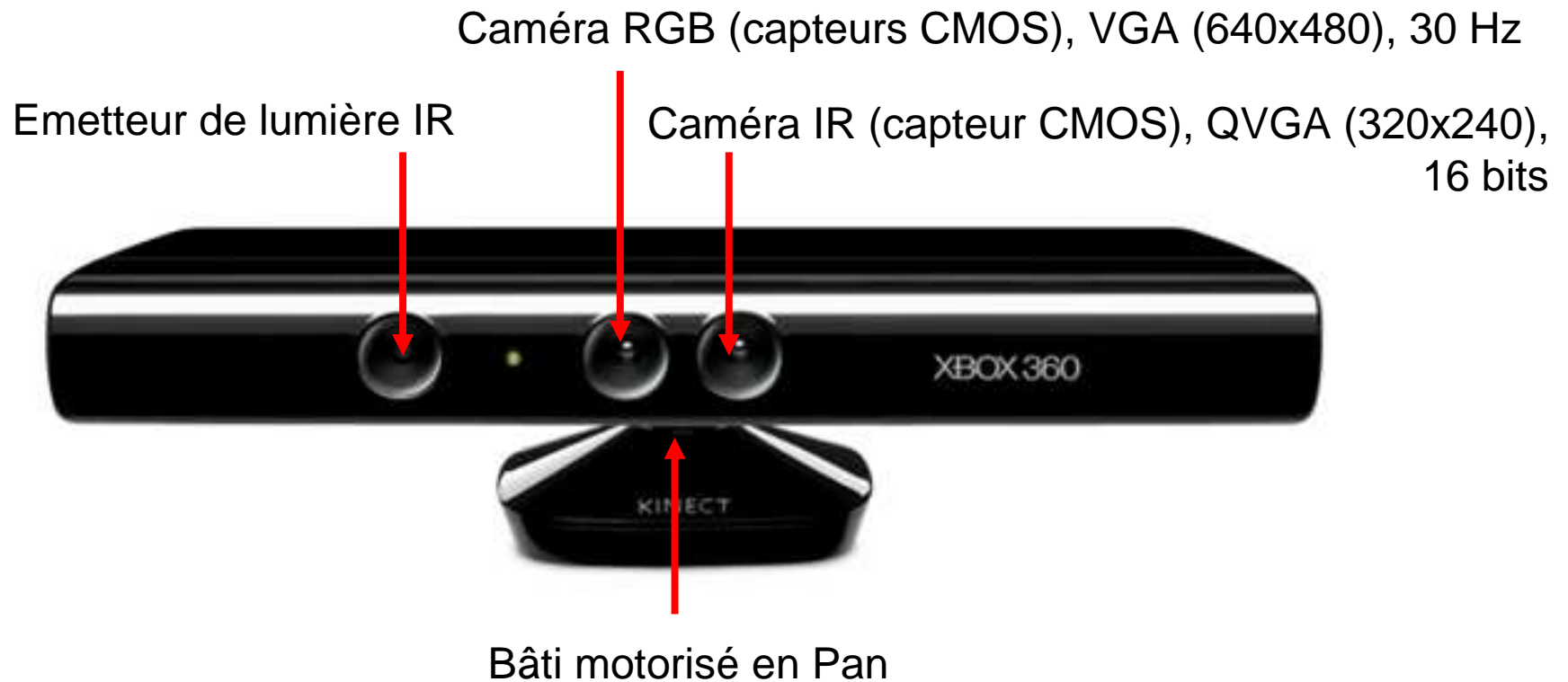


# La Kinect pour XBOX360



Puissance de la lumière IR reçue dépend de la distance à l'objet le plus proche (et d'autres facteurs)

Codage de la distance à l'objet le plus proche à partir d'une puce PS1080  
Intégrée dans la Kinect

# Accès aux données de la Kinect dans Unity

- Utilisation du logiciel FAAST comme producteur d'événements clavier/souris captables par Unity
  - Création d'une bibliothèque de gestes dans FAAST
    - Un geste <-> événement clavier/souris particulier
  - Récupération de l'événement dans Unity
    - `Input.GetKeyDown()`
    - `Input.GetMouseDown()`

# Accès aux données de la Kinect dans Unity

- Interface Microsoft Kinect SDK avec Unity
  - Package Kinect1.7UnityPackage
    - Récupération du flux RGB
    - Récupération du flux DEPTH
    - Récupération de la squelettisation
      - En position
        - » Classe C# *KinectPointController*
        - » 20 GameObjects en entrée = 20 parties du corps
        - » Relation avec un SkeletonWrapper (interface Kinect)
      - En rotation
        - » Classe C# *KinectModelControllerV2*
        - » 25 GameObjects en entrée = 25 parties du corps
        - » Relation avec un SkeletonWrapper (interface Kinect)

# Accès aux données de la Kinect dans Unity

- Interface Microsoft Kinect SDK avec Unity
  - Éléments de bas niveau
    - *KinectInterop.cs*
      - *poll{Color, Depth, Skeleton}*
      - *Get{Color, Depth, Skeleton}*
  - *Couche intermédiaire*
    - *KinectEmulator.cs*
      - *Données Replay d'un fichier*
    - *KinectSensor.cs*
      - *Données de la Kinect*
    - *DeviceOrEmulator.cs*
      - *Flag useEmulator=false par défaut*
  - *Couche de haut niveau*
    - *SkeletonWrapper.cs*
      - *État du ou des « joueurs » (tracké, non tracké, ...)*
      - *Positions 3D des parties du squelette*
      - *Orientations 4D des parties du squelette*

# Accès aux données de la Kinect dans Unity

- Communication avec le serveur VRPN de FAAST
  - Projet UIVA
    - <http://projects.ict.usc.edu/mxr/faast/>
    - Le squelette est décomposé en 24 capteurs virtuels
    - Trame par capteur = [position3D, rotation 4D, temps de capture]
- Serveur UIVA coté Unity
  - Récupère les trames de FAAST
  - Envoie les trames coté Unity C#

# Accès aux données de la Kinect dans Unity

- Client VRPN via UIVA Unity C#
  - *UIVA\_Client.cs*
    - GetKinectJointData()
  - Ne permet pas la récupération des flux RGB et DEPTH, ni de piloter l'axe PAN de la Kinect
  - Permet de se connecter à une Kinect distante
  - Permet d'interfacer d'autres périphériques (Wii par exemple) via un serveur VRPN dédié

# Squelettes Kinect sous Unity

- Création de GameObjects par éléments du squelette
- 2 possibilités:
  - Mettre tous les GameObjects au même niveau de hiérarchie dans le graphe de scène
    - Possible si accès aux positions absolues des éléments du squelette.
  - Créer une hiérarchie des GameObjects
    - Nécessaire si accès aux rotations d'un élément du squelette à l'autre ;
    - Possible si accès aux positions relatives des éléments du squelette.