le côté la roue en rotation, on ressent une résistance. C'est la conservation du moment de rotation qui tend à s'opposer à ce dernier mouvement.

La vidéo suivante (en anglais) explique très bien le fonctionnement d’un gyroscope.

<https://www.youtube.com/watch?v=zbdrqpXb-fY>

## Fonctionnement d’un gyroscope électronique

Un gyroscope électronique tel que le MPU6050 possède un fonctionnement assez différent des gyroscopes mécaniques conventionnels.

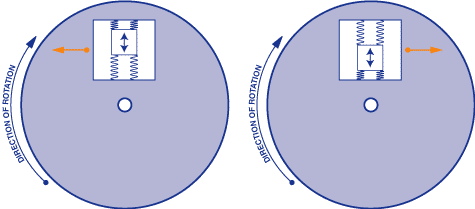
Tout d’abord, il est nécessaire de mentionner que dans un gyroscope électronique, il n’y a pas de pièce mobile visible à l’œil humain tel que dans le gyroscope de Foucault. Ce genre de gyroscope, dans notre cas, est appelé un [MEMS](http://en.wikipedia.org/wiki/Microelectromechanical_systems) (microelectromechanical system), système microélectronique en français. Brièvement, un MEMS est un système électronique nanoscopique.

Pour continuer, le gyroscope électronique possède lui aussi un système d’axe bien spécifique en lien direct avec les composantes électroniques internes. L’image suivante montre le système d’axe d’un gyroscope électronique à 3 axes :

  
Source : <https://learn.sparkfun.com/tutorials/gyroscope/how-a-gyro-works>

Une rotation dans le sens horaire autour de l’axe des Z, antihoraire autour des Y et antihoraire autour des X, par exemple seraient alors interprétées comme négatif. Note : les axes du MPU6050 sont exactement les mêmes et les sens de rotation sont identiques. La seule différence avec notre gyroscope, c’est qu’il possède un accéléromètre, donc c’est un capteur à 6 axes. Ensuite vient la question essentielle concernant le gyroscope : mais comment est-ce qu’un objet de cette taille peut utiliser le moment de force angulaire pour envoyer des signaux électriques ?

C’est relativement simple. Voici une image simplifié de l’intérieur d’un gyroscope autour d’un axe simple (dans le cas présent, supposons l’axe des Z) :



Dans le cas de la présence d’une rotation horaire, la force de Coriolis qui est appliqué perpendiculairement à la direction du mouvement, ici horizontale, fait en sorte qu’un élément mouvant de la masse à l’intérieur du gyroscope vibre et génère donc une quantité très minuscule de courant qui pourra par la suite être lue et interprété par un microcontrôleur, un arduino dans le cas du gant. Ce courant n’est notamment pas présent lors de l’inertie, ce qui explique la précision du calcul de l’angle actuel. Il ne s’agit que d’augmenter (ou diminuer, dépendamment le sens de la vibration de la masse) l’angle lors de la présence du courant, et de la garder inerte lors de l’inertie.

## Applications

* Pilotage automatique et stabilisation en radio modélisme
* Jouets, sports (Segway...)
* Reconnaissance de gestes
* Télécommandes 3D
* Capteur de mouvement à porter sur soi
* Stabilisation de caméra vidéo, nacelle photo (guimbal)
* Commandes par mouvements
* Détecteur de chocs
* Gant pour contrôler la souris

## Référence :

<http://fr.wikipedia.org/wiki/Gyroscope>

<http://www.futura-sciences.com/magazines/high-tech/infos/dico/d/technologie-gyroscope-11121/>

<http://fr.wikipedia.org/wiki/Gyrom%C3%A8tre>

<http://tiptopboards.free.fr/arduino_forum/viewtopic.php?f=2&t=28>

<http://fr.wikipedia.org/wiki/Gyrom%C3%A8tre_%C3%A0_fibre_optique>

<http://www5.epsondevice.com/en/sensing_device/gyroportal/about.html>

<http://en.wikipedia.org/wiki/Gyroscope>

<https://learn.sparkfun.com/tutorials/gyroscope/how-a-gyro-works>

# Capteur de flexion

## Capteur Papier Flexion Principe Papier LogikDescription et principes:

C’est un appareil analysant le degré de courbature qu’on lui attribue. Plus on le courbe, plus la valeur enregistrée sera grande.

En fait, quand on le courbe, la force de compression se retrouve concentrée sur le point de courbure (F), ce qui modifie la résistance. C’est cette variation que vérifie le capteur de flexion.

En général, il est constitué d’un polymère conducteur enveloppé de légères couches de métal. Lorsque l’appareil est plié, le polymère se déforme et grâce à la conductibilité du métal, la résistance aux bornes est modifiée.

## Utilité dans notre projet et usages possibles :

Comme dans notre projet, c’est un capteur souvent fixé sur les parties mobiles d’un appareil. Dans notre cas, il est installé sur le gant pour réagir avec le degré de courbature de l’un de nos doigts.

Par ailleurs, il est aussi utilisé en danse par exemple. Il se retrouve installé sur une partie du corps, comme la rotule, pour vérifier la précision des mouvements du danseur. Il peut s’avérer très utile et intéressant pour analyser un mouvement à l’aide de nombres et données.

## Comment l’utiliser adéquatement:

Il ne faut pas le plier mais bien le courber, de sorte que le capteur garde une forme circulaire.

Il faut éviter de faire des formes en "S". Il ne faut pas l'utiliser du côté creux des articulations. Par exemple, il ne faudrait pas l’installer derrière le genou, mais sur la rotule.

## Référence:

 http://jeromeabel.net/files/pdf/DocumentationCapteurs.pdf

http://www.papierlogik.com/fr/technologie/fabrication/capteur-de-flexion

http://www.pobot.org/Capteur-flex-sensor-resistif.html

# Accéléromètre

## Définition et principes:

C'est un appareil électromécanique qui peut être fixé à un smartphone ou plus généralement à un appareil portable (ordinateur, consoles de jeu, appareils photo etc...). Ce capteur permet de mesurer l'accélération de l'appareil auquel il est fixé, ce qui permet de générer un déplacement en temps réel. On parle encore d'accéléromètre même s'il s'agit en fait de 3 accéléromètres qui calculent les 3 accélérations linéaires selon 3 axes orthogonaux. Le principe est basé sur la loi fondamentale de la dynamique F = m\*a.

Il existe deux méthodes de fabrication pour l’accéléromètre, le MEMS et le NEMS. Dans notre cas, c’est le MEMS qui nous intéresse. Il s’agit d’un assemblage de dispositifs électrique de taille micrométrique comprenant plusieurs éléments mécaniques pouvant réaliser la fonction du capteur ou d’actionneur.

## Avantage:

### Application sportive:

L’accéléromètre inclus dans le téléphone intelligent est utilisé pour déterminer la vitesse et l’accélération d’un coureur. Il peut aussi être utilisé dans un GPS par exemple pour permettre aux utilisateurs de connaître leur vitesse et donc d’éviter dans certains cas, certaines infractions en limitant leur vitesse.

### Info circulation:

L’accéléromètre de nos téléphones intelligents permet aussi de nous renseigner sur l’état du trafic routier. En effet, si la base de données détecte un ensemble important d’utilisateurs dans un même secteur ayant une accélération faible, ils peuvent en conclure qu’il y a des embouteillages.

### Rotation du téléphone:

Cette fonctionnalité parait naturelle, mais c’est la perturbation de l’accéléromètre qui permet d’effectuer la rotation de l’écran du téléphone pour passer de portrait à paysage. C’est aussi l’accéléromètre qui permet à certains jeux de prendre en compte cette rotation.

## Inconvénients:

On doit faire attention à l’atteinte à la vie privée. L’accéléromètre est un mouchard, il permet de connaitre la position de chaque individu en tout temps. Ils peuvent donc être envoyés dans des bases de données puis conservées. Il est donc possible de connaitre plusieurs informations sur un individu, notamment ses déplacements et les magasins, restaurants, maisons qu’il fréquente! L’autonomie de la batterie est aussi un autre inconvénient. L’accéléromètre travail sans cesse ce qui réduit énormément l’énergie de cette dernière.

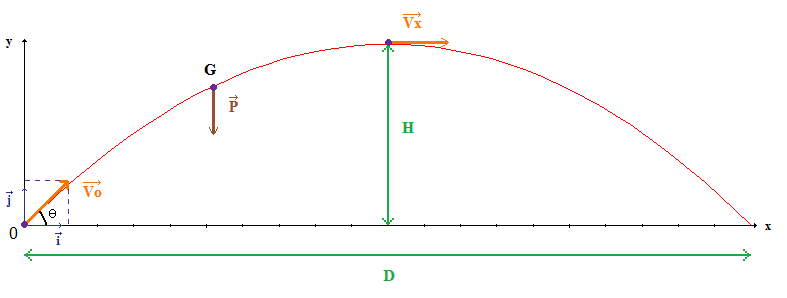
## Référence:

<http://www.telephoneportable-cours.com/acce/index.html>

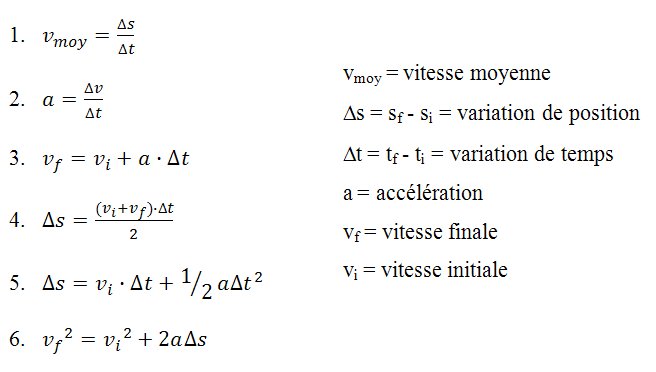
<http://fr.wikipedia.org/wiki/Acc%C3%A9l%C3%A9rom%C3%A8tre>

# Physique de l’application

Dans l’application, l’utilisateur doit d’abord positionner l’arc, pour ensuite lui donner une orientation et finalement une vitesse de tir. Le fonctionnement du mouvement de la flèche est similaire aux mouvements rectilignes uniformes (MRU) et aux mouvements rectilignes uniformes accélérés (MRUA) de la physique moderne. On donne d’abord une certaine gravité à la flèche. Ensuite, dépendamment de l’étirement donnée à la corde, on pose la vitesse de la flèche. Comme la vitesse en X est constante (vitesse initiale = vitesse finale), on peut trouver facilement la vitesse de la flèche en X grâce à un cosinus. Cependant, la vitesse en Y diffère au fil du temps. On doit donc commencer par trouver la vitesse initiale en Y. On peut par la suite déterminer la vitesse en Y au fil du temps avec la vitesse initiale, le temps écoulé et la gravité. Maintenant qu’on a nos vitesses, il est possible de trouver le déplacement en X et en Y de la flèche en appliquant les formule prévu à cet effet. Grâce aux données du positionnement de l’arc initialement choisies, on peut finalement trouver la position en X et en Y de la flèche en additionnant la position initiale avec le déplacement. Il faut par la suite orienter la flèche dans le sens qu’elle pointe. Il suffit de faire un rapport trigonométrique pour trouver un angle de rotation. Plusieurs rapports sont possibles, mais celui utilisé est le rapport entre la vitesse en Y et la vitesse de la flèche. Dans le cadre de cette application, quelques concepts ont été mit de côté comme la tension dans l’arc et la corde, les énergies cinétiques et potentiels et la friction de l’air sur la flèche. Néanmoins, nous nous sommes concentrés sur la trajectoire parabolique de la flèche qui est à notre avis assez réaliste.



## Formules utilisés:



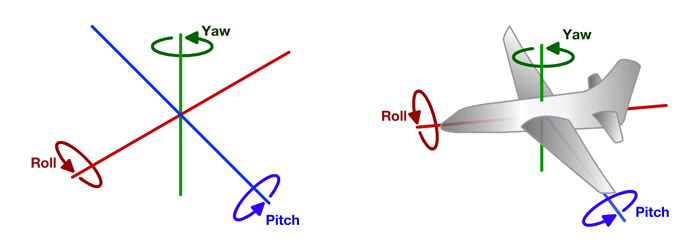
## Référence:

<http://bv.alloprof.qc.ca/physique/generalites-en-physique/les-principales-formules-utilisees-en-physique.aspx>

# Fonctionnement du gant

Tout d’abord, il est indispensable de comprendre le fonctionnement du gyroscope pour comprendre comment le gant fonctionne. En effet, une grande majorité du projet se base sur le fonctionnement du gyroscope ainsi que les angles que cet appareil nous envoie. Voir la documentation du gyroscope ci-haut.

Ensuite, les angles utilisés pour contrôler la souris sont nommés « La précession » (Yaw), « la nutation »(Pitch) et « la rotation propre »(Roll). L’angle « La précession » sert notamment à déplacer le curseur selon l’axe des «X» à l’écran. L’angle la nutation sert, quant-à lui, à déplacer le curseur selon l’axe des «Y» à l’écran. L’angle « la nutation » ne sert à rien de concret dans le projet pour l’instant, elle est présente pour montrer son existence. Voici une image représentant ces angles :



Source : [http://theboredengineers.com/WordPress3/wp-content/uploads/2012/05/la nutationRollLa précession.png](http://theboredengineers.com/WordPress3/wp-content/uploads/2012/05/PitchRollYaw.png)

Le devant de l’avion ci-dessus, est l’endroit où les doigts sont placés dans le gant. Traduit en mouvement, pour effectuer « la précession » (Yaw), il s’agit simplement de bouger la main de gauche à droite en s’assurant que cette dernière est parallèle au sol. Pour effectuer une rotation «Pitch» (la nutation), il faut bouger la main vers le haut, ou vers le bas, en s’assurant de ne pas bouger le bras, mais bien la main uniquement. Quant-à la rotation propre (roll), il faut simplement tourner le poignet comme si on voulait tourner une poignée de porte.

## Le quaternion

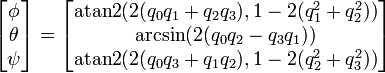
Par ailleurs, ces angles sont calculés en utilisant un quaternion qui lui est calculé en utilisant les angles bruts envoyés par le gyroscope. Un quaternion est un objet mathématique extrêmement complexe qui dépasse les connaissances vues dans les cours de mathématiques au cégep. Cependant, il est relativement simple de comprendre qu’un quaternion possède trois éléments : i, jet ksatisfaisant les relations quaternioniques suivantes :

i^2 = j^2 = k^2 = ijk = -1.

On constate notamment qu’un nombre au carré doit être égal à -1, qui est impossible sans la notion de nombre imaginaire. Inutile de s’avancer plus dans le domaine, puisque plusieurs ouvrages ont déjà expliqués la notion.

## Calcul des angles

En outre, pour calculer les angles La précession, la nutation, la rotation propre en ordre, voici la belle et simple formule se basant sur les quaternions. Où q est un quaternion, et le chiffre en index est l’élément du quaternion.



Source : <http://upload.wikimedia.org/math/a/2/9/a2925987257bc7469187cfc3c18da853.png>

La fonction atan2 renvoie l’angle entre deux points. Dans la figure ci-dessus, Phi représente la rotation propre (Roll), Thêta représente la nutation(Pitch) et Psi représentant la précession (Yaw).

Finalement, le calcul de la position de la souris s’effectue selon la méthode de mouvement choisie par l’utilisateur. Ces méthodes de mouvement sont :

1. Vitesse constante : le curseur se déplace toujours à la même vitesse, peu importe l’angle. La direction varie cependant.
2. Variation d’angle : le curseur se déplace plus rapidement plus l’angle est grand.

Les calculs sont simples :

## Calcul de la position

### vitesse constante :

Si Angle > (Angle Initiale + Angle Mort) alors  
Position = Dernière Position + Vitesse.

(Le calcul est le même dans le sens négatif, seulement les signes sont inversés)

### Avec la variation d’angle :

Si Angle > (Angle Initiale + Angle Mort) alors  
Position = Dernière Position + vitesse \* [abs (Angle - (Angle Initiale + Angle Mort))] / Sensibilité

(Le calcul dans le sens négatif est le même, seulement la première addition varie)

Où angle: l’angle actuel  
Angle initiale : l’angle initiale du gant  
Angle Mort : le delta angle où rien ne se produit  
Position : la position actuelle

Vitesse : la vitesse configuré par l’utilisateur,  
Sensibilité : la sensibilité du mouvement configuré.   
Note : une sensibilité de 6 est plus petite qu’une sensibilité de 3, c’est alors l’inverse de la sensibilité. (Une sensibilité de 6 équivaut à ⅙)

**Référence:**

<http://en.wikipedia.org/wiki/Conversion_between_quaternions_and_Euler_angles>

<http://theboredengineers.com/2012/05/the-quadcopter-basics/>

<http://fr.wikipedia.org/wiki/Quaternion>

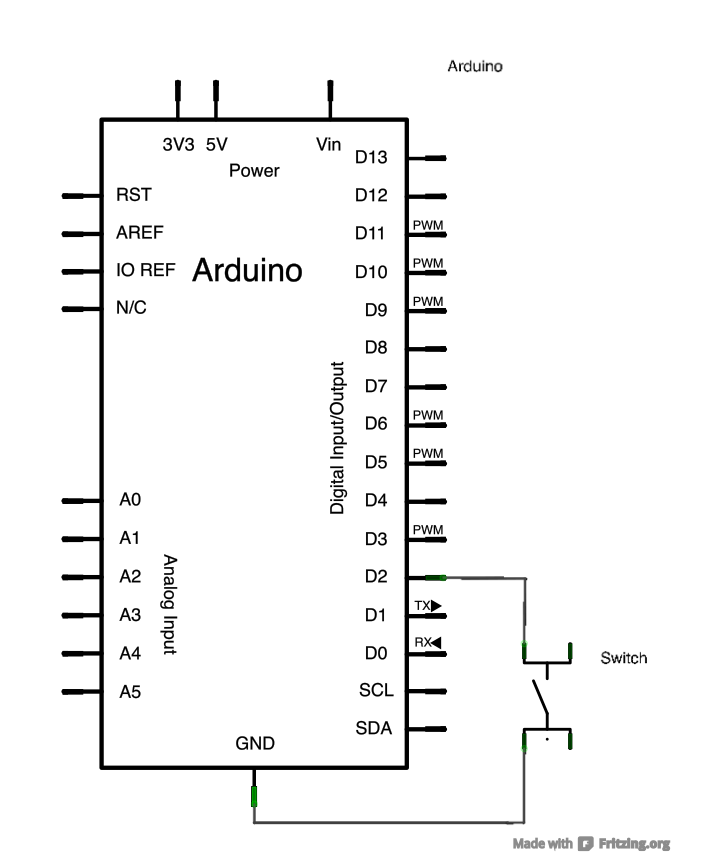
# Capteur de contact (input pull-up)

## Propriétés des pins configurés en «INPUT\_PULLUP»

À l’intérieur même du arduino sont soudés des résistances de 20 000 ohms. Ces résistances sont accessibles via le code inséré à l’intérieur du arduino même. En effet, elles sont accessibles en mettant un mode de pin (pin Mode) à «INPUT\_PULLUP». Du coup, cette commande inverse le fonctionnement interne des entrées où «HIGH» (1) signifie que le capteur de contact est levé, alors que «LOW» (0) veut dire qu’il y a contact.

Dans le cas du gant, lorsque l’index ou le majeur touche au pouce, on dit alors qu’il y a contact, donc l’arduino renvoie l’information  «LOW»  que l’on interprète comme un «clic» de souris.

## Fonctionnement du circuit

Tout d’abord, il est indispensable d’avoir un fil connecté au pin en «INPUT\_PULLUP», c’est la première moitié du capteur de contact.

Ensuite, il est nécessaire d’avoir le deuxième fil connecté dans le «Ground».

Dans le cas d’un contact entre le fil du pin en «INPUT\_PULLUP» et du «Ground», renvoyant ainsi la valeur «LOW» puisque c’est la lecture du «Ground» qui est effectuée.

Dans le cas contraire, où il n’y a pas de contact,  c’est le courant qui passe continuellement à travers le résisteur interne qui est lu.

## Référence

<http://www.arduino.cc/en/Tutorial/DigitalPins>

http://www.arduino.cc/en/uploads/Tutorial/inputPullupSerial\_sch.png