

Kat'dventures

Généré par Doxygen 1.8.7

Lundi 2 Juin 2014 18 :24 :49

Chapitre 1

Index des classes

1.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

BoxMove	3
CatManipulability	3
CCD3d	4
EnablePush	5
MurDoigts	5
MurRoot	6

Chapitre 2

Documentation des classes

2.1 Référence de la classe BoxMove

Est dérivée de MonoBehaviour.

Attributs publics

- Transform [patteGauche](#)
patte gauche du chat
- Transform [patteDroite](#)
patte droite du chat
- float [e](#)
epsilon

2.1.1 Description détaillée

Script pour déplacer un objet lorsqu'il est poussé. A chaque frame, on teste si la distance entre les pattes du chat et l'objet à pousser est inférieure à un epsilon petit. Si oui on applique une force sur l'objet.

2.2 Référence de la classe CatManipulability

Fonctions membres publiques

- float [ftr](#) (Vector3 direction, Transform from)
Méthode de calcul pour trouver la meilleure prise.

2.2.1 Description détaillée

Classe de calcul pour la meilleure position pour le choix des prises

2.2.2 Documentation des fonctions membres

2.2.2.1 float ftr (Vector3 direction, Transform from)

Méthode de calcul pour trouver la meilleure prise.

Paramètres

<i>direction</i>	la direction voulue par le joueur (vers le haut, le bas,...)
<i>from</i>	la Transform sur laquelle le calcul va se faire

Renvoie

une valeur de comparaison exprimant si la prise est un bon choix.

2.3 Référence de la classe CCD3d

Est dérivée de MonoBehaviour.

Fonctions membres publiques

- void [CCDStep3D](#) (Transform joint, Transform effector, Transform [target](#))
performs one step of the CCD algorithm in 3d. For each joint in the kinematic chain, we compute the angle theta = (effector.joint.target). We then compute the axis u = [target-joint] ^ [effector-joint] and apply the rotation theta around this axis to drive the effector towards the target. The method is recursive and calls itself by going up the joint hierarchy.

Attributs publics

- Transform [target](#)
cible à atteindre
- Transform [armStart](#)
début du squelette
- InfoMembre[] [tabMembre](#)
tableau des membres du squelettes avec leurs angles de rotation minimum et maximum sur x,y,z

Fonctions membres privées

- float [ComputeAngle3D](#) (Vector3 a, Vector3 b)
method for computing an angle value between two 3d vectors using their dot product.
- void [verifAngles](#) (Transform joint)
allows to check if angles are ok according to values setted up in Inspector In the case that the angles are not correct, their values are imposed

2.3.1 Description détaillée

Script pour appliquer la cinématique inverse sur un squelette, prend en compte les butées articulaires.

2.3.2 Documentation des fonctions membres

2.3.2.1 void CCDStep3D (Transform joint, Transform effector, Transform target)

performs one step of the CCD algorithm in 3d. For each joint in the kinematic chain, we compute the angle theta = (effector.joint.target). We then compute the axis u = [target-joint] ^ [effector-joint] and apply the rotation theta around this axis to drive the effector towards the target. The method is recursive and calls itself by going up the joint hierarchy.

Paramètres

<i>joint</i>	the current joint that will be rotated towards the target.
<i>effector</i>	the end effector transform.
<i>target</i>	the transform containing the position we want the end-effector to reach.

2.3.2.2 float ComputeAngle3D (Vector3 *a*, Vector3 *b*) [private]

method for computing an angle value between two 3d vectors using their dot product.

Renvoie

the value of the angle existing between a and b.

2.3.2.3 void verifAngles (Transform *joint*) [private]

allows to check if angles are ok according to values setted up in Inspector In the case that the angles are not correct, their values are imposed

Paramètres

<i>joint</i>	the joint for which angles have to be checked
--------------	---

2.4 Référence de la classe EnablePush

Est dérivée de MonoBehaviour.

Attributs privés

- bool `isPushing` = false
vrai si le chat est en mode poussée, faux sinon

2.4.1 Description détaillée

Script pour passer le chat en mode poussée ou en mode marche en appuyant sur la touche P.

2.5 Référence de la classe MurDoigts

Est dérivée de MonoBehaviour.

Attributs publics

- Transform `DebutMembre`
Toutes les prises du mur.
- Transform `PriseDeDebut`
Racine du membre (en général épaule)
- int `epsilonDoigtPrise`
Prise à la première frame.
- int `epsilonEpaulePrise`
Correspond à la distance limite entre les doigts et la prise.
- Vector3 `directionDNT`
Correspond à la distance minimale entre l'épaule et la prise (TECCC)

Fonctions membres privées

- Transform [getPrise](#) ()
Méthode pour obtenir la meilleure prise ; application de l'algorithme pour TECCC.

Attributs privés

- [CatManipulability](#) *catm*
Direction voulue par le joueur.
- Transform [priseEnCours](#)
L'objet [CatManipulability](#) pour accéder aux méthodes de calcul.

2.5.1 Description détaillée

Script s'appliquant aux doigts du chat

2.5.2 Documentation des fonctions membres

2.5.2.1 Transform [getPrise](#) () [private]

Méthode pour obtenir la meilleure prise ; application de l'algorithme pour TECCC.

Renvoie

la meilleure prise selon TECCC

2.6 Référence de la classe MurRoot

Est dérivée de MonoBehaviour.

Attributs publics

- int [distanceMinimale](#)
le mur d'escalade
- Transform[] [tabDoigts](#)
la distance minimale pour que le chat s'agrippe au mur

Fonctions membres privées

- void [Start](#) ()
tableau des doigts du chat
- void [activateScript](#) (bool b)
Active ou désactive le script MurDoigt pour les 4 membres du chat.
- void [passageDir](#) (Vector3 dir)
Envoie la direction du mouvement voulu aux scripts MurDoigt (dans la variable directionDNT)

2.6.1 Description détaillée

Script appliqué à la racine du chat : gestion des directions voulues par le joueur sur le mur.

2.6.2 Documentation des fonctions membres

2.6.2.1 void activateScript (bool *b*) [private]

Active ou désactive le script MurDoigt pour les 4 membres du chat.

Paramètres

<i>b</i>	état d'activation des scripts MurDoigt des membres
----------	--

2.6.2.2 void passageDir (Vector3 *dir*) [private]

Envoie la direction du mouvement voulu aux scripts MurDoigt (dans la variable directionDNT)

Paramètres

<i>dir</i>	la direction voulue par le joueur
------------	-----------------------------------