



Project Technique

La cinématique inverse

LES ANIMATIONS INTERAGISSENT AVEC LEURS ENVIRONNEMENT

I/ Présentation du sujet

- ▶ *La plus part des animations* sont *produites* en *tournant* des *joints* d'un squelette par des *valeurs prédéfinies*. La position et la rotation d'un joint dépendent de celles de son parent. Cette méthode s'appelle le **Forward Kinematics**.
- ▶ Il est aussi *possible* de *placer* des *joints* à une *position définit dans l'espace* et de *trouver* un *chemin valide* pour les orienter de sorte qu'elles puissent interagir et s'adapter à l'environnement dans lequel elles sont confrontées. Cette méthode s'appelle **l'Inverse Kinematics**.

II/ Objectifs du projet

- Réaliser différentes scènes *démontrant l'utilisation* de *l'IK* en utilisant le moteur *Unity puis recoder* mes systèmes *d'IK*.



III/ Différentes méthodes de résolution du problème de l'IK

- ▶ Analytic Two-Bone 2D.
- ▶ Cyclic Coordinate Descent 2D.
- ▶ Pseudo inverse.
- ▶ Dampned Least Squares.
- ▶ Réseau neuronal.
- ▶ Etc..

IV/1 Problématiques résolus et algorithmes utilisées

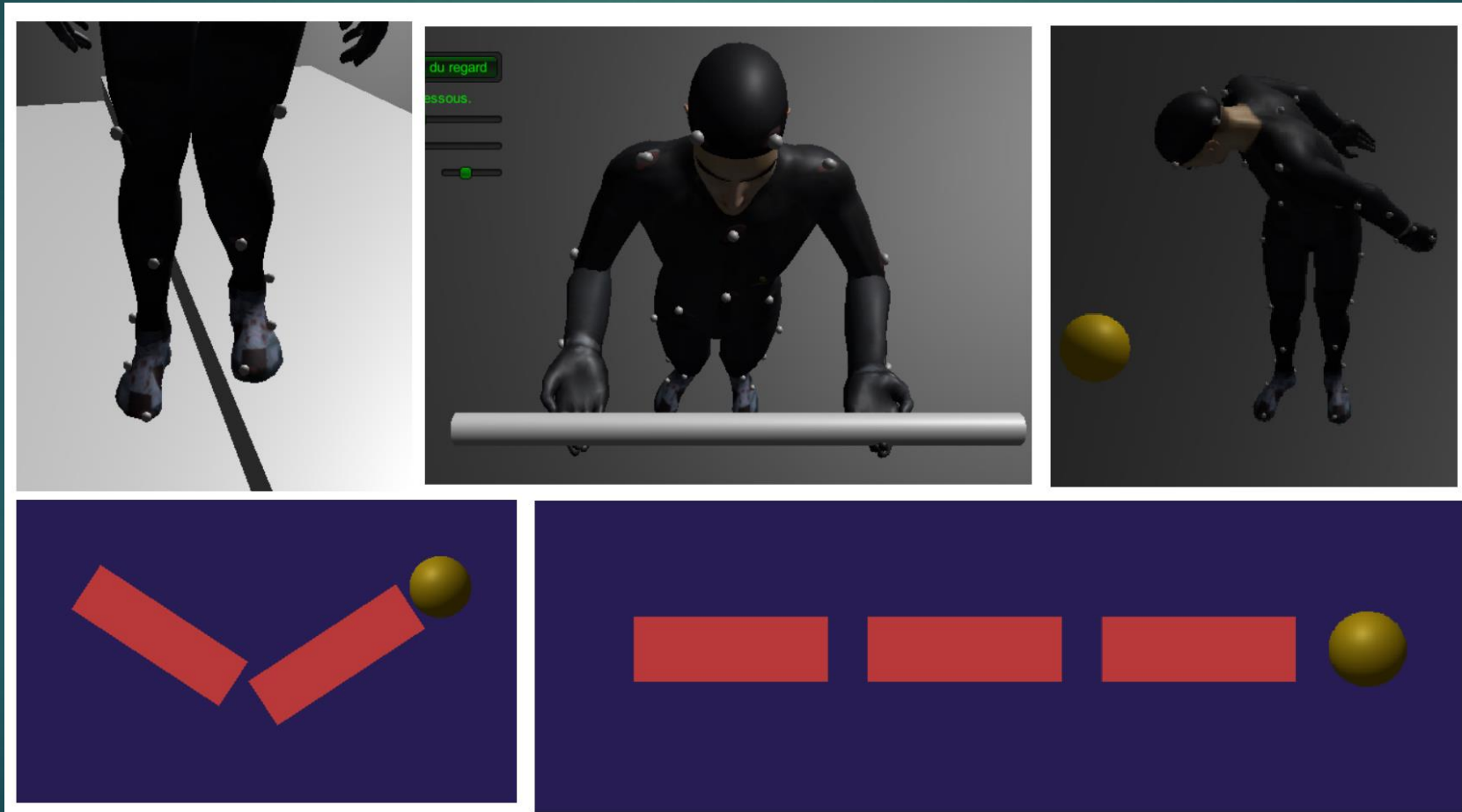
- ▶ Utilisation du système d'IK d'Unity :

- ▶ *Placement* des *pieds* d'un modèle en fonction d'un terrain.
- ▶ Placement des *mains* d'un modèle sur un objet (un bâton).
- ▶ Pointer le *corps*, les *yeux*, le *visage* d'un modèle sur une cible.

- ▶ Compréhension et implémentation de différents algorithmes :

- ▶ Analytic *Two-Bones* 2D.
- ▶ Cyclic *Cordinate* *D*escent 2D.
- ▶ Une *scène 3D* dans lequel on peut *placer 2 cibles de liens* à des positions et rotations prédéfinîtes.

IV/2 Problématiques résolues et algorithmes utilisées



V/ Les difficultés rencontrées

- ▶ Demande une bonne *compréhension* des *mathématiques*.
- ▶ Des fois les *algorithmes* présentées et trouvées sur des *articles* s'avérées *faux* et il fallait alors *réévaluer* le calcul et le résultat de ces *équations* afin d'arriver à un algorithme correct.
- ▶ **Revoir différents chapitres de mathématiques :**
 - ▶ La *trigonométrie*.
 - ▶ Le cercle trigonométrique.
 - ▶ Le théorème de *Pythagore*.
 - ▶ Le théorème *d'Al Kashi*.
 - ▶ La résolution *d'équations*.

VI/ Conclusion

- ▶ *L'IK* est un système mathématique qui peut s'avérer *lourd* selon les algorithmes car il doit bien *souvent* être *joué à chaque frame*.
- ▶ Par conséquent, il ne faut *pas l'utiliser sur* des *modèles trop présents* dans une application.
- ▶ *Faire* un système *d'IK avancé* réaliste qui gère un maximum de cas s'avère une tâche *complexe*, selon la qualité du rendu que l'on souhaite prétendre il faut *soit payer* un *plugin* *soit recoder le nôtre*.
- ▶ Néanmoins *plusieurs algorithmes 2D* peuvent *s'implémenter* assez *vite* grâce à la *multitude d'articles* qui traitent le sujet de façon découpé et itérative.
- ▶ Personnellement, j'ai trouvé ce *projet* très *intéressant* et m'a permis d'avoir une *idée* plus *globale* sur les *animations* et il m'a forcé à me *mettre* en *difficulté* en lisant et comprenant de la *documentation technique*.

Merci de m'avoir écouté

- Dorénavant je me remets en questions pour m'intégrer.

