

Τέταρτο μάθημα Ανάπτυξης Βιντεοπαιχνιδιών || Μαθηματικά εργαλεία

ΔΙΔΑΣΚΩΝ: ΚΩΝΣΤΑΝΤΙΝΟΣ ΤΣΙΧΛΑΣ

ΕΠΙΜΕΛΕΙΑ ΣΗΜΕΙΩΣΕΩΝ: ΠΟΤΟΣ ΒΑΣΙΛΕΙΟΣ, ΚΑΡΥΩΤΗ ΒΑΡΒΑΡΑ

Περιεχόμενα

Περιεχόμενα.....	1
Εισαγωγή	3
2D-σημεία και διανύσματα	4
Καρτεσιανό σύστημα	4
Επιπλέον συστήματα συντεταγμένων	6
Κυλινδρικό	6
Σφαιρικό	6
Επιλογή Συστήματος	7
Επιπλέον για τους χώρους συντεταγμένων	7
Χώρος Μοντέλου (Model/Object space).....	8
Χώρος Εικονικού Κόσμου (World Space)	8
Χώρος Προβολής (View space).....	8
Ιεραρχία Χώρων Συντεταγμένων	9
Μητρώο Αλλαγής Βάσης.....	9
Διανύσματα.....	11
Εσωτερικό γινόμενο	12
Μερικές Απλές Εφαρμογές	13
Ορατότητα.....	13
Εξωτερικό Γινόμενο	15
Εφαρμογές του Εξωτερικού Γινομένου	16
LERP (Linear intERPolation)	17
Μητρώα και Μετασχηματισμοί	18
Μετασχηματισμοί	18
Μητρώα Ατομικών Μετασχηματισμών.....	18
Στοιχειώδης Μεταφορά.....	19
Στοιχειώδης Περιστροφή	19
Κλιμάκωση.....	20
Βασική Ιδέα.....	20

Quaternions και 3D rotation	20
Πράξεις	22
Συζυγείς και Αντίστροφοι	22
Περιστροφή με Quaternions	22
Πολλαπλές Περιστροφές με Quaternion.....	22
Περιστροφική Παρεμβολή (LERP).....	23
Σφαιρικό LERP	23
LERP ή SLERP	24
Σύγκριση Αναπαραστάσεων Περιστροφής	25
Γωνίες Euler	25
3 × 3 Μητρώα	25
Άξονας/Γωνία	25
Quaternions	26
SQT (SRT) Μετασχηματισμοί	26
Χρήσιμα Γεωμετρικά Αντικείμενα	27
Σημείο – Κάθετος	27
Πλέγματα Τριγώνων	27
Axis Aligned Bounding Box.....	27
Oriented Bounding Box	27
Frusta	28
Rigging.....	29
Άσκηση	30
Λύση.....	30

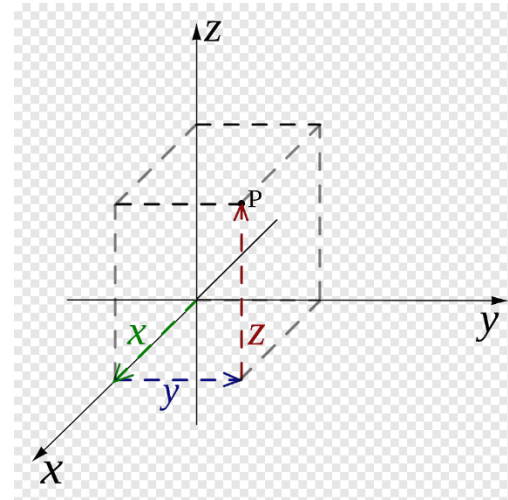
Μαθηματικά εργαλεία στα Βιντεοπαιχνίδια

Εισαγωγή

Τα μαθηματικά είναι απαραίτητα για την ανάπτυξη βιντεοπαιχνιδιών, προσφέροντας εργαλεία για τη δημιουργία ρεαλιστικών κόσμων και σύνθετων μηχανισμών. Βασικά στοιχεία για τον καθορισμό θέσεων και κινήσεων στο χώρο αποτελούν τα σημεία και τα διανύσματα. Η χρήση τους, σε συνδυασμό με το κατάλληλο σύστημα συντεταγμένων για κάθε περίπτωση, επιτρέπει τον ακριβή προσδιορισμό θέσεων σε δισδιάστατα και τρισδιάστατα περιβάλλοντα. Παράλληλα, τα μητρώα διευκολύνουν τους μετασχηματισμούς και την απεικόνιση των αντικειμένων, προσφέροντας σταθερότητα και ταχύτητα στις υπολογιστικές διαδικασίες. Ειδικά για τις περιστροφές, τα quaternions είναι απαραίτητα για την ομαλή απεικόνιση και τον έλεγχο, αποφεύγοντας προβλήματα όπως το gimbal lock που εμφανίζεται σε άλλες αναπαραστάσεις. Η σύγκριση αναπαραστάσεων μετασχηματισμών περιστροφής βοηθά επίσης στην επιλογή της καταλληλότερης τεχνικής για κάθε ανάγκη του παιχνιδιού. Με αυτά τα εργαλεία, τα βιντεοπαιχνίδια αποκτούν ρεαλισμό και δυναμική, προσφέροντας την καλύτερη δυνατή εμπειρία στους παίκτες.

2D-σημεία και διανύσματα

Στις περισσότερες περιπτώσεις, οι λειτουργίες στον 3D χώρο μπορούν να εφαρμοστούν και στον 2D χώρο, καθιστώντας χρήσιμο το 2D ως βάση για την κατανόηση και την ανάπτυξη τρισδιάστατων λειτουργιών. Ένα σημείο είναι μια θέση σε έναν n-διάστατο χώρο και συνήθως αναπαρίσταται σε ένα καρτεσιανό σύστημα συντεταγμένων. Σε αυτό το σύστημα, οι θέσεις ορίζονται μέσω δύο ή τριών αμοιβαία κάθετων αξόνων, όπως οι άξονες x , y , z στον τρισδιάστατο χώρο. Έτσι, ένα σημείο P στον χώρο αυτό μπορεί να οριστεί ως μια τριάδα συντεταγμένων (P_x, P_y, P_z) , που προσδιορίζει την ακριβή του θέση. Από την άλλη, ένα διάνυσμα p , με συνιστώσες p_x , p_y , p_z , διαφοροποιείται από το σημείο ως προς το ότι περιλαμβάνει και κατεύθυνση, όχι μόνο θέση.



Καρτεσιανό σύστημα

Το καρτεσιανό σύστημα συντεταγμένων, όπως αναφέραμε, είναι το πιο κοινό σύστημα για τον προσδιορισμό θέσεων και κινήσεων στον 2D και 3D χώρο, καθώς χρησιμοποιεί απλούς, κάθετους άξονες (συνήθως x , y , και z) για τον ορισμό των συντεταγμένων. Σε τρισδιάστατα περιβάλλοντα, οι άξονες αυτοί ορίζουν το μήκος, το ύψος και το βάθος, κάτι που επιτρέπει τον ακριβή υπολογισμό της θέσης, της κατεύθυνσης και της κίνησης των αντικειμένων.

Υπάρχουν δύο κύριες παραλλαγές του καρτεσιανού συστήματος: το **αριστερόστροφο** και το **δεξιόστροφο** σύστημα. Η διαφορά τους έγκειται στη φορά περιστροφής γύρω από τους άξονες. Στα αριστερόστροφα συστήματα, τα οποία προτιμούν οι προγραμματιστές γραφικών, το y δείχνει προς τα πάνω, το x προς τα δεξιά και το z δείχνει προς την οθόνη. Αυτή η διάταξη είναι συμβατή με πολλά γραφικά APIs και μηχανές παιχνιδιών και διευκολύνει τον σχεδιασμό του βάθους και της προοπτικής στον χώρο.



Η μετατροπή μεταξύ αριστερόστροφου και δεξιόστροφου συστήματος είναι σχετικά απλή και γίνεται με την αντιστροφή της κατεύθυνσης σε έναν από τους άξονες (π.χ., αναστροφή του z). Αυτό επιτρέπει την προσαρμογή του συστήματος ανάλογα με τις απαιτήσεις των γραφικών ή των φυσικών μηχανισμών που θέλει να επιτύχει ο προγραμματιστής.

Η βοήθεια των παραπάνω στο z -buffering (ή βάθος z) είναι σημαντική καθώς κύριος στόχος του z -buffering είναι να εξασφαλίσει ότι τα αντικείμενα που είναι πιο κοντά στον παρατηρητή καλύπτουν τα αντικείμενα που βρίσκονται πιο μακριά, επιτυγχάνοντας έτσι τον σωστό υπολογισμό της προοπτικής και της απόδοσης βάθους.

Σημαντικό είναι να αναφέρουμε την ύπαρξη επιπλέον συστημάτων συντεταγμένων, εκτός του καρτεσιανού, όπως το κυλινδρικό και το σφαιρικό. Σε κάποιες περιπτώσεις η χρήση κάποιου από αυτά μπορεί να είναι καλύτερη από τη χρήση του καρτεσιανού, βελτιώνοντας το παιχνίδι ως προς την απόδοση αλλά και ως προς το υπολογιστικό κομμάτι. Στη συνέχεια θα αναλύσουμε τα παραπάνω συστήματα και κάποιες περιπτώσεις χρήσης τους.

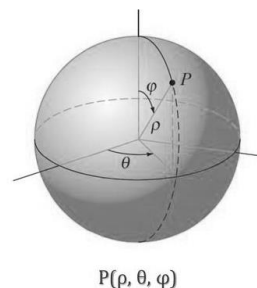
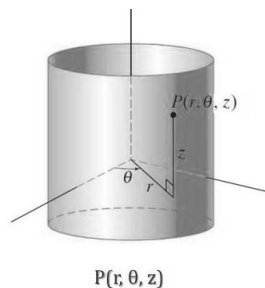
Επιπλέον συστήματα συντεταγμένων

Κυλινδρικό

Σε ορισμένες περιπτώσεις, αντί για το παραδοσιακό καρτεσιανό σύστημα συντεταγμένων, χρησιμοποιείται ένα σύστημα που βασίζεται σε έναν άξονα ύψους (h), έναν ακτινωτό άξονα (r) και μία γωνία εκτροπής (yaw , θ). Σε αυτό το σύστημα, κάθε σημείο μπορεί να αναπαρασταθεί ως μια τριάδα συντεταγμένων (p_h, p_r, p_θ), όπου p_h είναι το ύψος του σημείου, p_r η απόστασή του από τον κεντρικό άξονα και p_θ η γωνία που σχηματίζει με ένα σημείο αναφοράς. Αυτή η αναπαράσταση είναι ιδανική για κυλινδρικές δομές, καθώς επιτρέπει την εύκολη περιγραφή περιστροφικών κινήσεων και την τοποθέτηση αντικειμένων γύρω από έναν κεντρικό άξονα. Η χρήση του ύψους, της απόστασης και της γωνίας διευκολύνει τη διαχείριση θέσεων και την υλοποίηση κινήσεων, καθιστώντας το σύστημα αυτό κατάλληλο για σχεδιασμό σε παιχνίδια όπου υπάρχουν κυκλικές ή περιστροφικές τροχιές, όπως σε πλανήτες, πίστες ή οποιοδήποτε περιβάλλον απαιτεί ακριβείς κυλινδρικούς υπολογισμούς.

Σφαιρικό

Σε ένα σφαιρικό σύστημα συντεταγμένων, η θέση ενός σημείου καθορίζεται από την ακτινική απόσταση (*ακτινωτός άξονας radial* – r), τη γωνία βήματος (*pitch*, ϕ) και τη γωνία εκτροπής (yaw , θ). Στην αναπαράσταση αυτή, κάθε σημείο περιγράφεται ως μια τριάδα (p_r, p_ϕ, p_θ), όπου το p_r αντιπροσωπεύει την απόσταση του σημείου από το κέντρο, το p_ϕ τη γωνία του σημείου ως προς τον κατακόρυφο άξονα (γωνία βήματος), και το p_θ τη γωνία του σημείου ως προς τον οριζόντιο άξονα αναφοράς (γωνία εκτροπής). Αυτό το σύστημα είναι ιδιαίτερα χρήσιμο για την περιγραφή θέσεων σε σφαιρικά σχήματα ή για κινήσεις που ακολουθούν καμπύλες τροχιές, όπως στην περίπτωση ουρανίων σωμάτων ή αντικειμένων που περιστρέφονται σε σφαιρική επιφάνεια ή στην περίπτωση προσομοίωσης έκρηξης εντός του παιχνιδιού.

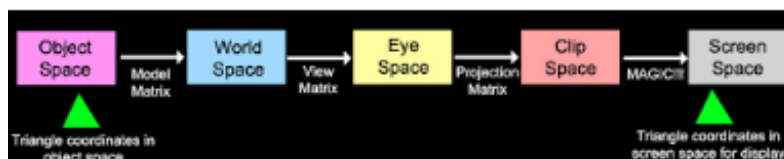


Επιλογή Συστήματος

Το καρτεσιανό σύστημα συντεταγμένων χρησιμοποιείται κυρίως στην ανάπτυξη βιντεοπαιχνιδιών, καθώς επιτρέπει την απλή και κατανοητή περιγραφή θέσεων και κινήσεων στον δισδιάστατο και τρισδιάστατο χώρο μέσω των αξόνων x , y , και z . Ωστόσο, σε ορισμένες περιπτώσεις, η χρήση άλλων συστημάτων είναι πιο αποδοτική και διευκολύνει τους υπολογισμούς που πιθανώς χρειαστεί να γίνουν. Για παράδειγμα, αντικείμενα που κινούνται σε κυκλικές ή περιστροφικές τροχιές γύρω από έναν κεντρικό χαρακτήρα (όπως αντικείμενα που στροβιλίζονται) είναι ευκολότερο να τα προσεγγίσουμε χρησιμοποιώντας κυλινδρικό σύστημα συντεταγμένων, που χρησιμοποιεί ύψος, απόσταση από τον κεντρικό άξονα και γωνία. Αντίστοιχα, φαινόμενα όπως οι εκρήξεις ή άλλες κυκλικές διασπορές είναι συχνά πιο απλές σε σφαιρικό σύστημα συντεταγμένων, όπου η θέση των κομματιών της έκρηξης μπορεί να οριστεί με ακτίνα και γωνίες, διευκολύνοντας τη φυσική διάδοση σε κάθε κατεύθυνση. Η επιλογή συστήματος συντεταγμένων, επομένως, εξαρτάται από την περίπτωση και μπορεί να συμβάλει στην απλοποίηση των υπολογισμών, βελτιώνοντας την απόδοση και τον ρεαλισμό της σκηνής.

Επιπλέον για τους χώρους συντεταγμένων

Ένα σημείο στο χώρο μπορεί να θεωρηθεί ως διάνυσμα σε σχέση με ένα συγκεκριμένο σύστημα αξόνων. Αυτό σημαίνει ότι η θέση του σημείου καθορίζεται από τις συντεταγμένες του στους άξονες του συστήματος, οι οποίες ορίζουν τις αποστάσεις του σημείου από κάθε άξονα. Οι άξονες αυτοί παρέχουν ένα πλαίσιο αναφοράς και δημιουργούν έναν χώρο συντεταγμένων, όπου οι θέσεις σημείων ή αντικειμένων μπορούν να περιγραφούν με ακρίβεια. Για παράδειγμα, σε ένα καρτεσιανό σύστημα τριών αξόνων (x , y , z), η θέση ενός σημείου καθορίζεται από τρεις συντεταγμένες, μία για κάθε άξονα. Αυτό το πλαίσιο αναφοράς είναι σημαντικό γιατί προσφέρει ένα σταθερό σημείο αναφοράς για τον υπολογισμό αποστάσεων, γωνιών, και κατευθύνσεων μεταξύ διαφόρων σημείων ή αντικειμένων. Ακολουθεί σύντομη ανάλυση των σημαντικότερων χώρων συντεταγμένων από αυτούς που παρουσιάζονται στο παρακάτω σχήμα.



Χώρος Μοντέλου (Model/Object space)

Ο Χώρος Μοντέλου (Model ή αλλιώς Object Space) είναι το σύστημα συντεταγμένων όπου καθορίζονται οι κορυφές ενός μοντέλου κατά τη φάση δημιουργίας του. Σε αυτό το πλαίσιο αναφοράς, κάθε κορυφή του μοντέλου έχει συγκεκριμένες συντεταγμένες, οι οποίες εκφράζονται σε σχέση με το κέντρο του αντικειμένου. Η αρχή των συντεταγμένων του χώρου μοντέλου βρίσκεται συνήθως στο κέντρο του ίδιου του αντικειμένου, διευκολύνοντας έτσι τον προσδιορισμό της θέσης και των μεγεθών του. Οι άξονες του χώρου μοντέλου αντιστοιχούν συνήθως σε κατευθύνσεις όπως "μπροστά" "αριστερά" και "επάνω" παρέχοντας ένα σαφές πλαίσιο προσανατολισμού. Αυτές οι κατευθύνσεις είναι ιδιαίτερα χρήσιμες για την τοποθέτηση και την κίνηση του μοντέλου μέσα στο εικονικό περιβάλλον, καθώς το καθένα μεταφράζεται σε έναν αντίστοιχο παγκόσμιο άξονα όταν το μοντέλο εντάσσεται στη σκηνή.

Ο χώρος μοντέλου είναι το σημείο εκκίνησης για όλες τις μεταγενέστερες μετατροπές ενός μοντέλου (όπως μεταφορά, κλίμακα, και περιστροφή), που εφαρμόζονται για να το τοποθετήσουν σωστά στον τρισδιάστατο κόσμο ενός παιχνιδιού ή μιας σκηνής.

Χώρος Εικονικού Κόσμου (World Space)

Ο Χώρος Εικονικού Κόσμου (World Space) είναι ένα σταθερό σύστημα συντεταγμένων όπου τοποθετούνται και ορίζονται όλα τα αντικείμενα, οι προσανατολισμοί και οι κλίμακες τους μέσα σε μια σκηνή ή σε έναν κόσμο παιχνιδιού. Ο χώρος αυτός λειτουργεί ως γενικό σημείο αναφοράς για όλα τα μοντέλα και τις οντότητες, επιτρέποντας τη συνδυασμένη αλληλεπίδρασή τους στο περιβάλλον του παιχνιδιού. Το κέντρο του συστήματος συνήθως τοποθετείται στο κέντρο της περιοχής παιχνιδιού, ώστε οι συντεταγμένες να επεκτείνονται θετικά και αρνητικά προς όλες τις κατευθύνσεις από αυτό το σημείο. Ο προσανατολισμός των αξόνων είναι αυθαίρετος αλλά καθορίζεται για να ταιριάζει με τις ανάγκες της εφαρμογής. Συνήθως, ο άξονας y ή ο άξονας z επιλέγεται ως ο κατακόρυφος άξονας ("επάνω"), που διευκολύνει τον διαχωρισμό των κατακόρυφων και οριζόντιων κινήσεων.

Μεταβιβάζοντας τα αντικείμενα από τον χώρο του μοντέλου στον χώρο του εικονικού κόσμου, μπορούμε να τα τοποθετούμε, να τα περιστρέφουμε και να τα κλιμακώνουμε με βάση το σταθερό αυτό σύστημα συντεταγμένων, εξασφαλίζοντας συνοχή στη σκηνή και ομαλή ενσωμάτωση κάθε αντικειμένου στον κόσμο του παιχνιδιού.

Χώρος Προβολής (View space)

Ο Χώρος Προβολής είναι ένα σύστημα συντεταγμένων που έχει ως αρχή τη θέση της κάμερας. Σε αυτόν τον χώρο, η κάμερα τοποθετείται στο κέντρο του συστήματος και προσανατολίζεται έτσι ώστε να κοιτάζει προς μια συγκεκριμένη κατεύθυνση, με αποτέλεσμα όλα τα αντικείμενα της σκηνής να περιγράφονται σε σχέση με την κάμερα. Η μετατροπή των

αντικειμένων από τον Χώρο Εικονικού Κόσμου (World Space) στον Χώρο Προβολής είναι απαραίτητη για να υπολογιστούν ποια μέρη της σκηνής είναι ορατά από την κάμερα και πώς θα αποδοθούν στην τελική εικόνα. Ο Χώρος Προβολής επιτρέπει την ευθυγράμμιση της σκηνής με τη "ματιά" της κάμερας, κάνοντας δυνατή τη δημιουργία προοπτικής και βάθους στην απεικόνιση των αντικειμένων. Αυτή η μετατροπή είναι θεμελιώδης για τη διαδικασία της απόδοσης γραφικών, καθώς ορίζει τη γωνία, την κλίμακα, και την τοποθέτηση των αντικειμένων όπως θα τα αντιληφθεί ο θεατής.

Ιεραρχία Χώρων Συντεταγμένων

Καθώς κάθε χώρος συντεταγμένων είναι σχετικός με κάποιον άλλο, δημιουργείται μια ιεραρχία, όπου κάθε χώρος αναφέρεται σε έναν "γονέα" για τον προσανατολισμό και τη θέση του. Στη ρίζα αυτής της ιεραρχίας βρίσκεται ο Χώρος του Εικονικού Κόσμου (World Space), ο οποίος είναι το σταθερό σημείο αναφοράς για όλα τα αντικείμενα της σκηνής. Όλα τα μοντέλα και οι κάμερες τοποθετούνται και κινούνται σε σχέση με αυτόν τον χώρο. Η κάμερα έχει μια θέση μέσα στον Χώρο του Εικονικού Κόσμου και έτσι ο Χώρος Προβολής (View/Camera Space) που ορίζεται από την κάμερα έχει τον Χώρο του Κόσμου ως γονέα. Κατ' αυτόν τον τρόπο, οι μετασχηματισμοί μπορούν να συνδεθούν, ώστε οι αλλαγές σε έναν χώρο (όπως η θέση της κάμερας) να επηρεάζουν σωστά τους εξαρτώμενους χώρους και να διατηρούν τη συνοχή της σκηνής.

Η μετατροπή ενός χώρου σε αυτόν του γονιού του πραγματοποιείται με την χρήση του μητρώου αλλαγής βάσης, το οποίο θα αναλύσουμε στη συνέχεια.

Μητρώο Αλλαγής Βάσης

Το μητρώο που μετασχηματίζει έναν χώρο σε αυτόν του γονιού του είναι το $M_{C \rightarrow P}$.

Με αυτό το μητρώο επιτυγχάνεται η αντιστοίχιση των θέσεων και προσανατολισμών από το σύστημα συντεταγμένων του παιδιού σε αυτό του γονιού. Με αυτό το μητρώο, κάθε θέση P_C στο χώρο του παιδιού μπορεί να μετασχηματιστεί στη θέση P_P στο χώρο του γονιού μέσω της εξίσωσης: $P_P = P_C M_{C \rightarrow P}$

Όπου,

$$M_{C \rightarrow P} = \begin{bmatrix} i_C & 0 \\ j_C & 0 \\ k_C & 0 \\ t_C & 1 \end{bmatrix} = \begin{bmatrix} i_{Cx} & i_{Cy} & i_{Cz} & 0 \\ j_{Cx} & j_{Cy} & j_{Cz} & 0 \\ k_{Cx} & k_{Cy} & k_{Cz} & 0 \\ t_{Cx} & t_{Cy} & t_{Cz} & 1 \end{bmatrix}$$

Και έχουμε ότι:

i_C είναι το μοναδιαίο διάνυσμα στον άξονα x του χώρου του παιδιού, εκφρασμένο στο σύστημα συντεταγμένων του γονιού.

j_c είναι το μοναδιαίο διάνυσμα στον άξονα y του χώρου του παιδιού, εκφρασμένο στο σύστημα συντεταγμένων του γονιού.

k_c είναι το μοναδιαίο διάνυσμα στον άξονα z του χώρου του παιδιού, εκφρασμένο στο σύστημα συντεταγμένων του γονιού.

t_c είναι το διάνυσμα μεταφοράς που τοποθετεί το σύστημα του παιδιού στο χώρο του γονιού.

Διανύσματα

Παρακάτω θα δούμε ορισμένες από τις βασικές πράξεις μεταξύ διανυσμάτων.

1. Ομοιόμορφη κλιμάκωση χωρίς αλλαγή κατεύθυνσης: $\mathbf{sa} = (sa_x, sa_y, sa_z)$

2. Ανομοιόμορφη κλιμάκωση: $\mathbf{s} \otimes \mathbf{a} = (s_x a_x, s_y a_y, s_z a_z)$

$$\mathbf{aS} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} = \begin{bmatrix} s_x a_x & s_y a_y & s_z a_z \end{bmatrix}$$

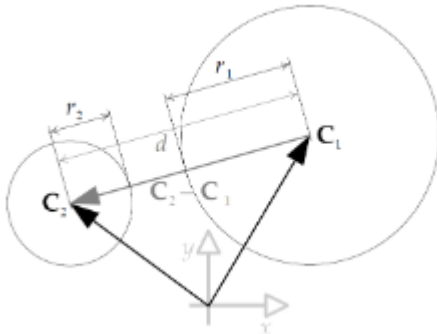
3. Πρόσθεση και αφαίρεση: $\mathbf{a} + \mathbf{b} = [(a_x + b_x), (a_y + b_y), (a_z + b_z)]$

$$\mathbf{a} - \mathbf{b} = [(a_x - b_x), (a_y - b_y), (a_z - b_z)]$$

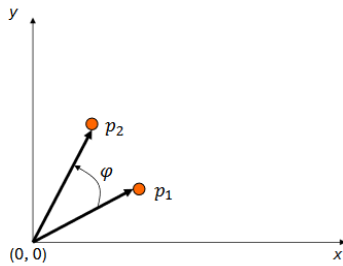
4. Υπολογισμός μέτρου (απόσταση): $|\mathbf{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$

5. Μετακίνηση αντικειμένων: $\mathbf{P}_2 = \mathbf{P}_1 + \mathbf{v}\Delta t$

6. Σύγκρουση αντικειμένων: Αν $d < r_1 + r_2$ τότε ενδεχομένως να υπάρχει σύγκρουση και χρειάζονται πιο περίπλοκοι υπολογισμοί για να καθορισθεί κάτι τέτοιο. Υπολογιστικά, αντί να κάνουμε την παραπάνω σύγκριση είναι πιο γρήγορο να συγκρίνουμε $d^2 < r_1^2 + r_2^2$ γλυτώνοντας των υπολογισμό τετραγωνικών ριζών.



Εσωτερικό γινόμενο

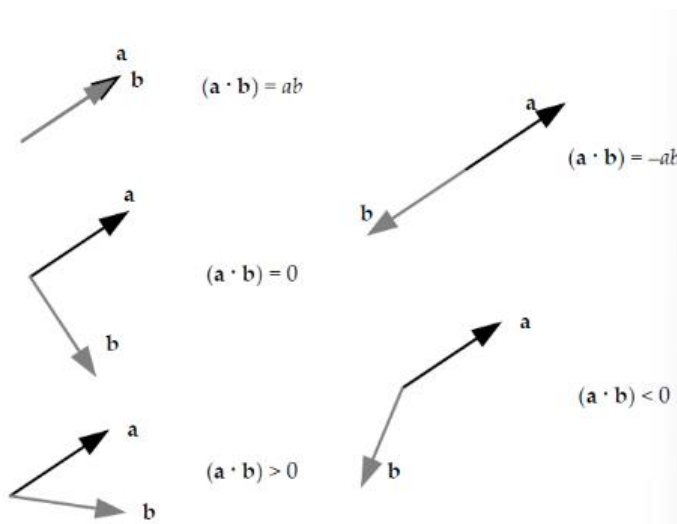


$$p_1 \cdot p_2 = x_1x_2 + y_1y_2 = p_2 \cdot p_1 = |p_1| |p_2| \cos \varphi$$

Με το εσωτερικό γινόμενο μπορούμε να ελέγξουμε τη σχετική θέση δύο διανυσμάτων.

Όπως φαίνεται στην εικόνα, μπορούμε ανάλογα με την τιμή του εσωτερικού γινομένου να καταλάβουμε τότε:

- 2 διανύσματα είναι συνευθειακά και τη γωνία έχουν μεταξύ τους (0° ή 180°)
- 2 διανύσματα είναι κάθετα
- 2 διανύσματα είναι αντίστροφα
- 2 διανύσματα σχηματίζουν οξεία ή αμβλεία γωνία

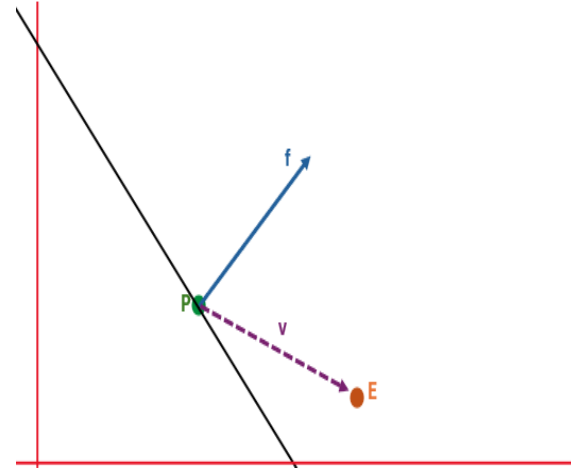


Μερικές Απλές Εφαρμογές

Ορατότητα

Για να διαπιστώσουμε αν ο **παίκτης P** μπορεί να δει έναν εχθρό E, υπολογίζουμε τη σχετική θέση του E σε σχέση με τον P μέσω του **διανύσματος v**, το οποίο δίνεται από την εξίσωση: $v = E - P$

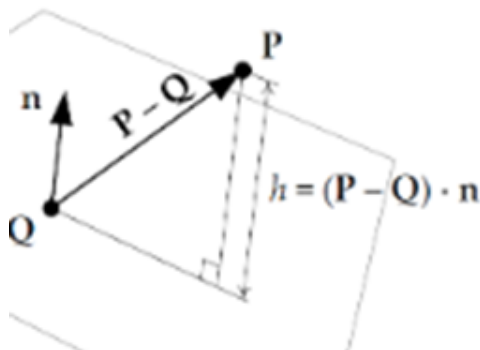
Αυτό το διάνυσμα v δείχνει την κατεύθυνση από τον παίκτη προς τον εχθρό. Αν ορίσουμε το **διάνυσμα f** ως το διάνυσμα που καθορίζει την κατεύθυνση που κοιτάζει ο παίκτης, μπορούμε να χρησιμοποιήσουμε το εσωτερικό γινόμενο των δύο διανυσμάτων για να κατανοήσουμε τη σχέση τους. Ειδικότερα, αν το εσωτερικό γινόμενο $v \cdot f$ είναι αρνητικό, αυτό σημαίνει ότι ο **εχθρός E** βρίσκεται πίσω από τον παίκτη P. Εάν είναι θετικό τότε ο εχθρός E είναι ορατός από τον P.



Για να υπολογίσουμε την απόσταση ενός σημείου P από ένα συγκεκριμένο επίπεδο που ορίζεται από ένα σημείο Q και ένα κάθετο μοναδιαίο διάνυσμα n προς το επίπεδο, μπορούμε πάλι να χρησιμοποιήσουμε το εσωτερικό γινόμενο. Η απόσταση ή το ύψος h του σημείου P πάνω από το επίπεδο υπολογίζεται μέσω της προβολής του διανύσματος v πάνω στο διάνυσμα n. Ο υπολογισμός αυτός μπορεί να γίνει ως εξής:

$$h = (\overrightarrow{P - Q}) \cdot \vec{n}$$

Αν το ύψος h είναι θετικό, τότε το σημείο P βρίσκεται πάνω από το επίπεδο, ενώ αν είναι αρνητικό είναι κάτω από το επίπεδο (όπου τα «πάνω» και «κάτω» ορίζονται σε σχέση με την κατεύθυνση του διανύσματος $\vec{n}$). Με αυτόν τον τρόπο μπορούμε επίσης να κάνουμε και μία αρχική δοκιμή ορατότητας του P από το Q όταν η κατεύθυνση προς την οποία κοιτά δίνεται από το διάνυσμα $\vec{n}$.



Εξωτερικό Γινόμενο

Ουσιαστικά, δίνει ένα επιπλέον διάνυσμα, το οποίο είναι κάθετο στο επίπεδο που ορίζουν τα διανύσματα που πολλαπλασιάζονται. Μπορεί να χρησιμοποιηθεί και για υπολογισμό επιφάνειας και δίνεται από τον εξής τύπο:

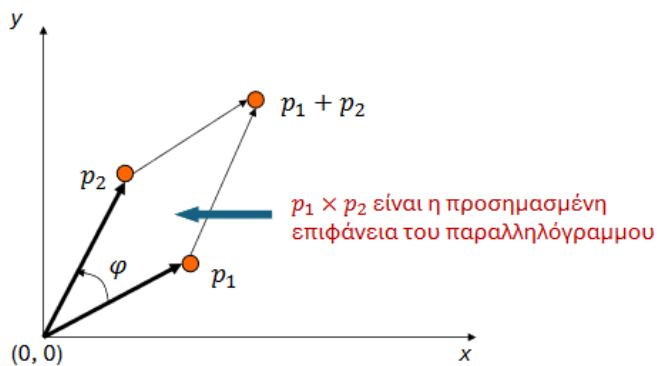
$$a \times b = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix}$$

Στις δύο διαστάσεις ισχύει $a_z = b_z = 0$, άρα ο παραπάνω τύπος διαμορφώνεται ως εξής:

$$a \times b = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_x & a_y & 0 \\ b_x & b_y & 0 \end{vmatrix} = \hat{k} \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix}$$

Επίσης, έχουμε $p_1 \times p_2 = x_1 y_2 - x_2 y_1 = -p_2 \times p_1 = |p_1| |p_2| \sin \varphi$

Όπου τα p_1, p_2 είναι συνευθειακά αν και μόνο αν $p_1 \times p_2 = 0$.



Εφαρμογές του Εξωτερικού Γινομένου

Αν έχουμε τρία σημεία P_1, P_2, P_3 στο χώρο, μπορούμε να υπολογίσουμε ένα κανονικό διάνυσμα (normal vector) που είναι κάθετο στο επίπεδο που σχηματίζεται από αυτά τα σημεία. Το κάθετο μοναδιαίο διάνυσμα υπολογίζεται ως εξής:

$$n = \text{normalize}((P_2 - P_1) \times (P_3 - P_1))$$

όπου η συνάρτηση $\text{normalize}(P_x, P_y, P_z)$ θα επιστρέψει ένα διάνυσμα p τέτοιο ώστε να έχει ίδια κατεύθυνση με το διάνυσμα $p = (P_x, P_y, P_z)$ αλλά με μέτρο $d = \sqrt{P_x'^2 + P_y'^2 + P_z'^2} = 1$.

Επίσης, μπορούμε να υπολογίσουμε την ροπή, η οποία θα βοηθήσει στην περιγραφή της στροφικής κίνησης ενός αντικειμένου. Δίνεται δύναμη F και διάνυσμα r που ορίζει την απόσταση από το κέντρο μάζας του αντικειμένου, άρα η ροπή N υπολογίζεται ως εξής:

$$N = r \times F$$

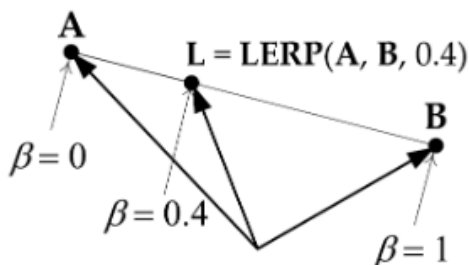
LERP (Linear intERPolation)

Η τεχνική της γραμμικής παρεμβολής (LERP - Linear intERPolation) είναι ένα σημαντικό εργαλείο στα γραφικά υπολογιστών και την ανάπτυξη παιχνιδιών. Χρησιμοποιείται κυρίως για την ομαλή κίνηση αντικειμένων, ή οποιαδήποτε άλλη εφαρμογή που απαιτεί σταδιακή αλλαγή από μία κατάσταση σε μια άλλη. Η γραμμική παρεμβολή μεταξύ δύο σημείων A και B ορίζεται με τον εξής τύπο:

$$L = \text{LERP}(A, B, \beta) = (1-\beta)A + \beta B = [(1-\beta)A_x + \beta B_x, (1-\beta)A_y + \beta B_y, (1-\beta)A_z + \beta B_z]$$

Όπου β είναι μια παράμετρος που κυμαίνεται από 0 έως 1:

- Όταν $\beta=0$, το αποτέλεσμα L είναι το σημείο A.
- Όταν $\beta=1$, το αποτέλεσμα είναι το σημείο B.
- Για τιμές $0 < \beta < 1$, το L βρίσκεται σε κάποια ενδιάμεση θέση μεταξύ των δύο σημείων.



Στα παιχνίδια, συχνά χρειάζεται να βρούμε ένα διάνυσμα που βρίσκεται στη μέση μεταξύ δύο διανυσμάτων. Για παράδειγμα, εάν θέλουμε να κινούμε ομαλά ένα αντικείμενο από το σημείο A στο σημείο B κατά τη διάρκεια δύο δευτερολέπτων με 30 fps, θα πρέπει να υπολογίσουμε 59 ενδιάμεσες θέσεις μεταξύ A και B. Άρα, ανάμεσα στα σημεία A,B θα έχουμε 59 σημεία για να έχουμε 60 θέσεις και με την μετακίνηση από σημείο σε σημείο ανά frame θα δημιουργείται η ψευδαίσθηση της κίνησης.

Μητρώα και Μετασχηματισμοί

Μετασχηματισμοί

Ένα μητρώο 2×2 μπορεί να χρησιμοποιηθεί για να αναπαραστήσει περιστροφές σε 2 διαστάσεις, με τον εξής τύπο:

$$\begin{bmatrix} r'_x & r'_y \end{bmatrix} = \begin{bmatrix} r_x & r_y \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix}$$

Με αντίστοιχο τρόπο ένα μητρώο 3×3 μπορεί να χρησιμοποιηθεί για αναπαράσταση περιστροφών σε 3 διαστάσεις, με τον εξής τύπο:

$$\begin{bmatrix} r'_x & r'_y & r'_z \end{bmatrix} = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Τα μητρώα 3×3 χρησιμοποιούνται για περιστροφή και για κλιμάκωση μέσω της πράξης πολλαπλασιασμού μητρώων αλλά όχι για αναπαράσταση μετασχηματισμού μεταφοράς που απαιτεί την πράξη του αθροίσματος διανυσμάτων. Για να ομογενοποιήσουμε τους μετασχηματισμούς χρησιμοποιούμε ομογενείς συντεταγμένες που αναπαρίστανται από μητρώα διάστασης 4×4 , με τον εξής τύπο για τη μεταφορά:

$$r + t = \begin{bmatrix} r_x & r_y & r_z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} = \begin{bmatrix} r_x+t_x & r_y+t_y & r_z+t_z & 1 \end{bmatrix}$$

Μητρώα Ατομικών Μετασχηματισμών

Ένα 4×4 μητρώο μετασχηματισμού διαχωρίζεται σε 4 υπομητρώα ως εξής:

$$M = \begin{bmatrix} U_{3 \times 3} & 0_{3 \times 1} \\ t_{1 \times 3} & 1 \end{bmatrix}$$

Όπου,

U : αναπαριστά περιστροφή ή/και κλιμάκωση

t : αναπαριστά μεταφορά

$$0 = [0 \ 0 \ 0]^T$$

Βαθμωτό μέγεθος 1

Όταν πολλαπλασιάζουμε ένα σημείο με ένα μητρώο μετασχηματισμού τότε έχουμε

$$[r'_{1x3} \ 1] = [r_{1x3} \ 1] \begin{bmatrix} U_{3x3} & 0_{3x1} \\ t_{1x3} & 1 \end{bmatrix} = [rU + T \ 1]$$

Στοιχειώδης Μεταφορά

$$r + t = [r_x \ r_y \ r_z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} = [r \ 1] \begin{bmatrix} I & 0 \\ t & 1 \end{bmatrix} = [r + t \ 1]$$

Και το αντίστροφο το υπολογίζουμε ομοίως αλλά με αρνητικό t .

Στοιχειώδης Περιστροφή

$$\text{rotate}_x(r, \varphi) = [r_x \ r_y \ r_z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

και με αντίστοιχο τρόπο έχουμε και τα εξής:

$$\text{rotate}_y(r, \theta) = [r_x \ r_y \ r_z \ 1] \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{rotate}_z(r, \gamma) = [r_x \ r_y \ r_z \ 1] \begin{bmatrix} \cos \gamma & \sin \gamma & 0 & 0 \\ -\sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Έχουμε ότι στο μητρώο περιστροφής το 1 εμφανίζεται στον άξονα περιστροφής, δηλαδή την κατεύθυνση γύρω από την οποία πραγματοποιείται η περιστροφή, ενώ τα ημίτονα και συνημίτονα τοποθετούνται στις θέσεις των άλλων αξόνων. Αυτό αποτρέπει τη μετακίνηση των συντεταγμένων κατά μήκος του άξονα περιστροφής και επιτρέπει την ομαλή περιστροφή γύρω από τον συγκεκριμένο άξονα.

Οι θετικές περιστροφές ακολουθούν έναν συγκεκριμένο προσανατολισμό στο χώρο:

- Από τον x προς τον y όταν περιστρέφουμε γύρω από τον z ,
- Από τον y προς τον z όταν περιστρέφουμε γύρω από τον x ,
- Από τον z προς τον x όταν περιστρέφουμε γύρω από τον y .

Αυτό το σύστημα συνδέεται με τον ορισμό των θετικών γωνιών στο τρισδιάστατο χώρο και διευκολύνει τους προγραμματιστές να ορίσουν συγκεκριμένες κατευθύνσεις περιστροφής.

Το αντίστροφο μιας στοιχειώδους περιστροφής είναι το ανάστροφο μητρώο της, το οποίο ουσιαστικά αναιρεί την αρχική περιστροφή. Αν έχουμε περιστρέψει ένα αντικείμενο γύρω

από έναν άξονα κατά μια γωνία θ , η αντίστροφη περιστροφή γίνεται με τη γωνία $-\theta$, και αυτό μπορεί να αναπαρασταθεί υπολογιστικά μέσω του ανάστροφου μητρώου περιστροφής.

Κλιμάκωση

$$rS = \begin{bmatrix} r_x & r_y & r_z & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r & 1 \end{bmatrix} \begin{bmatrix} S_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} rS_{3 \times 3} & 1 \end{bmatrix}$$

Για την αντιστροφή της κλιμάκωσης χρησιμοποιούμε τα αντίστροφα των συντελεστών κλιμάκωσης, δηλαδή αν οι συντελεστές κλιμάκωσης $S_{x,y,z}$ έχουν χρησιμοποιηθεί για να κλιμακωθεί ένα αντικείμενο στους άξονες x, y, z , η αντίστροφη κλιμάκωση μπορεί να επιτευχθεί χρησιμοποιώντας τους συντελεστές $1/S_{x,y,z}$. Με αυτόν τον τρόπο αναστρέφεται η παραμόρφωση, αν το αντικείμενο είχε ήδη κλιμακωθεί.

Αν όλοι οι συντελεστές κλιμάκωσης είναι ίδιοι, τότε έχουμε ομοιόμορφη κλιμάκωση, δηλαδή αυξάνεται με τον ίδιο τρόπο το μέγεθος του αντικειμένου σε κάθε άξονα, διατηρώντας τις αναλογίες του.

Βασική Ιδέα Εφαρμογής

Το πλέγμα (mesh) ενός αντικειμένου αποτελείται από ένα σύνολο κορυφών που ορίζουν το σχήμα και τη μορφή του. Για να εφαρμόσουμε οποιονδήποτε μετασχηματισμό στο πλέγμα, χρειάζεται να μετασχηματίσουμε κάθε θέση κορυφής μεμονωμένα, κάτι που επιτυγχάνεται μέσω μαθηματικών υπολογισμών με τη χρήση μητρώων. Κάθε απλός μετασχηματισμός, όπως περιστροφή, κλίμακα και μεταφορά, μπορεί να εκτελεστεί με τον πολλαπλασιασμό της θέσης κάθε κορυφής με έναν πίνακα μετασχηματισμού.

Quaternions και 3D rotation

Τα quaternions $q = [q_x \ q_y \ q_z \ q_w]$ είναι ένα ισχυρό εργαλείο για την αναπαράσταση περιστροφών σε τρεις διαστάσεις, παρέχοντας μια πιο αποδοτική εναλλακτική λύση σε σχέση με τα 3×3 μητρώα. Ενώ η χρήση ενός 3×3 μητρώου για την αναπαράσταση περιστροφών απαιτεί 9 αριθμούς κινητής υποδιαστολής, τα quaternions χρησιμοποιούν μόνο 4 αριθμούς. Αυτό επιτρέπει τη μείωση της υπολογιστικής πολυπλοκότητας, διότι η περιστροφή ενός διανύσματος με τη χρήση quaternions απαιτεί μόνο 4 πολλαπλασιασμούς

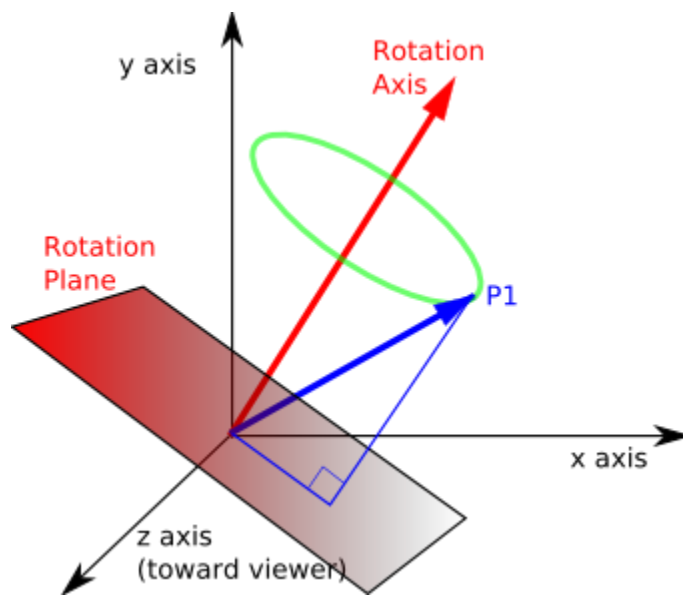
και 3 προσθέσεις, αντί για τους 9 πολλαπλασιασμούς και 6 προσθέσεις που απαιτούνται με τα μητρώα.

Τα quaternions που χρησιμοποιούνται στις αναπαραστάσεις περιστροφής είναι μοναδιαία, δηλαδή ισχύει η σχέση $q_x^2 + q_y^2 + q_z^2 + q_w^2 = 1$. Η χρήση τους επιτρέπει επίσης την εύκολη παρεμβολή (interpolation) μεταξύ διαφορετικών περιστροφών, καθώς τα quaternions δεν υποφέρουν από το φαινόμενο gimbal lock, το οποίο μπορεί να περιορίσει τη δυνατότητα περιστροφής όταν χρησιμοποιούνται άλλες μέθοδοι, όπως οι γωνίες Euler.

Ουσιαστικά, αποτελείται από 2 στοιχεία:

- Ένα μοναδιαίο άξονα περιστροφής $\hat{e}$ που κλιμακώνεται με το ημίτονο της μισής γωνίας περιστροφής. Η φορά περιστροφής ακολουθεί τον κανόνα του δεξιού χεριού.
- Το συνημίτονο της μισής γωνίας

Άρα, έχουμε $q = [q_v q_s] = [a \sin\left(\frac{\vartheta}{2}\right) \cos\left(\frac{\vartheta}{2}\right)]$



$$q = [q_x q_y q_z q_w]$$

όπου,

$$q_x = \hat{e}_x \sin\left(\frac{\vartheta}{2}\right)$$

$$q_y = \hat{e}_y \sin\left(\frac{\vartheta}{2}\right)$$

$$q_z = \hat{e}_z \sin\left(\frac{\vartheta}{2}\right)$$

$$q_w = \cos\left(\frac{\vartheta}{2}\right)$$

Πράξεις

Σημαντικό είναι να αναφερθεί ότι η πρόσθεση δύο quaternions δεν αντιστοιχεί στο άθροισμα δύο γωνιών. Ωστόσο, ο πολλαπλασιασμός είναι η πιο σημαντική πράξη, καθώς αντιπροσωπεύει μία περιστροφή ακολουθούμενη από μία άλλη, χρησιμοποιώντας τον εξής τύπο:

$$pq = [(p_s q_v + q_s p_v + p_v \times q_v) \quad (p_s q_s - p_v q_v)]$$

Συζυγείς και Αντίστροφοι

Το αντίστροφο q^{-1} ενός quaternion q , έχει την εξής ιδιότητα:

$$qq^{-1} = 0i + 0j + 0k + 1$$

$$\text{Συζυγής: } q^* = [-q_v \quad q_s]$$

$$\text{Αντίστροφος: } q^{-1} = q^* / |q|^2$$

Επιπλέον ιδιότητες είναι οι εξής:

$$(pq)^* = q^* p^*$$

$$(pq)^{-1} = q^{-1} p^{-1}$$

Περιστροφή με Quaternions

Για να περιστρέψουμε ένα διάνυσμα v κατά ένα quaternion q :

1. Γράφουμε το διάνυσμα σε μορφή quaternion

$$v = [v_x \quad v_y \quad v_z \quad 0]$$

2. Πολλαπλασιάζουμε:

$$v' = \text{rotate}(q, v) = qvq^{-1} \quad \text{ή} \quad v' = \text{rotate}(q, v) = qvq^*$$

3. Παίρνουμε το διάνυσμα από το $v' = [v' \quad 0]$

Πολλαπλές Περιστροφές με Quaternion

$$q_{\text{net}} = q_3 q_2 q_1$$

$$v' = q_3 q_2 q_1 v q_1^{-1} q_2^{-1} q_3^{-1} = q_{\text{net}} v q_{\text{net}}^{-1}$$

Περιστροφική Παρεμβολή (LERP)

Δοθέντων δύο quaternions q_A και q_B μπορούμε να βρούμε μία ενδιάμεση περιστροφή κατά ποσοστό β μεταξύ τους ως:

$$q_{LERP} = LERP(q_A, q_B, \beta) = \frac{(1 - \beta)q_A + \beta q_B}{|(1 - \beta)q_A + \beta q_B|} = \text{normalize} \left(\begin{bmatrix} (1 - \beta)q_{Ax} + \beta q_{Bx} \\ (1 - \beta)q_{Ay} + \beta q_{By} \\ (1 - \beta)q_{Az} + \beta q_{Bz} \\ (1 - \beta)q_{Aw} + \beta q_{Bw} \end{bmatrix}^T \right)$$

Υπάρχει όμως το πρόβλημα ότι το LERP παρεμβάλλει σε μία χορδή αντί της πραγματικής επιφάνειας της 4d υπερ-σφαίρας που περιγράφεται από το μοναδιαίο quaternion, άρα η γωνιακή ταχύτητα δεν είναι σταθερή. Όπως φαίνεται και από το σχήμα της επόμενης σελίδας, η αρχή και το τέλος είναι πιο αργά από τη μέση.

Σφαιρικό LERP

Το παραπάνω πρόβλημα επιλύεται με την σφαιρική γραμμική παρεμβολή (SLERP), καθώς το βάρος μεταξύ των δύο quaternion αλλάζει ανάλογα με τη μεταξύ τους γωνία, όπως φαίνεται από τον παρακάτω τύπο:

$$SLERP(p, q, \beta) = w_p p + w_q q$$

$$w_p = \frac{\sin(1 - \beta)\theta}{\sin \theta}$$

$$w_q = \frac{\sin \beta\theta}{\sin \theta}$$

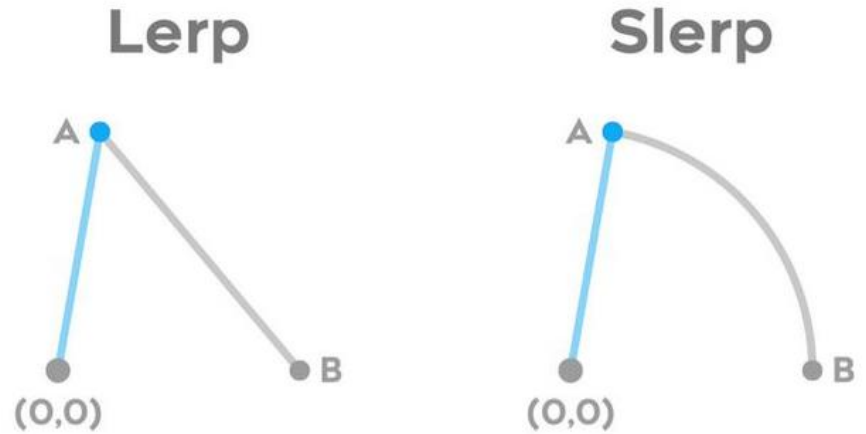
Για τον υπολογισμό της γωνίας θ έχουμε τον εξής τύπο:

$$\cos(\theta) = p \cdot q = p_x q_x + p_y q_y + p_z q_z + p_w q_w$$

$$\theta = \cos^{-1}(p \cdot q)$$

LERP ή SLERP

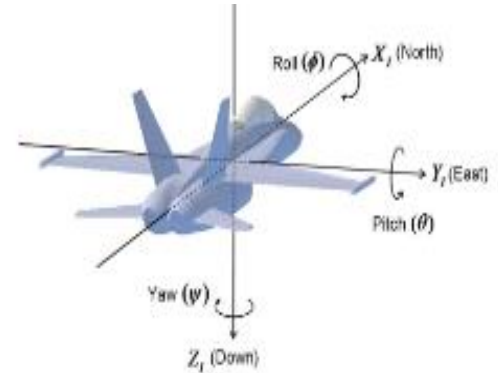
Όσον αφορά στην επιλογή ανάμεσα στα δύο παραπάνω είναι σημαντικό να αναφερθεί ότι το SLERP αν και καλύτερη μέθοδος είναι πιο αργή από τη LERP. Παρόλα αυτά, το LERP είναι στις περισσότερες περιπτώσεις επαρκές, χωρίς να υπάρχει διαφορά από την πλευρά του παίκτη.



Σύγκριση Αναπαράστασεων Περιστροφής

Γωνίες Euler

Αποτελούν μια απλή και μικρού μεγέθους αναπαράσταση περιστροφής, με τρεις γωνίες (Yaw, Pitch, Roll) που διευκολύνουν την οπτικοποίηση και είναι εύκολες στη χρήση και την παρεμβολή, εκτός εάν η περιστροφή γίνεται γύρω από αυθαίρετο άξονα. Παρουσιάζουν, ωστόσο, το πρόβλημα του gimbal lock, όπου η ευθυγράμμιση δύο αξόνων περιορίζει τους βαθμούς ελευθερίας (από τρεις βαθμούς ελευθερίας σε δύο) και εμποδίζει την ομαλή περιστροφή. Επιπλέον, η σειρά εφαρμογής των γωνιών (π.χ., PYR, YPR, RYP) επηρεάζει το τελικό αποτέλεσμα της περιστροφής, καθώς οι περιστροφές δεν είναι αντιμεταθετικές και η αλλαγή σειράς οδηγεί σε διαφορετική τελική στάση του αντικειμένου.



3 × 3 Μητρώα

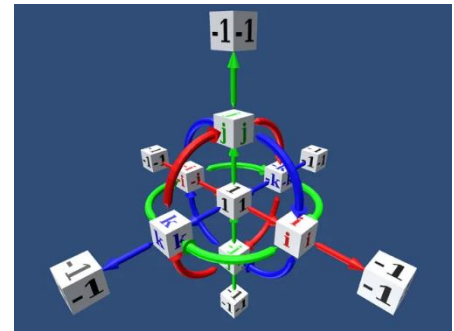
Χρησιμοποιούνται ευρέως στις CPU και GPU για αναπαράσταση περιστροφών χωρίς gimbal lock. Παρέχουν αξιόπιστη αναπαράσταση, αλλά απαιτούν υπολογιστικούς πόρους και δεν είναι εύκολη η παρεμβολή ή η ανάγνωσή τους.

Άξονας/Γωνία

Αναπαρίσταται με τέσσερις αριθμούς (ένας άξονας και μία γωνία), επιτρέποντας συμπαγή αναπαράσταση. Εύκολη στην κατανόηση, αλλά δύσκολη για παρεμβολή και χρήση σε σημεία ή διανύσματα χωρίς μετατροπή.

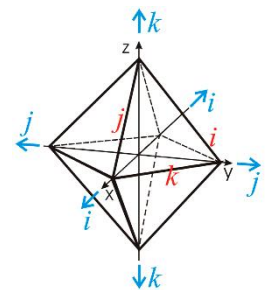
Quaternions

Παρέχουν αποδοτική και συμπαγή αναπαράσταση περιστροφών χωρίς gimbal lock και διευκολύνουν την παρεμβολή. Χρησιμοποιούνται ευρέως σε 3D γραφικά, αλλά η ανάγνωσή τους είναι δυσκολότερη. Για αναλυτικότερη περιγραφή βλ. [Quaternions](#).



SQT (SRT) Μετασχηματισμοί

Συνδυάζουν quaternions για περιστροφή με κλιμάκωση και μεταφορά. Αποτελούν μια συμπαγή και εύκολη στην κατανόηση αναπαράσταση για animation, υποστηρίζοντας παρεμβολή με LERP/SLERP. Αναλυτικότερα, γίνεται χρήση LERP σε κλιμάκωση και μεταφορά και LERP ή SLERP στην περιστροφή. Επίσης, χαρακτηριστικό τους είναι η συμπαγής αναπαράσταση (10 αριθμοί κινητής υποδιαστολής ή 8 ανάλογα αν έχουμε ομοιόμορφη ή ανομοιόμορφη κλιμάκωση – με ένα μητρώο 4×3 (χρειαζόμαστε 12 συνολικά) και παρουσιάζονται με την μορφή: $SQT = [s \ q \ t]$.



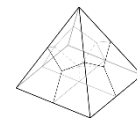
Χρήσιμα Γεωμετρικά Αντικείμενα

Σημείο – Κάθετος

Ένα επίπεδο αναπαρίσταται μέσω ενός σημείου και ενός κάθετου μοναδιαίου διανύσματος προς το επίπεδο.

Πλέγματα Τριγώνων

Τα τρισδιάστατα μοντέλα συντίθενται από τρίγωνα, προσφέροντας δυνατότητα διαχείρισης του φωτισμού και της ορατότητας χρησιμοποιώντας το κάθετο διάνυσμα ως προς το επίπεδο του τριγώνου σε σχέση με την πηγή φωτός. Σημαντικό είναι να ορίσουμε τις έννοιες τριγωνοποίηση και κατεύθυνση διαπέρασης. Τριγωνοποίηση είναι η διαδικασία διαχωρισμού πολυγώνου σε τρίγωνα. Η κατεύθυνση διαπέρασης καθορίζει τη σειρά με την οποία αναφέρουμε τις κορυφές του τριγώνου όταν βλέπουμε την εξωτερική του πλευρά (αυτή την πλευρά που βλέπει ο παίκτης).

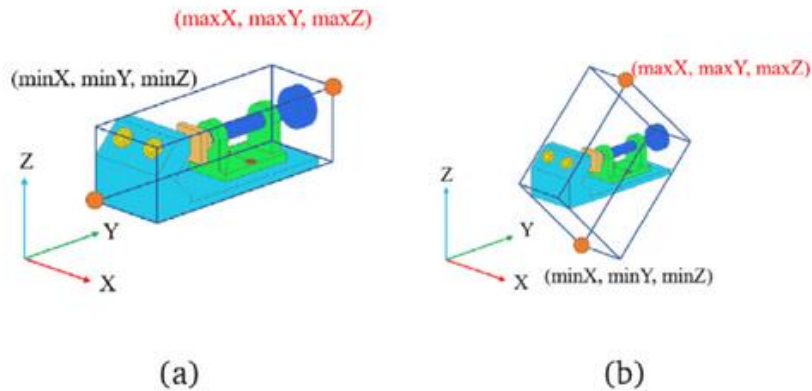


Axis Aligned Bounding Box

Ένα ορθοκανονικό πλαίσιο οριοθέτησης διευκολύνει τον έλεγχο σύγκρουσης μέσω έξι τιμών ή δύο οριακών σημείων (P_{min} , P_{max}). Ουσιαστικά, πρόκειται για ένα απλό μοντέλο περικλεισμένο σε ένα παραλληλεπίπεδο ευθυγραμμισμένο με τους άξονες X , Y , Z , με σημάνσεις για το ελάχιστο και το μέγιστο σημείο του.

Oriented Bounding Box

Πρόκειται για ένα αντικείμενο σε τρισδιάστατο χώρο περικλεισμένο σε ένα πλαίσιο οριοθέτησης που ακολουθεί τον προσανατολισμό του αντικειμένου και δεν είναι ευθυγραμμισμένο με τους άξονες. Διαφέρει από το AABB καθώς προσανατολίζεται με τον τοπικό χώρο συντεταγμένων του αντικειμένου, απαιτώντας μετατροπή του σημείου στο σύστημα συντεταγμένων του OBB για έλεγχο τομής.

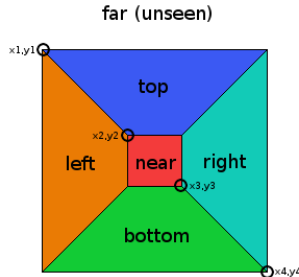


(a) Axis Aligned Bounding Box

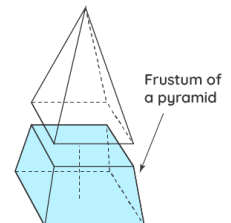
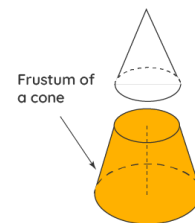
(b) Oriented Bounding Box

Frusta

Αναπαρίσταται ως μία κομμένη πυραμίδα με έξι επίπεδα, χρήσιμο για τη διαχείριση του χώρου ορατότητας (view frustum) σε γραφικά. Συνήθως αναπαρίσταται από έναν πίνακα 6 επιπέδων με μορφή σημείο-κάθετος.

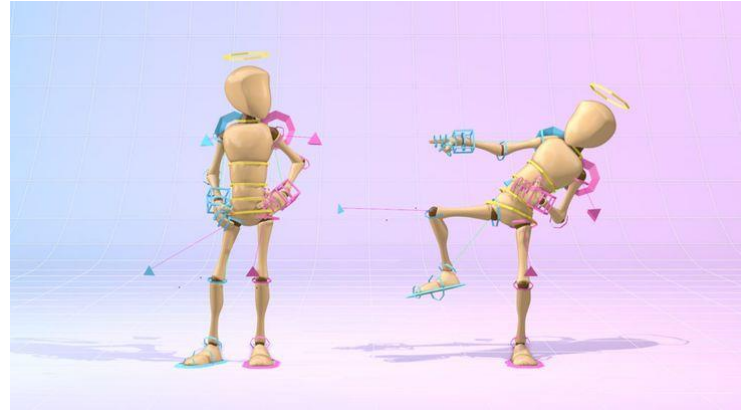


Types of Frustum



Rigging

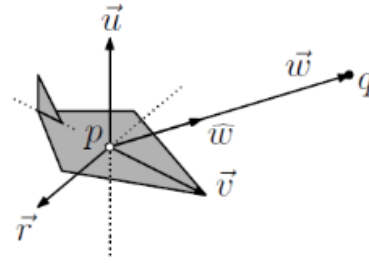
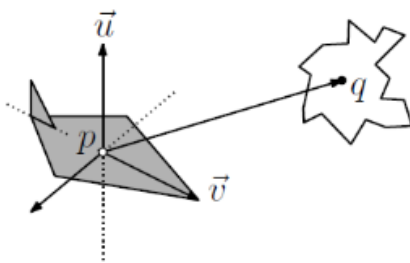
Διαδικασία δημιουργίας σκελετού για τρισδιάστατο μοντέλο, επιτρέποντας την κίνηση με άρθρωση. Ο σκελετός αποτελεί ένα ιεραρχικό δέντρο μετασχηματισμών. Αναλυτικότερα, κάθε άρθρωση έχει έναν μετασχηματισμό που σχετίζεται με αυτήν, για παράδειγμα τα δάχτυλα μετασχηματίζονται από ένα γινόμενο μητρώων που ξεκινά από το στήθος, μετά στον ώμο, μετά στο μπράτσο, μετά στον αγκώνα, μετά στον καρπό, κ.λπ.



Άσκηση

Μανούβρες Αποφυγής

Στο παιχνίδι που φτιάχνετε, προσομοιώνετε την τεχνητή νοημοσύνη για μερικά εξωγήινα διαστημόπλοια. Κάθε διαστημόπλοιο σχετίζεται με την τρέχουσα θέση του, ένα σημείο p και δύο διανύσματα μοναδιαίου μήκους. Το πρώτο διάνυσμα $\vec{v}$ υποδεικνύει την κατεύθυνση στην οποία κινείται το διαστημόπλοιο. Το δεύτερο διάνυσμα $\vec{u}$ είναι ορθογώνιο προς το $\vec{v}$ και υποδεικνύει την κατεύθυνση «πάνω» σε σχέση με τον πιλότο του διαστημόπλοιου. Υπάρχουν διάφορα εμπόδια που πρέπει να αποφευχθούν (αστεροειδείς) και το σύστημα AI να πρέπει να δίνει εντολές αλλαγής πορείας για να αποφευχθούν τα εμπόδια. Με δεδομένο ένα εμπόδιο σε κάποιο σημείο q , θέλουμε να καθορίσουμε αν πρέπει να στρίψει (yaw) προς τα αριστερά ή προς τα δεξιά και αν πρέπει να στρίψει (pitch) πάνω ή κάτω για να αποφευχθεί η σύγκρουση. (Δεν μας ενδιαφέρουν οι γωνίες αλλά μόνο η κατεύθυνση.)



Λύση

1. Φτιάχνουμε διάνυσμα $\vec{w} = q - p$, όπως φαίνεται στην δεύτερη εικόνα.
2. Χρήση νοητού άξονα που δημιουργείται από διάνυσμα $\vec{u}$ για να δούμε αν θα κινηθούμε προς τα πάνω ή προς τα κάτω.
Αν η γωνία είναι μικρότερη από 90° (όπως στο παράδειγμα), το εσωτερικό γινόμενο των διανυσμάτων $\vec{u}, \vec{w}$ είναι μεγαλύτερο ή ίσο του 0 και άρα ο αστεροειδής βρίσκεται από πάνω μας. Επομένως εμείς επιλέγουμε να πάμε προς τα κάτω. Συμμετρική είναι και η άλλη περίπτωση όπου το εσωτερικό γινόμενο είναι αρνητικό και πρέπει να κινηθούμε προς τα πάνω.
3. Δημιουργία διανύσματος για να επιλέξουμε κίνηση προς τα δεξιά ή αριστερά.
Δημιουργούμε διάνυσμα $\vec{r}$ από το εξωτερικό γινόμενο των $\vec{u}$ και $\vec{v}$, το οποίο είναι κάθετο σε αυτά τα διανύσματα. Μελετάμε το εσωτερικό γινόμενο των διανυσμάτων

$\vec{r}$ και $\vec{w}$. Αν είναι μεγαλύτερο ή ίσο του 0 τότε η γωνία που σχηματίζουν είναι οξεία. Αυτό σημαίνει ότι το εμπόδιο βρίσκεται στα δεξιά και εμείς πρέπει να πάμε προς τα αριστερά για να επιτευχθεί η αποφυγή του. Ομοίως, αν το εσωτερικό γινόμενο είναι αρνητικό τότε θα πρέπει να στρίψουμε προς τα δεξιά.

Δεν μπορούμε με αυτά τα δεδομένα να υπολογίσουμε τη γωνία περιστροφής, καθώς εξαρτάται από πολλές παραμέτρους όπως ο όγκος του αστεροειδούς, η θέση των ακραίων σημείων του και άλλα. Μην ξεχνάτε όμως ότι επειδή εσείς φτιάχνετε το παιχνίδι μπορείτε να καθορίσετε τους αστεροειδής έτσι ώστε αυτοί οι υπολογισμοί να είναι εύκολοι να γίνουν. Κάτι τέτοιο βεβαίως δεν θα ίσχυε σε ένα πραγματικό διαστημόπλοιο (αν θέλατε να φτιάξετε ένα αυτόματο σύστημα αποφυγής εμποδίων) στο οποίο τα αντικείμενα μπορεί να έχουν αυθαίρετο σχήμα αν και υπόκεινται στους φυσικούς νόμους.