Rapport de projet semestre 3

Projet Application d’aide à la personne



Projet encadré par

M. SAMER MOHAMED

DUT Réseaux et télécom

# Sommaire :

Sommaire : 1

Intro : 1

Détail du travail effectué : 1

Git hub 3

LabVIEW 4

Notes lab view 4

Application android 4

Conclusion : 4

Annexes 4

Glossiare 4

GitHub 4

LabVIEW 7

Android studio 8

Bibliographie 9

GitHub 9

LabVIEW 9

android studio 9

Résumé 9

# Introduction

Dans le cadre des notre seconde et dernière année d’IUT R&T, il nous est proposé un projet durant le semestre 2, 3 et 4 nous permettant de mettre en pratique nos connaissances et en développer de nouvelles au travers d’un cahier des charges ayant pour finalité la conception d’une application d’aide à la personne

Pour cela, nous allons tout d’abord dans un premier temps nous allons voir comment nous nous sommes réparti les charges de travail.

Puis nous allons observer les paramètres que nous allons devoir respecter de sorte que le patient ne puisse pas se blesser.

Enfin, nous verrons la réalisation du projet sous forme de trois grandes parties : tout d’abord le site/logiciel GitHub qui nous a permis de mettre au commun nos avancement pour le projet. Puis une deuxième partie sur le logiciel LabVIEW avec la conception des actions de l’orthèse. Et enfin, nous verrons dans une troisième et dernière partie la création de l’application, grâce au logiciel Android Studio, permettant de faire marcher l’orthèse.

# I Détail du travail effectué

Dans ce semestre trois nous avons pu commencer à développer l’application et à écarter toutes les voies sur lesquelles on va se diriger pour finir ce projet au semestre 4

Voici la vue générale de toutes les réunions de mise en commun du 3éme semestre.

Tableau d’attribution de lettres aux étudiants

|  |  |
| --- | --- |
| RAMOS GONCALVES Wilson | W |
| IM-SAROEUN Paul | P |
| NICOLAS Jules | J |
| BELLANGER Liam | L |

Tableau des réunions effectuées

|  |  |  |  |
| --- | --- | --- | --- |
| Réunion | Thème | Contributeurs | Dates |
| 1 | Répartition des tâches | W,P | Jeudi 09/12 |
| 2 | Recherches + début d’application | W,P | Dimanche 12/12 |
| 3 | Mise en commun | W,P,J | Lundi 03/01 |

#### Rapport de réunion jeudi 09/12 :

Discussion sur les éléments qu’on va mettre dans le rapport :

* Vidéo du fonctionnement de l'orthèse
* Explication du fonctionnement du code LabVIEW
* Explication de l'implémentation java LabVIEW
* Création de l’application
* Répartition du travail:(pour l’instant Wilson et Paul)
* Paul :
  + Développement de l'application Android
  + Explication du fonctionnement
* Wilson :
  + Recherches sur l'implémentation app LabVIEW
  + Recherches sur mode de communication

#### Rapport de réunion jeudi 12/12 :

* Paul :
  + Début du développement de l’application
  + Création d’un dépôt GitHub
* Wilson
  + Rédaction rapport + recherches
  + Création d’un dépôt GitHub

#### Rapport de réunion jeudi 03/01 :

* Wilson :
  + Mise en place du repository GitHub
  + LabVIEW
* Paul :
  + Développement de l’application
* Jules
  + Début de rédaction du rapport

#### Rapport de réunion jeudi 14/01 :

* Wilson :
  + Dev application
  + Rédaction rapport
* Paul :
  + Dev application
  + Rédaction rapport

## 1.1 Git hub

Pendant le développement de l’application on s’est aperçu que c’était difficile de s’accorder et de se partager le travail de développement nous avons donc décidé d’utiliser GitHub.

Dans cette partie Premièrement nous allons présenter la plateforme GitHub ensuite nous expliquerons les bonnes pratiques a ne pas negliger et pour finir montrer comment nous l’avons utilisé pour collaborer tous ensemble.

GitHub est une plateforme open source de gestion de versions et de collaboration destiné aux développeurs. Elle utilise Git un système de gestion de code open source crée par Linus Torvalds dans le but d’accélérer le développement logiciel.

Git permet de stocker le code source d'un projet et de suivre l'historique complet de toutes les modifications apportées à ce code. Grâce aux outils qu'elle fournit pour gérer les conflits éventuels résultant des changements apportés par plusieurs développeurs, il est possible de collaborer efficacement sur un même projet. GitHub facilite la programmation collaborative en mettant une interface Web à disposition du référentiel de code de Git, ainsi que des outils d'administration favorisant la collaboration.

Nous allons donc vous montrer étape par étape comment créer et gérer un dépôt git et les commandes de base et les bonnes pratiques pour gérer un projet collaboratif.

Tout d’abord nous devons installer git sur nos machines respectives :

-on va sur <https://git-scm.com/downloads> et on télécharge la dernière version

Ensuite sur le terminal git bash on se place dans le répertoire du projet avec la commande cd « chemin » :

Une image contenant texte

Description générée automatiquement

Sur ce terminal on tape la commande git add –all et git commit pour indiquer à Git qu'il doit gérer tous les fichiers du dossier et d'enregistrer cette action.

Enfin on envoi nos fichiers sur GitHub en utilisant la commande git push -u Origin main

Notre dépôt est donc créé nous pouvons donc passer à notre utilisation de GitHub.

À présent nous allons vous monter comment on a utilisé git.

Dans un premier temps nous nous sommes tous créé un compte GitHub, installé git sur nos machines puis finalement créé le dépôt du projet :

Une image contenant texte, moniteur, noir, capture d’écran

Description générée automatiquement

Grâce à l’url du dépôt nous avons tous pu créer une version du projet sur nos machines grâce à la commande *git clone* [*https://github.com/Twisterbad1/Projet-tut*](https://github.com/Twisterbad1/Projet-tut) , on se retrouve donc avec un répertoire contenant un .git qui permettra le versioning et tous les éléments mis au préalable par tous les membres du groupe :

Une image contenant texte

Description générée automatiquement

Ce dépôt a principalement servi pour le développement de l’application donc à chaque modification du code source nous effectuons les commandes *git commit -a -m « nom la modification »* puis *git push* pour mettre à jour le dépôt

Nous pouvons désormais parler de la partie LabVIEW

## 1.2 LabVIEW

LabVIEW est le logiciel qui va nous permettre de contrôler la carte EPOS qui elle-même contrôle le moteur de notre orthèse.

Dans cette parie on va dans un premier temps définir LabVIEW et vous montrer les prototypes de communication de notre application avec le server TCP/IP

LabVIEW est un environnement de développement spécialisé en informatique industrielle et scientifique. Sa particularité est qu'il s'appuie sur le langage G, créé par National Instruments, qui est entièrement graphique. Il permet de créer des logiciels complexes tout en facilitant la programmation. Grâce à ses librairies de fonctions dédiées à l'acquisition de données, l'instrumentation, à l'analyse mathématique des mesures, mais également grâce à la création rapide d'interfaces graphiques de qualité et le codage simplifié.

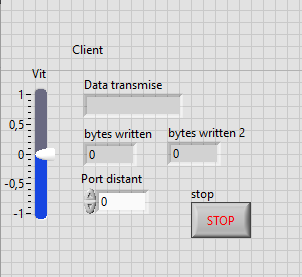
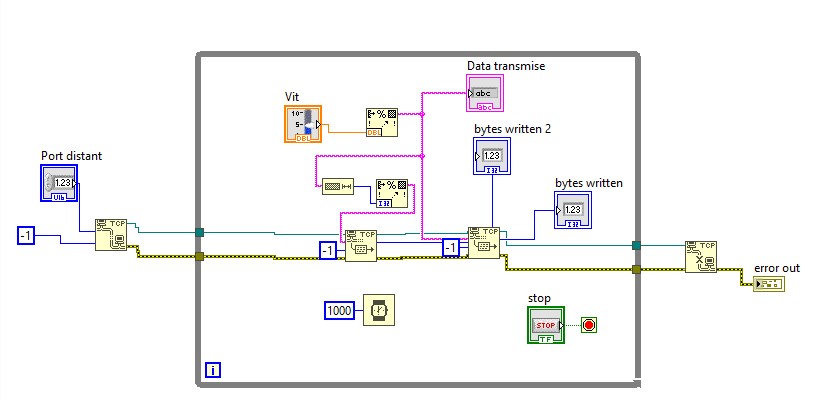
LabVIEW est particulièrement recommandé pour développer des systèmes de contrôle, supervision et les bancs de test et mesure c’est donc l’outil parfait pour notre projet

Au semestre 2, nous avons imaginé un prototype de contrôle de l’orthèse il avait des failles donc notre tuteur nous a fournis le programme permettant de contrôler la carte EPOS. Donc à ce semestre nous avons décidé de nous concentrer sur la partie communication en cherchant des protocoles qui nous permettant de de faire communiquer notre téléphone avec la carte.

On a choisi le protocole tcp/ip car il est plus fiable et dans le cas de notre application cela est essentiel.

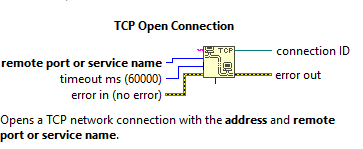
On va donc vous présenter notre prototype client-serveur TCP/ip

Client (Schéma en annexe)



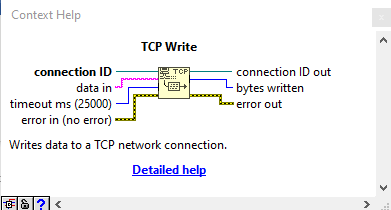
Le principe est simple on a sur l’interface graphique une glissière et plusieurs afficheurs un qui affiche les données transmises au serveur et les 2 autres qui affichent les octets envoyés ensuite on a un on a une option nous permettant de choisir le port de communication avec le serveur TCP sur le schéma nous avons différents modules :

un module de connexion :



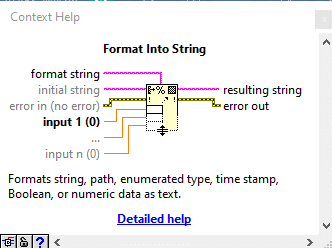
Ce module permet de spécifier un port une adresse IP ou un service et assure la connexion entre le client et le serveur

À module TCP Write :



Permettant la transmission des données reçu en paramètre

Un module format into string :



Qui nous permet de transformer une donnée sous format texte

Un module string lenght :

Une image contenant texte

Description générée automatiquement

Qui permet de lire la longueur d'un texte et le transformer en chiffres

Un module White ms:

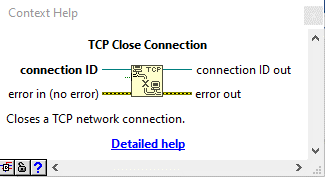
Une image contenant texte

Description générée automatiquement

Nous permettons d'inclure une temporisation

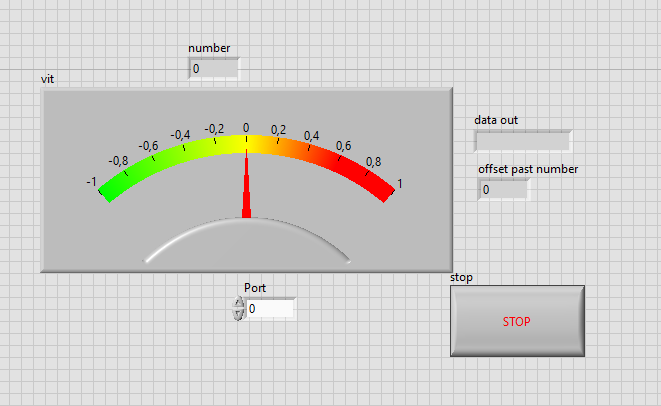
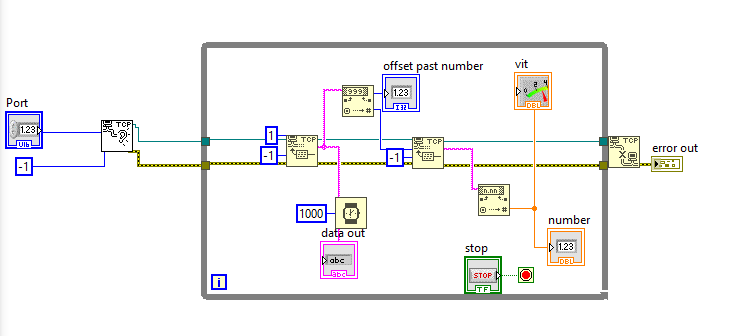
et pour finir.

Un module TP close connection :

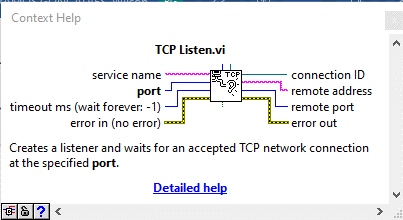


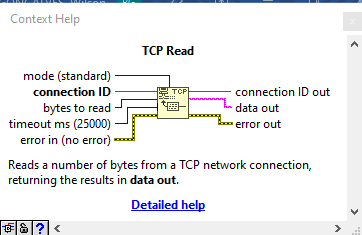
qui nous permet de mettre fin à la connexion TCP

Quand on se connecte au serveur on a le module vitesse qui est une glissière qui envoie des données sous forme de strig au serveur TCP qui analyse ces données là pour contrôler un afficheur de vitesse que je montrerai ci-dessous



L'architecture du serveur est plus ou moins la même que celle du client sauf que à la place de TCP connexion on a module TCP Listen et à la place du module TP write on a un module TCP read qui lit les informations reçues par le client





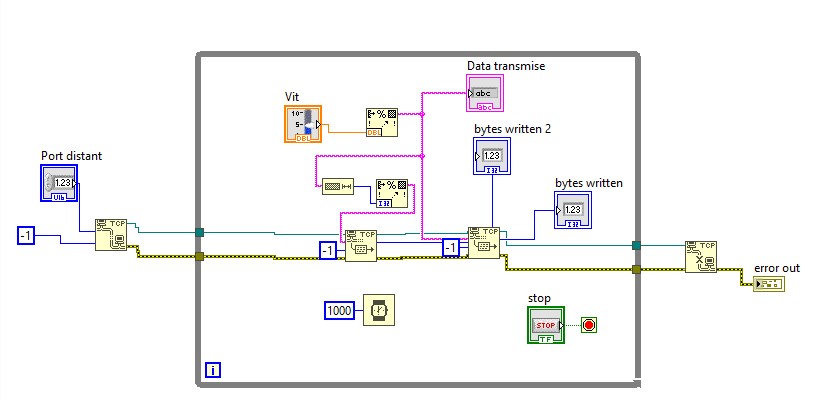
Le fonctionnement du prototype :

Dans un premier temps on démarre le serveur TPC puis le client, ensuite on a la glissière qui nous permet de contrôler la vitesse affichée sur le serveur si on choisit 0 on aura une vitesse de 0 si on choisit un on aura une vitesse de un ainsi de suite.

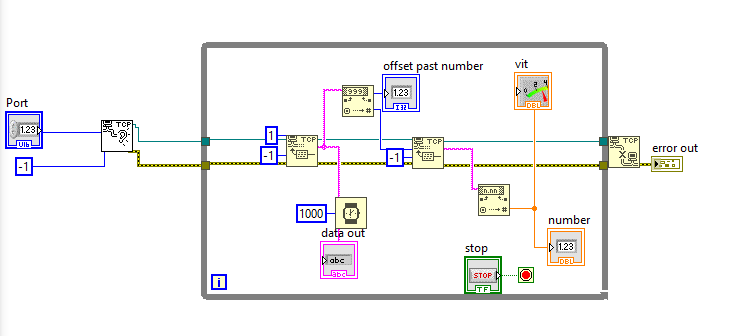
## 1.2 Application android

# II CONCLUSION

# III Annexes

Schéma Client TCP/ip

Schema server TCP/ip



# IV Glossiare

## 4.1 GitHub

**Access token**

Un token qui est utilisé à la place d’un mot de passe pour effectuer des opérations via HTTPS avec GIT dans un terminal ou une API

**Branche**

**Une branche est une version parallèle d'un dépôt. Elle est contenue dans le dépôt, mais n'affecte pas la branche primaire ou principale, ce qui nous permet de travailler librement sans perturber la version "courante". Lorsque on a apporté les modifications souhaitées, on peut fusionner la branche avec la branche principale pour publier les changements.**

**Checkout**

On utilise git checkout dans un terminal pour créer une nouvelle branche, changer la branche de travail actuelle pour une branche différente, ou même pour passer à une version différente d'un fichier provenant d'une branche différente avec git checkout [nom de la branche] [chemin du fichier]. La commande "checkout" met à jour tout ou une partie de l'arbre de travail avec un objet ou un blob de la base de données des objets, et met à jour l'index et le HEAD si l'arbre de travail entier pointe vers une nouvelle branche.

**Clone**

Un clone est une copie d'un dépôt sur notre ordinateur plutôt que sur le serveur de GitHub, ou l'acte de faire cette copie. Lorsqu'on fait un clone, on peut éditer les fichiers dans son éditeur préféré et utiliser Git pour garder la trace de ses modifications sans avoir à être en ligne. Le dépôt qu'on a cloné est toujours connecté à la version distante de sorte qu'on peut push les modifications locales vers la version distante pour les synchroniser quand on est en ligne.

**Collaborateur**

Un collaborateur est une personne ayant un accès en lecture et en écriture à un dépôt et qui a été invitée à contribuer par le propriétaire du dépôt.

**Commit**

Un commit, ou "révision", est une modification individuelle d'un fichier (ou d'un ensemble de fichiers). Lorsqu'on réalise un commit pour sauvegarder son travail, Git crée un identifiant unique (alias le "SHA" ou "hash") qui permet de garder une trace des modifications spécifiques commises ainsi que de l'auteur et de la date. Les commits contiennent généralement un message de commit qui est une brève description des changements effectués.

**Git**

Git est un programme open source pour le suivi des changements dans les fichiers texte. Il a été écrit par l'auteur du système d'exploitation Linux et constitue la technologie de base sur laquelle est construite GitHub, l'interface sociale et celle de l'utilisateur.

**Pages GitHub**

Également appelé Pages. Un service d'hébergement de sites statiques conçu pour héberger vos pages personnelles, d'organisation ou de projet directement à partir d'un dépôt GitHub.

**Main**

La branche de développement par défaut. Chaque fois que vous créez un dépôt Git, une branche nommée main est créée, et devient la branche active. Dans la plupart des cas, elle contient le développement local, bien que cela soit purement par convention et ne soit pas nécessaire.

**Merge**

La fusion prend les changements d'une branche (dans le même dépôt ou dans une branche secondaire), et les applique à une autre. Cette opération s'effectue souvent sous la forme d'une "pull request" (que l'on peut considérer comme une demande de fusion), ou via la ligne de commande. Une fusion peut être effectuée par une demande de pull via l'interface web de GitHub.com s'il n'y a pas de changements conflictuels, ou peut toujours être effectuée via la ligne de commande.

**Merge conflict**

Une différence qui se produit entre les branches fusionnées. Les conflits de fusion se produisent lorsque des personnes apportent des modifications différentes à la même ligne d'un même fichier, ou lorsqu'une personne modifie un fichier et qu'une autre personne supprime le même fichier. Le conflit de fusion doit être résolu avant que vous puissiez fusionner les branches.

**Open source**

Les logiciels libres sont des logiciels qui peuvent être librement utilisés, modifiés et partagés (sous forme modifiée ou non) par quiconque. Aujourd'hui, le concept de " open source " est souvent étendu au-delà des logiciels, pour représenter une philosophie de collaboration dans laquelle les matériaux de travail sont mis à disposition en ligne pour que chacun puisse les utiliser, les modifier, en discuter et y contribuer.

**Pull**

Pull fait référence au moment où l'on récupère des modifications et où on les fusionne. Par exemple, si quelqu'un a modifié le fichier distant sur lequel on travaille tous les deux, on voudra transférer ces modifications dans la copie locale pour qu'elle soit à jour.

**Pull request**

Les demandes de Pull sont des propositions de modifications d'un référentiel soumises par un utilisateur et acceptées ou rejetées par les collaborateurs du référentiel. Comme les questions, les demandes de retrait ont chacune leur propre forum de discussion.

**Push**

Push signifie qu'on envoie les modifications apportées à un dépôt distant sur GitHub.com. Par exemple, si on modifie quelque chose localement, on peut pousser ces modifications pour que d'autres puissent y accéder.

**Repository**

Un dépôt est l'élément le plus fondamental de GitHub. Il est plus facile de l'imaginer comme le dossier d'un projet. Un dépôt contient tous les fichiers du projet (y compris la documentation), et stocke l'historique des révisions de chaque fichier. Les dépôts peuvent avoir plusieurs collaborateurs et peuvent être publics ou privés.

## 4.2 LabVIEW

**Boucle While**

Structure en boucle qui répète une section de code jusqu'à ce qu'une condition ait lieu.

**Bouts de fils de liaison**

Fils de liaison tronqués qui apparaissent à côté de terminaux non câblés lorsque vous déplacez l'outil Bobine sur un VI ou un nœud de fonction.

**Connecteur**

Partie du VI ou du nœud de fonction qui contient les terminaux d'entrée et de sortie. Les données passent par un nœud et sortent d'un nœud par le biais d'un connecteur.

Zone dans le coin droit supérieur de la face-avant ou du diagramme qui affiche le modèle des terminaux du VI. Il définit les entrées et les sorties que vous pouvez câbler à un VI.

**Entier**

Tous les nombres entiers naturels, leur opposé, ou zéro.

**Exécution continue**

Mode d'exécution dans lequel un VI s'exécute de manière répétitive jusqu'à ce que l'opérateur l'arrête. Cliquez sur le bouton **Exécuter en continu** pour exécuter un VI de manière répétitive.

**Incrément de temps**

Intervalle de t à t + dt.

**Info-bulle**

Petites bannières de texte jaunes qui identifient le nom du terminal et facilitent l'identification des terminaux pour le câblage.

**LabVIEW**

Laboratory Virtual Instrument Engineering Workbench — LabVIEW est un langage de programmation graphique qui utilise des icônes au lieu de lignes de texte pour créer des applications.

**TCP/IP**

Transmission Control Protocol/Internet Protocol — Format standard de transmission de données en packets d'un ordinateur à un autre. Les deux composantes du TCP/IP sont le TCP, qui concerne la construction de paquets de données, et le IP, qui les achemine d'un ordinateur à un autre.

**UDP**

User Datagram Protocol

## 4.3 Android studio

**Android SDK**

Android SDK, ou « Software Developement Kit » du français Kit de développement logiciel), est une boîte à outil qui permet aux développeurs de tester leurs applications sur différentes versions d'Android, en les émulant. Le SDK regroupe les outils nécessaires à la connexion et à l'envoi de commandes ADB. Il est utile aux utilisateurs qui veulent modifier leur terminal Android, installer des ROM ou encore simplement installer des drivers USB sur l'ordinateur.

**Boot**

Le boot désigné la première étape de démarrage d'un système informatique. Sur les appareils mobiles Android, on parle de boot pour qualifier le démarrage du système, du recovery custom, du mode download ou encore du mode bootloader. Par extension, le terme Reboot veut dire redémarrage.

**Android Studio**

Android Studio est l’environnement de développement intégré (IDE) officiel pour les appareils Android. Il prend en charge Windows, Mac et Linux, est basé sur le logiciel IntelliJ IDEA de JetBrains et utilise Java, Cotlin et C ++. Le logiciel gratuit peut être téléchargé à partir d’Android Studio site officiel et contient des fonctionnalités qui incluent un éditeur de mise en page riche et une intégration intégrée avec Google Cloud. Outre Android Studio, les applications Android peuvent être conçues sur Visual Studio, AIDE, Eclipse, Droid4X, Bluestacks, etc.

**Versions Android**

Les systèmes d’exploitation Android ont vu de nombreuses versions différentes depuis son premier lancement, qui retracent une évolution substantielle au fil des ans. Les versions Android suivantes ont été publiées par Google.

# V Bibliographie

## 5.1 GitHub

<https://zone.ni.com/reference/fr-XX/help/371361R-0114/lvconcepts/glossary/>

## 5.2 LabVIEW

<https://www.youtube.com/watch?v=rVwJ1mIwuqM>

<https://www.myandroidsolutions.com/2012/07/20/android-tcp-connection-tutorial/>

[https://www.ni.com/fr-fr/support/documentation/supplemental/06/basic-tcp-ip-communication-in-LabVIEW.html](https://www.ni.com/fr-fr/support/documentation/supplemental/06/basic-tcp-ip-communication-in-labview.html)

<https://www.youtube.com/watch?v=iiLmwBGUsyU>

<https://www.youtube.com/watch?v=LMtdLqmRnVU>

<https://www.youtube.com/watch?v=L3zirD3axCk>

<https://www.youtube.com/watch?v=29y4X65ZUwE>

<https://zone.ni.com/reference/fr-XX/help/371361R-0114/lvconcepts/glossary/>

## 5.3 Android studio

<https://www.youtube.com/watch?v=IibybM4oM1w&t=72s>

<https://www.youtube.com/watch?v=6pT112_KGkc>

<https://www.youtube.com/watch?v=IibybM4oM1w>

<https://www.youtube.com/watch?v=9Rwopuah2Q0>

# VI Résumé

Imperator resistente convolantibus ambigebat cogitationibus Alamanna capesseret ventum Alamanna callibus telis ubi vi cogitationibus difficultatibus resistente itaque ad suspendere capesseret vi convolantibus conpage nimia vi navium et telis multis videretur itaque suspendere Romani vi supercilia itaque vi plurimis Rheni capesseret quid telis plurimis quid suspendere nimia Rauracum conpage conpage ritu est et pontem pontem multis telis ambigebat ambigebat ritu multis convolantibus conpage inpossibile undique suspendere multitudine ubi est fluminis prope ad attonitus cum convolantibus itaque videretur vetabantur vi imperator multis obrutis attonitus fluminis telis obrutis Rauracum fluminis imperator telis Romani conpage prope quid Rauracum convolantibus suspendere quid Rauracum nive vetabantur. Imperator resistente convolantibus ambigebat cogitationibus Alamanna capesseret ventum Alamanna callibus telis ubi vi cogitationibus difficultatibus resistente itaque ad suspendere capesseret vi convolantibus conpage nimia vi navium et telis multis videretur itaque suspendere Romani vi supercilia itaque vi plurimis Rheni capesseret quid telis plurimis quid suspendere nimia Rauracum conpage conpage ritu est et pontem pontem multis telis ambigebat ambigebat ritu multis convolantibus conpage inpossibile undique suspendere multitudine ubi est fluminis prope ad attonitus cum convolantibus itaque videretur vetabantur vi imperator multis obrutis attonitus fluminis telis obrutis Rauracum fluminis imperator telis Romani conpage prope quid Rauracum convolantibus suspendere quid Rauracum nive vetabantur. Imperator resistente convolantibus ambigebat cogitationibus Alamanna capesseret ventum Alamanna callibus telis ubi vi cogitationibus difficultatibus resistente itaque ad suspendere capesseret vi convolantibus conpage nimia vi navium et telis multis videretur itaque suspendere Romani vi supercilia itaque vi plurimis Rheni capesseret quid telis plurimis quid suspendere nimia Rauracum conpage conpage ritu est et pontem pontem multis telis ambigebat ambigebat ritu multis convolantibus conpage inpossibile undique suspendere multitudine ubi est fluminis prope ad attonitus cum convolantibus itaque videretur vetabantur vi imperator multis obrutis attonitus fluminis telis obrutis Rauracum fluminis imperator telis Romani conpage prope quid Rauracum convolantibus suspendere quid Rauracum nive vetabantur. Imperator resistente convolantibus ambigebat cogitationibus Alamanna capesseret ventum Alamanna callibus telis ubi vi cogitationibus difficultatibus resistente itaque ad suspendere capesseret vi convolantibus conpage nimia vi navium et telis multis videretur itaque suspendere Romani vi supercilia itaque vi plurimis Rheni capesseret quid telis plurimis quid suspendere nimia Rauracum conpage conpage ritu est et pontem pontem multis telis ambigebat ambigebat ritu multis convolantibus conpage inpossibile undique suspendere multitudine ubi est fluminis prope ad attonitus cum convolantibus itaque videretur vetabantur vi imperator multis obrutis attonitus fluminis telis obrutis Rauracum fluminis imperator telis Romani conpage prope quid Rauracum convolantibus suspendere quid Rauracum nive vetabantur. Imperator resistente convolantibus ambigebat cogitationibus Alamanna capesseret ventum Alamanna callibus telis ubi vi cogitationibus difficultatibus resistente itaque ad suspendere capesseret vi convolantibus conpage nimia vi navium et telis multis videretur itaque suspendere Romani vi supercilia itaque vi plurimis Rheni capesseret quid telis plurimis quid suspendere nimia Rauracum conpage conpage ritu est et pontem pontem multis telis ambigebat ambigebat ritu multis convolantibus conpage inpossibile undique suspendere multitudine ubi est fluminis prope ad attonitus cum convolantibus itaque videretur vetabantur vi imperator multis obrutis attonitus fluminis telis obrutis Rauracum fluminis imperator telis Romani conpage prope quid Rauracum convolantibus suspendere quid Rauracum nive vetabantur.