

# Artik & PapART

réunion d'avancement t+14 mois

# Artik & PapART

réunion d'avancement t+14 mois

Contenu de la présentation:

- Rappel des objectifs du projet.
- Discussion des objectifs.
- historique et déroulement du projet.

# Artik & PapART

réunion d'avancement t+14 mois

Contenu de la présentation:

- Rappel des objectifs du projet.
- Discussion des objectifs.
- historique et déroulement du projet.
- Bilan t+14.
- Avancées logicielles.
- Discussions sur les objectifs

# Objectifs

(Présentation précédente)

- Création d'un Toolkit stable, optimisé, facile à prendre en main.
- Outils de recherche.
- Outils de démonstrations.
- Permettant le transfert des résultats.

# Objectifs

(Présentation précédente)

- Création d'un Toolkit stable, optimisé, facile à prendre en main.
- Outils de recherche.
- Outils de démonstrations.
- Permettant le transfert des résultats.

Objectifs actuels:

- Création d'un prototype «one button».
- Stabilisation du SDK.
- Création de deux applications: Dessin et jeu interactif.

# Objectifs - startup

## [Droit]

Étude sur la possibilité de vendre le SDK, dans sa version d'aujourd'hui.

## [Application]

Vendre la technologie au lieu de la donner / échanger contre de la main d'oeuvre.

## [Hardware]

Créer un matériel «maison» au lieu de Kinect + projecteur.

-> Trouver des partenaires pour l'électronique et l'optique pour chiffrer le coût.

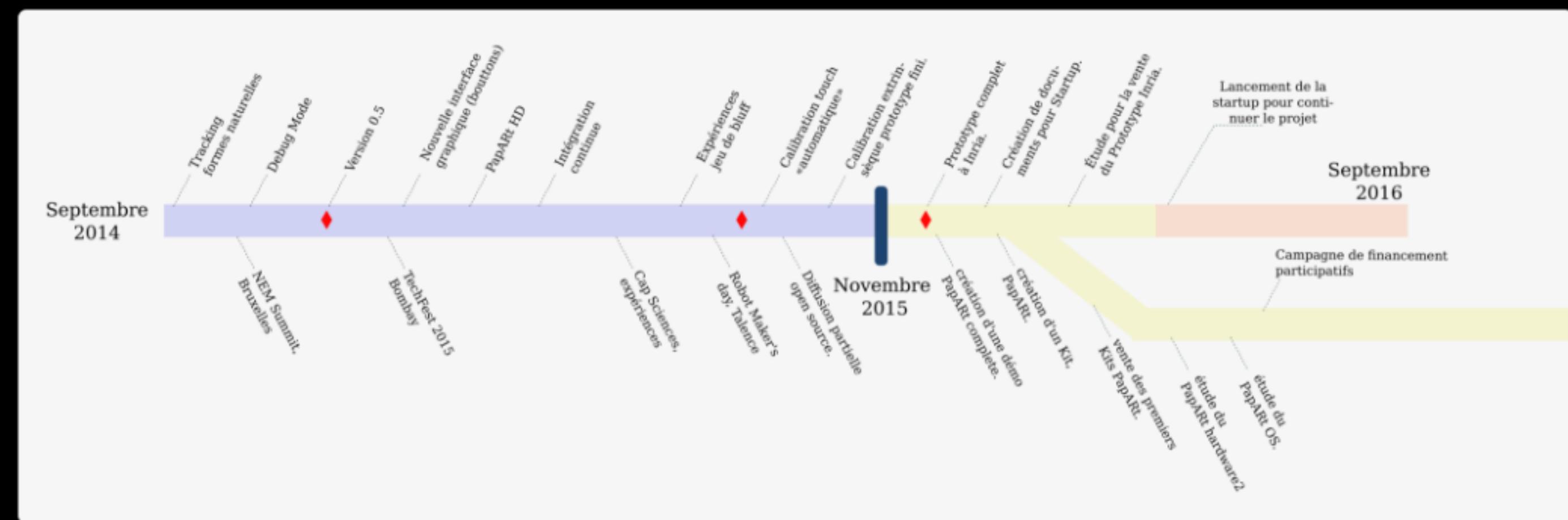
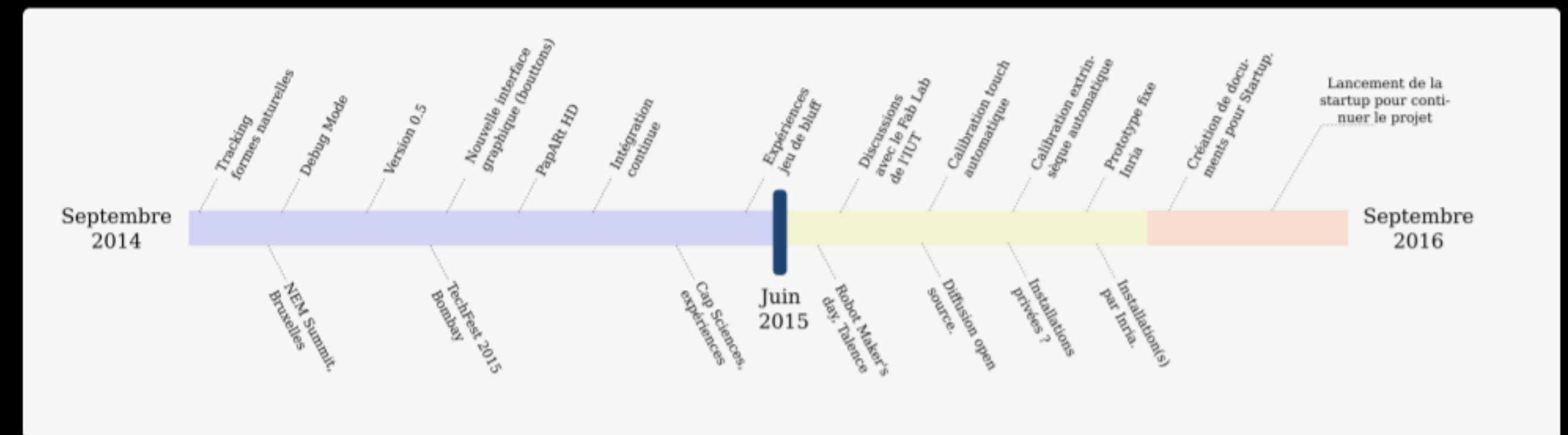
## [Financement]

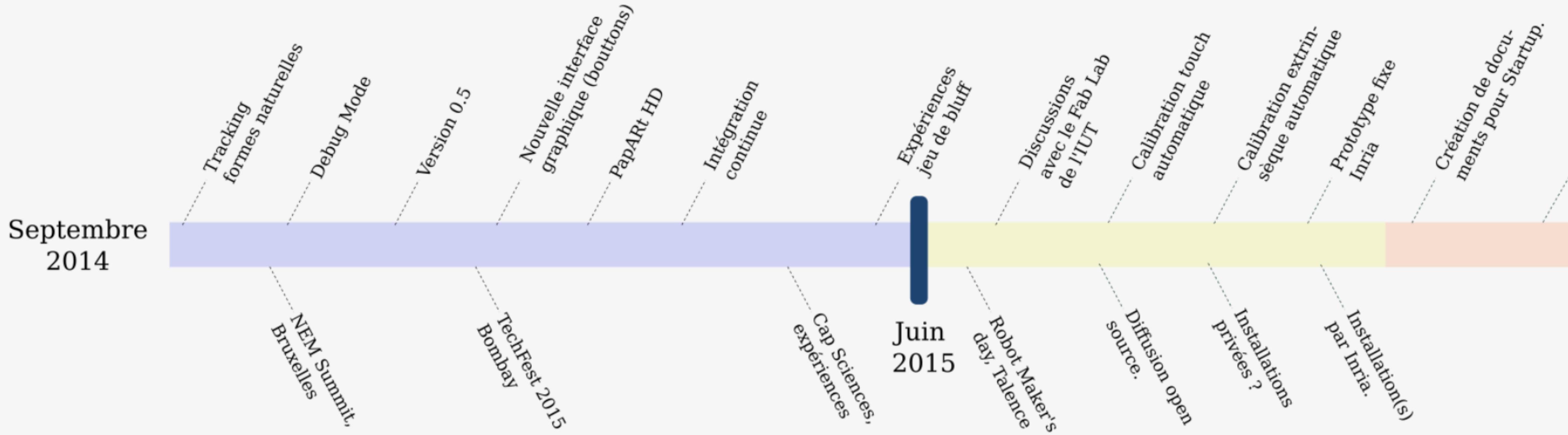
Candidater à des financements publics et faire une campagne de financement participatif.

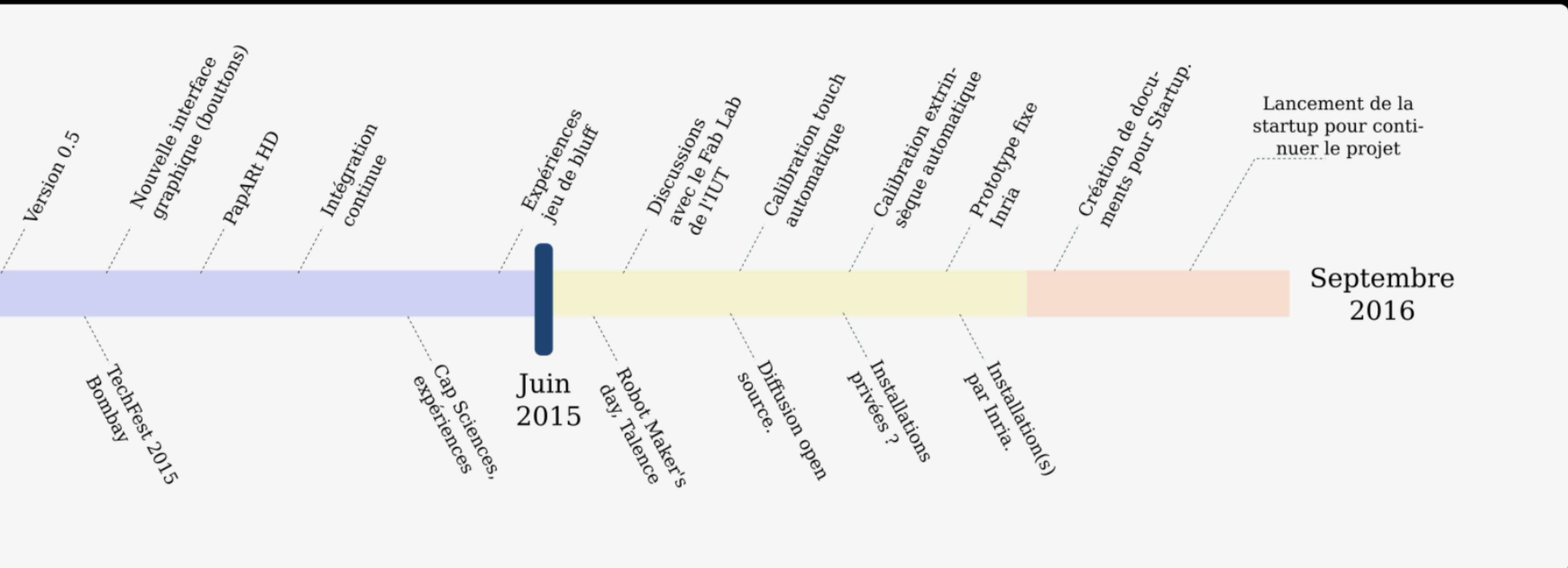
## [Software]

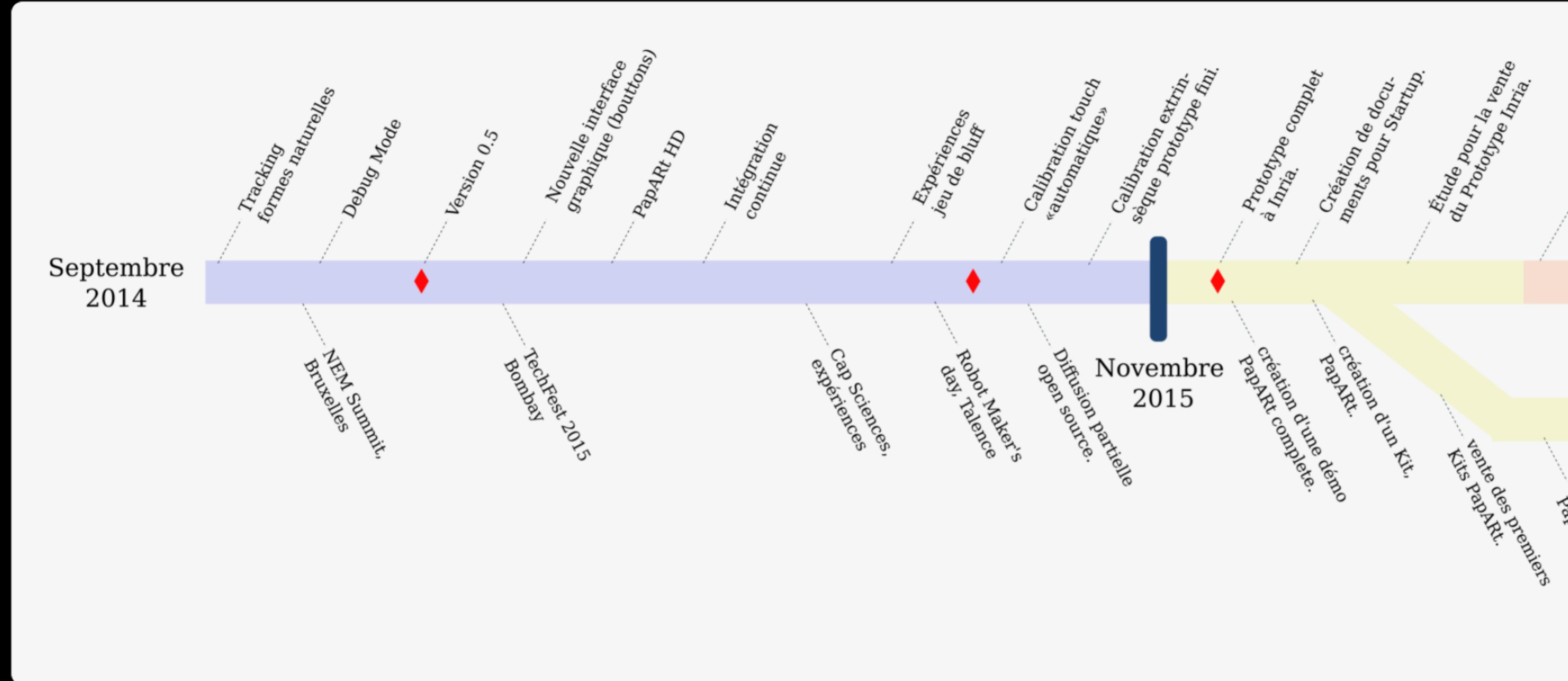
Étude sur le coût de développement des applications

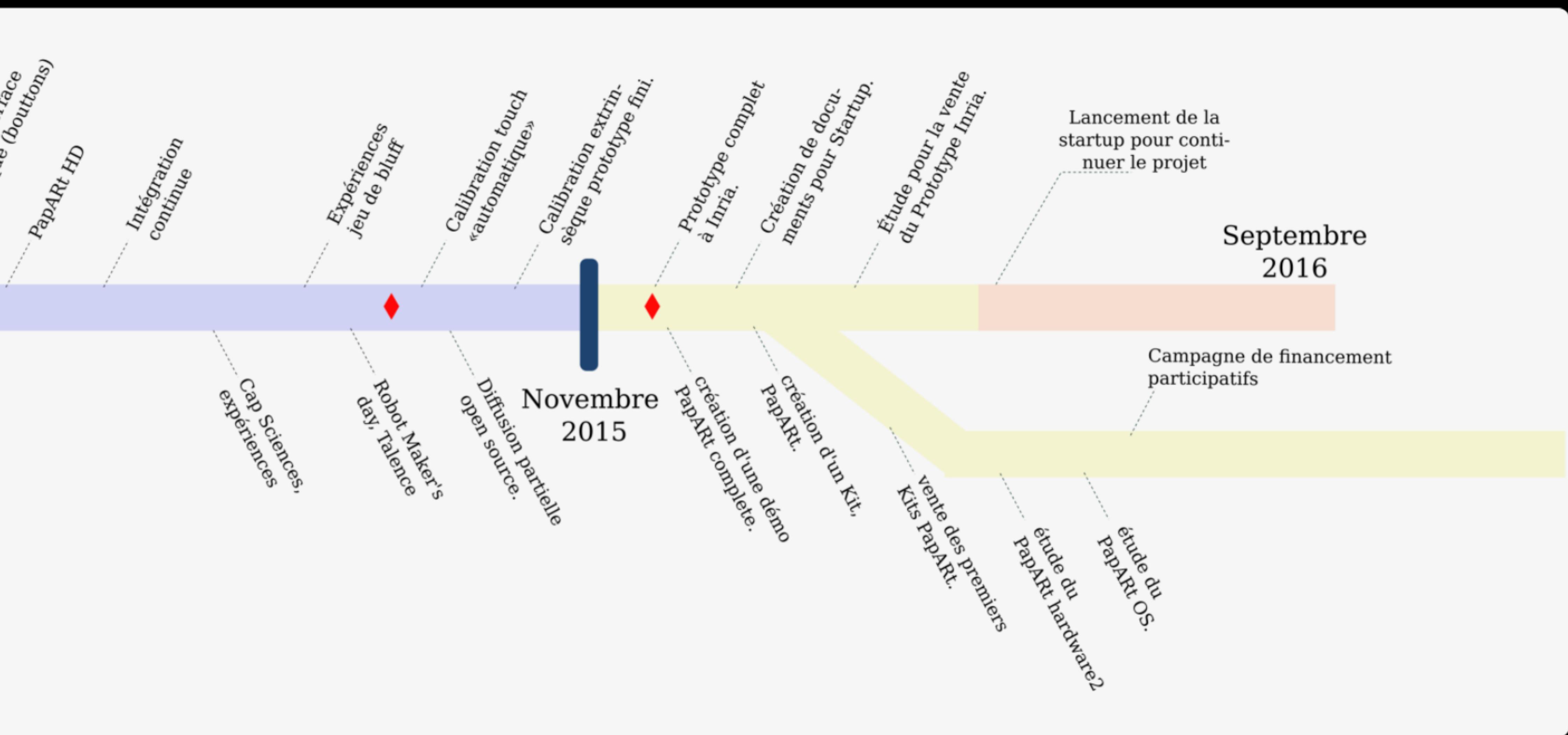
-> Trouver des partenaires pour la création des applications (jeu / dessin / présentations).









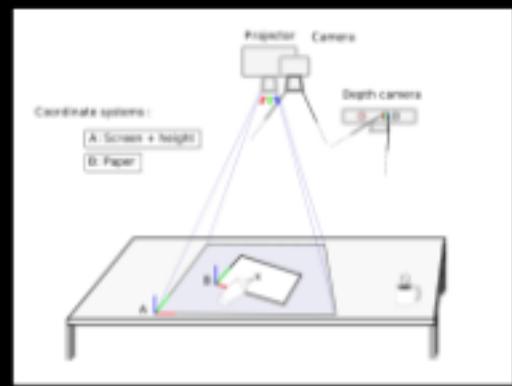


# Bilan

Avancement du projet à T + 14.

## Matériel

théorique



## Logiciel

fonctionnalités principales

### Logiciel de calibration extrinsèque

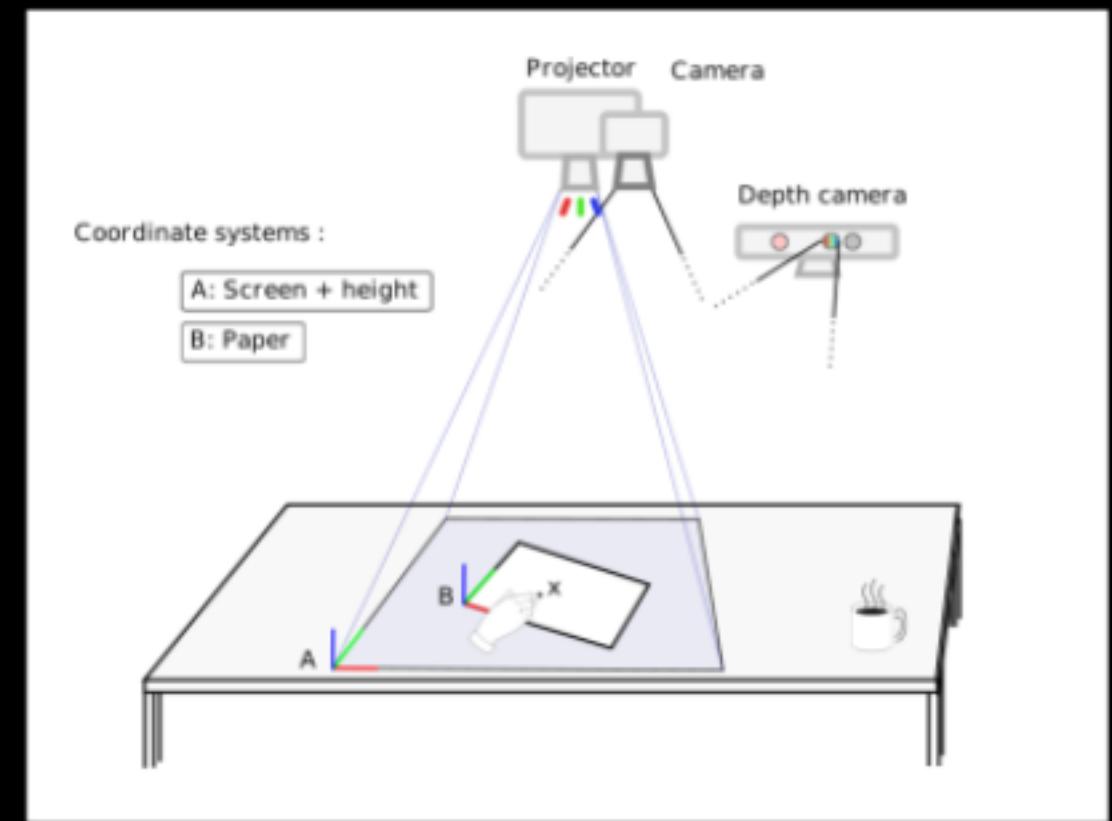
- Avec des marqueurs.
- Manuellement à la souris.
- Avec lumière structurée (Gray Code)  
-> Non intégré.

### Logiciel de calibration intrinsèque

- Totalement manuel à la souris.  
-> peu abouti car intérêt faible pour le projet.

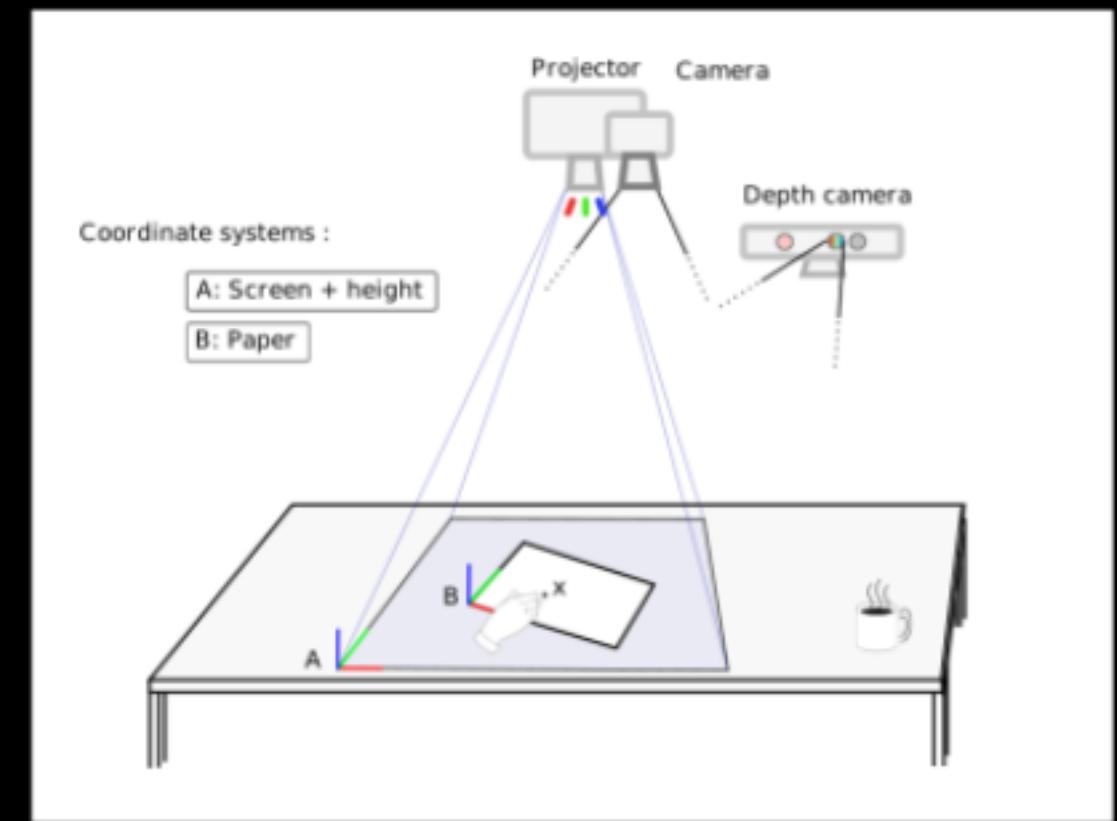
# Matériel

## théorique

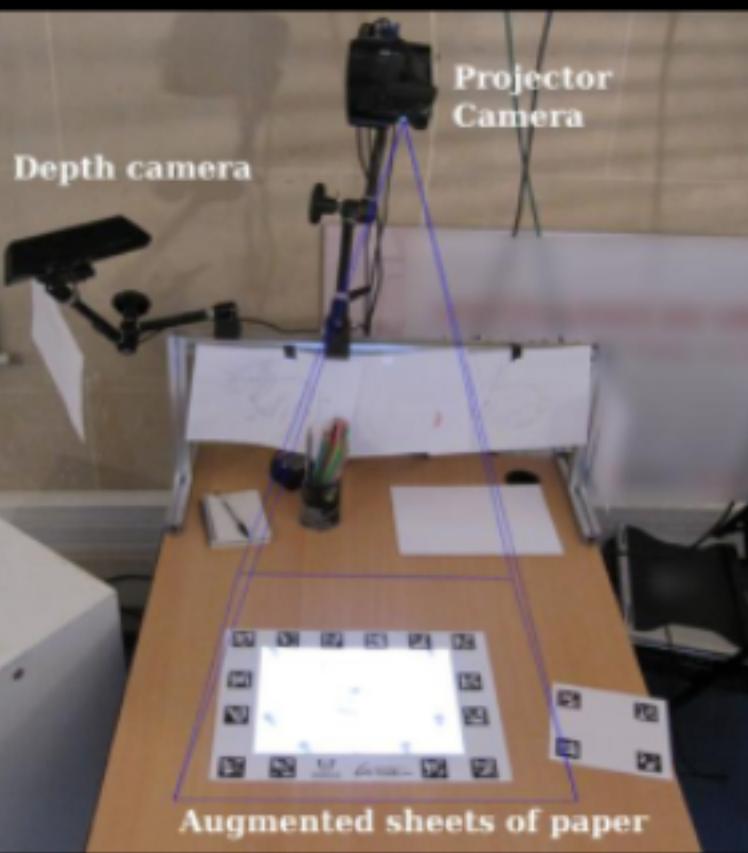


# Matériel

théorique

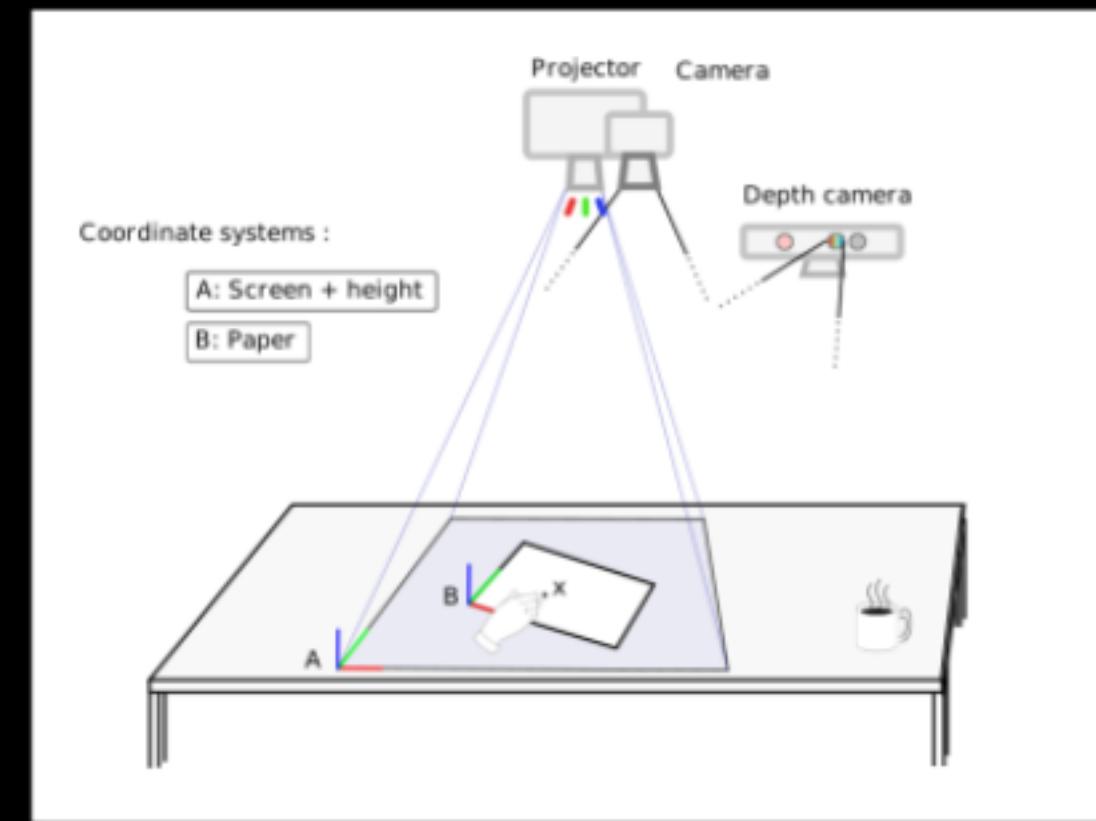


bricolé

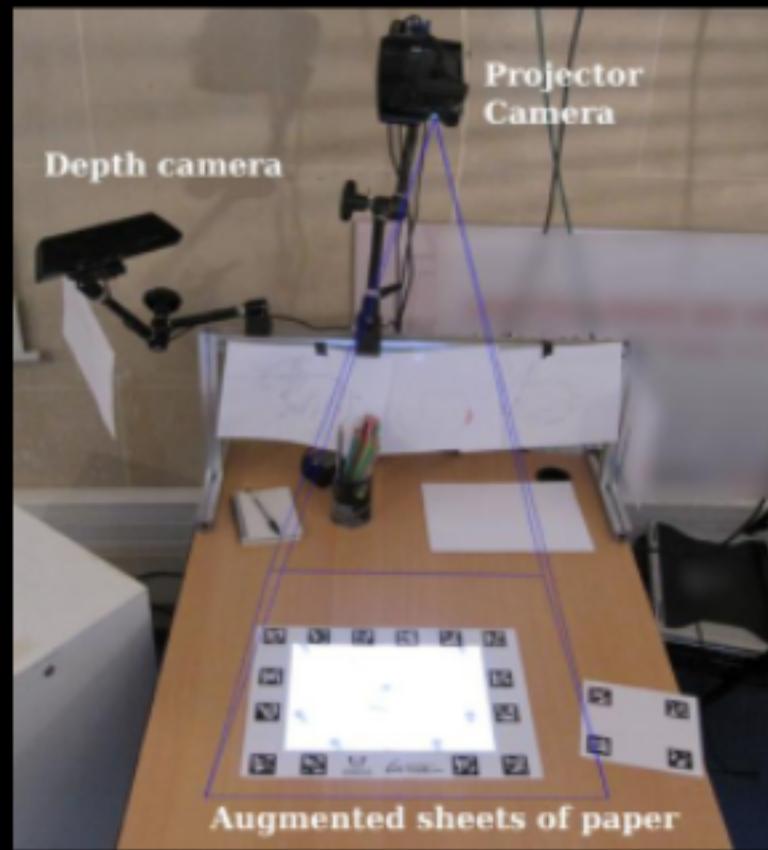


# Matériel

théorique



bricolé

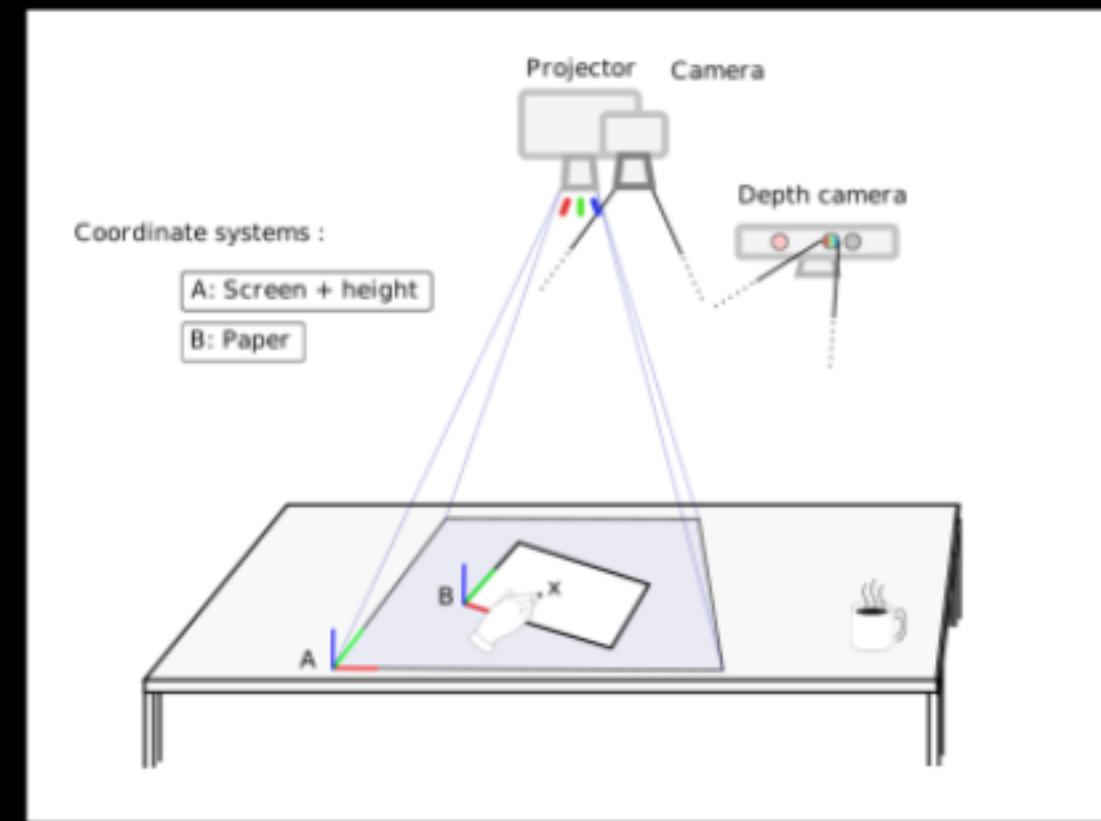


intégré

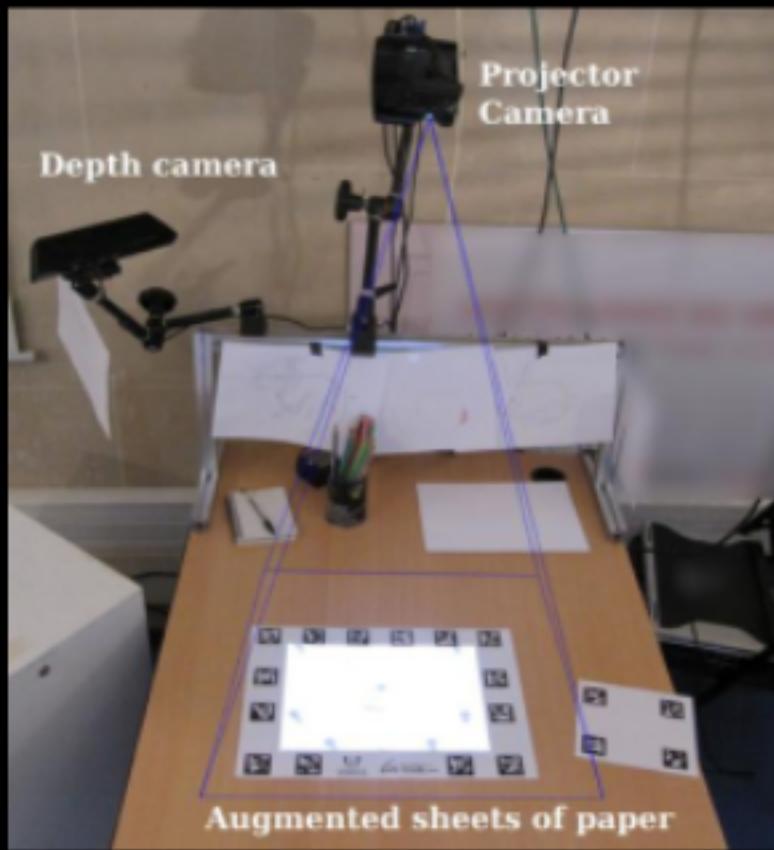


# Matériel

théorique



bricolé



intégré



designé et  
construit



# Logiciel

## fonctionnalités principales

### Logiciel de calibration extrinsèque

- Avec des marqueurs.
- Manuellement à la souris.
- Avec lumière structurée (Gray Code)
  - > Non intégré.

### Logiciel de calibration intrinsèque

- Totalement manuel à la souris.
- > peu abouti car intérêt faible pour le projet.

# Logiciel

## fonctionnalités principales

### Logiciel de calibration extrinsèque

- Avec des marqueurs.
- Manuellement à la souris.
- Avec lumière structurée (Gray Code)  
-> Non intégré.

### Surfaces planes interactives

- Sur une table, interaction avec les doigts.
- Sur un mur, interaction avec la main.
- Sur le sol, interaction avec les pieds.

### Logiciel de calibration intrinsèque

- Totalement manuel à la souris.
- > peu abouti car intérêt faible pour le projet.

# Logiciel

## fonctionnalités principales

### Logiciel de calibration extrinsèque

- Avec des marqueurs.
- Manuellement à la souris.
- Avec lumière structurée (Gray Code)  
-> Non intégré.

### Logiciel de calibration intrinsèque

- Totalement manuel à la souris.  
-> peu abouti car intérêt faible pour le projet.

### Surfaces planes interactives

- Sur une table, interaction avec les doigts.
- Sur un mur, interaction avec la main.
- Sur le sol, interaction avec les pieds.

### Application avec projection interactive

- Des feuilles de papier sont suivies et répondent au doigts des utilisateurs.
- Des objets peuvent interagir avec le système (style lego, pions, figurines etc...)
- Des objets peuvent être texturé avec la projection.

# Logiciel

avantages et points forts

Support d'entrées vidéo très large

Caméra couleur / niveaux de gris:

- OpenCV : fournit beaucoup d'entrées.
- Vidéo de Processing : une entrée par OS.
- FlyCapture : caméras Point Grey
- FFmpeg : Caméras IP, capture de bureau.
- Autres captures de bureau sous Windows et OSX.
- > à venir Android.

Caméra de profondeur:

- OpenKinect et OpenKinect 2

# Logiciel

avantages et points forts

## Support d'entrées vidéo très large

Caméra couleur / niveaux de gris:

- OpenCV : fournit beaucoup d'entrées.
- Vidéo de Processing : une entrée par OS.
- FlyCapture : caméras Point Grey
- FFmpeg : Caméras IP, capture de bureau.
- Autres captures de bureau sous Windows et OSX.
- > à venir Android.

Caméra de profondeur:

- OpenKinect et OpenKinect 2

## Tracking varié

OpenCV : ORB, FAST, SURF, AKAZE

-> à venir Agast, Brisk etc...

- ARToolkitPlus

-> à venir ChiliTags (EPFL)

- Touch avec Kinect, mains, objets.

-> extensible accès aux couleurs.

# Logiciel

avantages et points forts

## Support d'entrées vidéo très large

Caméra couleur / niveaux de gris:

- OpenCV : fournit beaucoup d'entrées.
- Vidéo de Processing : une entrée par OS.
- FlyCapture : caméras Point Grey
- FFmpeg : Caméras IP, capture de bureau.
- Autres captures de bureau sous Windows et OSX.
- > à venir Android.

Caméra de profondeur:

- OpenKinect et OpenKinect 2

## Tracking varié

OpenCV : ORB, FAST, SURF, AKAZE

-> à venir Agast, Brisk etc...

- ARToolkitPlus

-> à venir ChiliTags (EPFL)

- Touch avec Kinect, mains, objets.

-> extensible accès aux couleurs.

## En plus

- Compilation dans l'intégration continue.

- Marche en Live - USB sans installation.

# Logiciel

points en cours et à améliorer

logiciel de calibration

- Interface graphique à designer.
- Papart Panel à étudier (debug live)
- Manuel utilisateur (développeur).

# Logiciel

points en cours et à améliorer

## logiciel de calibration

- Interface graphique à designer.
- Papart Panel à étudier (debug live)
- Manuel utilisateur (développeur).

## API et exemples

- Description des exemples.
- Exemples avancés à créer.
- API Touch avancé à développer.
- API de reconnaissance d'objets à faire.

# Logiciel

points en cours et à améliorer

## logiciel de calibration

- Interface graphique à designer.
- Papart Panel à étudier (debug live)
- Manuel utilisateur (développeur).

## Démos

- Démos avancées à faire sans marqueurs carrés.
- Création/Design d'un feedback de base pour savoir ce qui est analysé (touch, objets etc...).

## API et exemples

- Description des exemples.
- Exemples avancés à créer.
- API Touch avancé à développer.
- API de reconnaissance d'objets à faire.

# Logiciel

points en cours et à améliorer

## logiciel de calibration

- Interface graphique à designer.
- Papart Panel à étudier (debug live)
- Manuel utilisateur (développeur).

## Démos

- Démos avancées à faire sans marqueurs carrés.
- Création/Design d'un feedback de base pour savoir ce qui est analysé (touch, objets etc...).

## API et exemples

- Description des exemples.
- Exemples avancés à créer.
- API Touch avancé à développer.
- API de reconnaissance d'objets à faire.

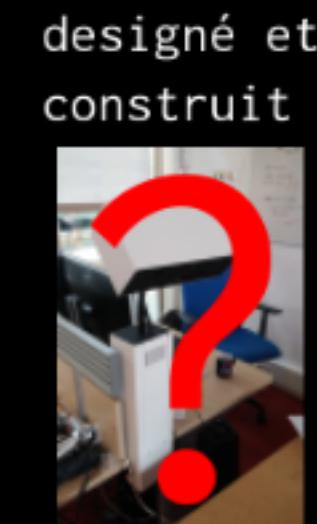
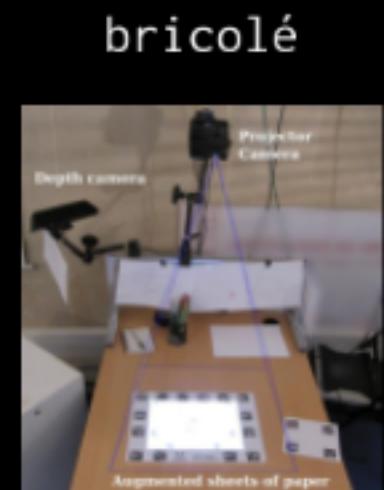
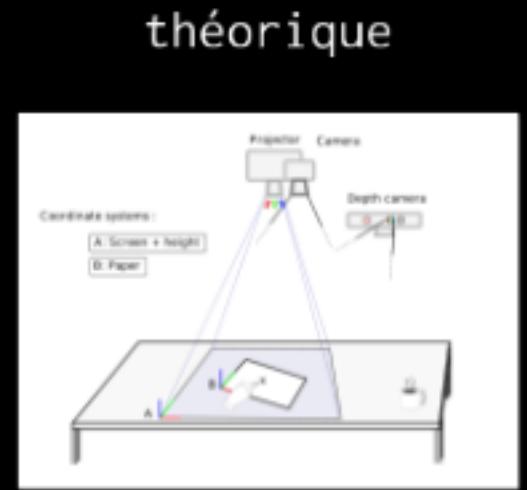
## Autre

- Preset java pour Libfreenect 2 à finir.
- API Boutons / sliders en SAR à designer.

# Bilan

Avancement du projet à T + 14.

## Matériel



## Logiciel

fonctionnalités principales

### Logiciel de calibration extrinsèque

- Avec des marqueurs.
- Manuellement à la souris.
- Avec lumière structurée (Gray Code)

-> Non intégré.

### Logiciel de calibration intrinsèque

- Totalement manuel à la souris.
- > peu abouti car intérêt faible pour le projet.

### Surfaces planes interactives

- Sur une table, interaction avec les doigts.
- Sur un mur, interaction avec la main.
- Sur le sol, interaction avec les pieds.

### Application avec projection interactive

- Des feuilles de papier sont suivies et répondent au doigt des utilisateurs.
- Des objets peuvent interagir avec le système (style lego, pions, figurines etc...)
- Des objets peuvent être texturé avec la projection.



# Projets connexes...

Hébergés chez github

Soby, logiciel de présentation

- Compatible avec PapARt.
- Design de slides + code.



# Projets connexes...

Hébergés chez github

Soby, logiciel de présentation

- Compatible avec PapARt.
- Design de slides + code.

PapAR, version Open Source

- PapARt sans projection, ni caméra de profondeur.
- Processing n'a pas de bonne / autre bibliothèque de RA.
- Applications compatibles PapARt.



# Projets connexes...

Hébergés chez github

## Soby, logiciel de présentation

- Compatible avec PapARt.
- Design de slides + code.

## PapAR, version Open Source

- PapARt sans projection, ni caméra de profondeur.
- Processing n'a pas de bonne / autre bibliothèque de RA.
- Applications compatibles PapARt.

## Skatolo, interface graphique

- Pour expérimenter les interfaces projetées.
- Compatible avec le logiciel de présentation.



# Projets connexes...

Hébergés chez github

## Soby, logiciel de présentation

- Compatible avec PapARt.
- Design de slides + code.

## Poppy en Ruby

- Pour faire des applications Poppy - PapARt.
- Discussions avec Flowers pour faire de l'enseignement avec la projection et robots.

## PapAR, version Open Source

- PapARt sans projection, ni caméra de profondeur.
- Processing n'a pas de bonne / autre bibliothèque de RA.
- Applications compatibles PapARt.

## Processing sur Maven

- Utilisé dans PapARt et d'autres bibliothèques.

## Skatolo, interface graphique

- Pour expérimenter les interfaces projetées.
- Compatible avec le logiciel de présentation.

## Libfreenect2 en Java via JavaCPP

- Pour avoir la Kinect 2 sous Windows , OSX et Linux avec une seule API.

# Objectifs - Discussion

## [Prototype]

Création d'un prototype «one button».

Stabilisation du SDK.

Création de deux applications: Dessin et jeu interactif.

## [Droit]

Étude sur la possibilité de vendre le SDK, dans sa version d'aujourd'hui.

## [Application]

Vendre la technologie au lieu de la donner / échanger contre de la main d'oeuvre.

## [Hardware]

Créer un matériel «maison» au lieu de Kinect + projecteur.

-> Trouver des partenaires pour l'électronique et l'optique pour chiffrer le coût.

## [Financement]

Candidater à des financements publics et faire une campagne de financement participatif.

## [Software]

Étude sur le coût de développement des applications

-> Trouver des partenaires pour la création des applications (jeu / dessin / présentations).