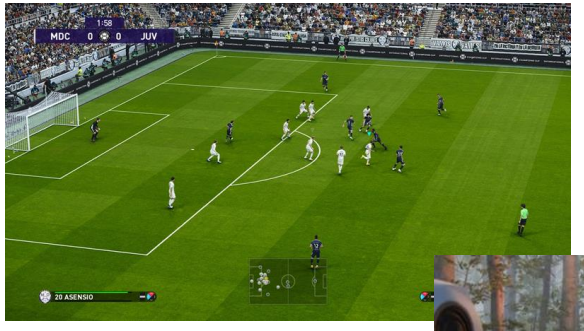
Το Σύστημα Φυσικής και το Σύστημα Ήχου

Η Φυσική στα Παιχνίδια

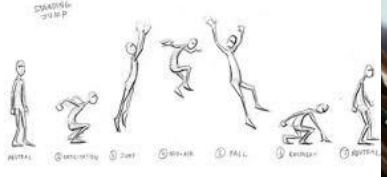
- Δεν θέλουμε ένα τέλειο φυσικό μοντέλο (είναι ακριβό!)
- Συνήθως γίνεται προσέγγιση υποθέτοντας Νευτώνεια φυσική και άκαμπτα σώματα (rigid bodies)
- Χρήση τεχνικών διακριτής προσομοίωσης
- Η διαχείριση του κέντρου μάζας είναι συνήθως αρκετή

Γιατί να Προσθέσω Φυσική στο Παιχνίδι μου;

- Λόγω της εμπειρίας του παίκτη
  - Οι κινήσεις του πραγματικού κόσμου βασίζονται στη φυσική
  - Η φυσική μπορεί να κάνει πιο γνώριμους τους εικονικούς κόσμους
  - Σε ορισμένους τύπους παιχνιδιών απαιτείται

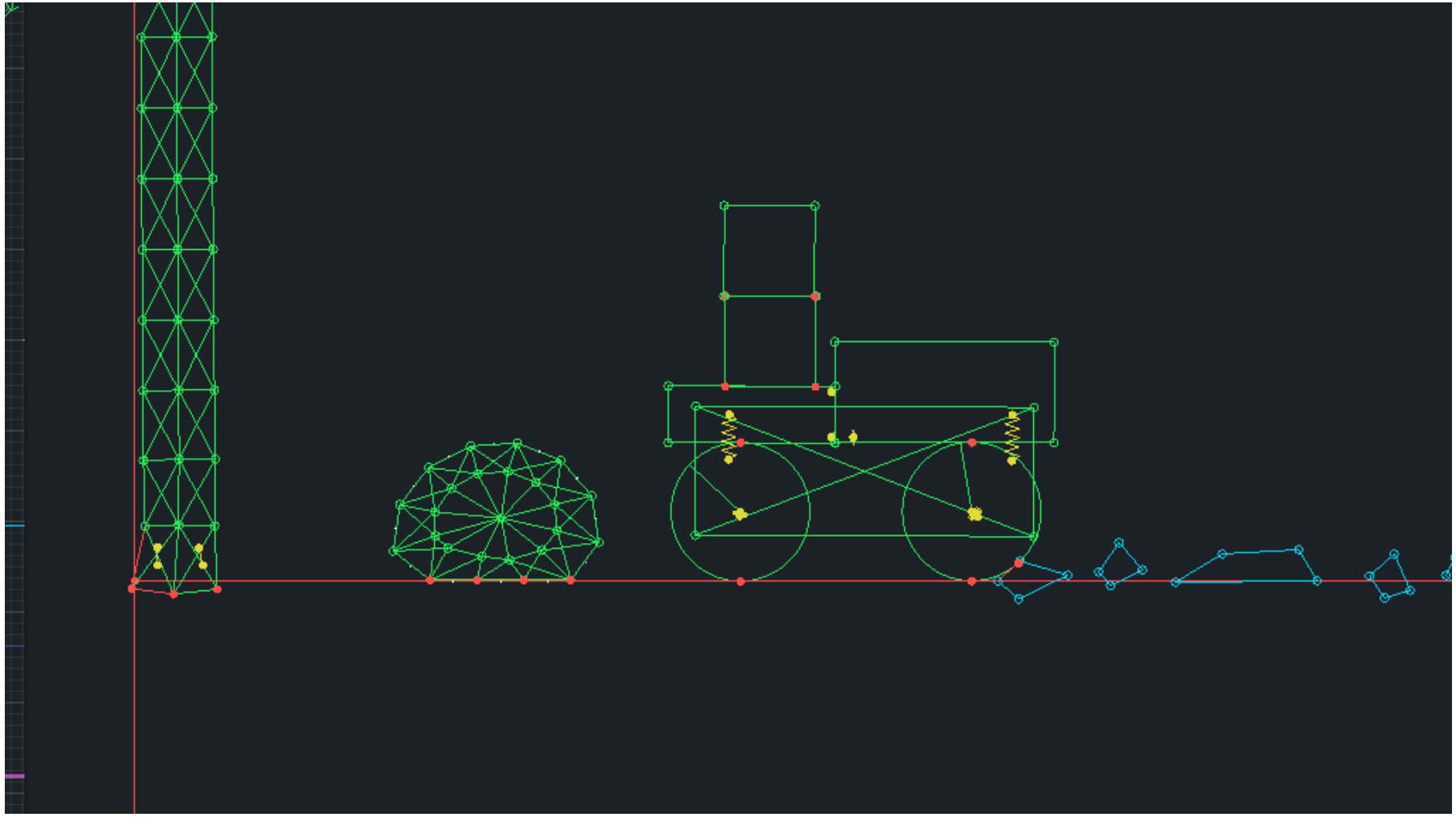


- Αναδυόμενη Συμπεριφορά
  - Η προσομοίωση φυσικής προσφέρει μία πιο πλούσια εμπειρία



- Κόστος
  - Κλασικές προσεγγίσεις για τη δημιουργία ρεαλιστικής κίνησης:
    - Κινούμενα σχέδια με καρέ από καλλιτέχνη
    - Λήψη κίνησης (Motion Capture)
    - Και τα δύο ακριβά και χρονοβόρα
  - Προσομοίωση φυσικής:
    - Αλγοριθμική κίνηση
    - Θεωρητικά απαιτεί «ελάχιστη» καλλιτεχνική συμβολή
    - Δυνατότητα ουσιαστικής μείωσης του κόστους ανάπτυξης περιεχομένου

Τι Αφορά;



- Συγκρούσεις μεταξύ αντικειμένων
  - Ανίχνευση σύγκρουσης: Συνέβη σύγκρουση;
  - Απόκριση σύγκρουσης: Αν ναι, τι κάνουμε;

- Μετακίνηση αντικειμένων στην οθόνη
  - Κινηματική: Κίνηση που αγνοεί τις εξωτερικές δυνάμεις (μόνο θέση, ταχύτητα, επιτάχυνση)
  - Δυναμική: Η επίδραση των δυνάμεων

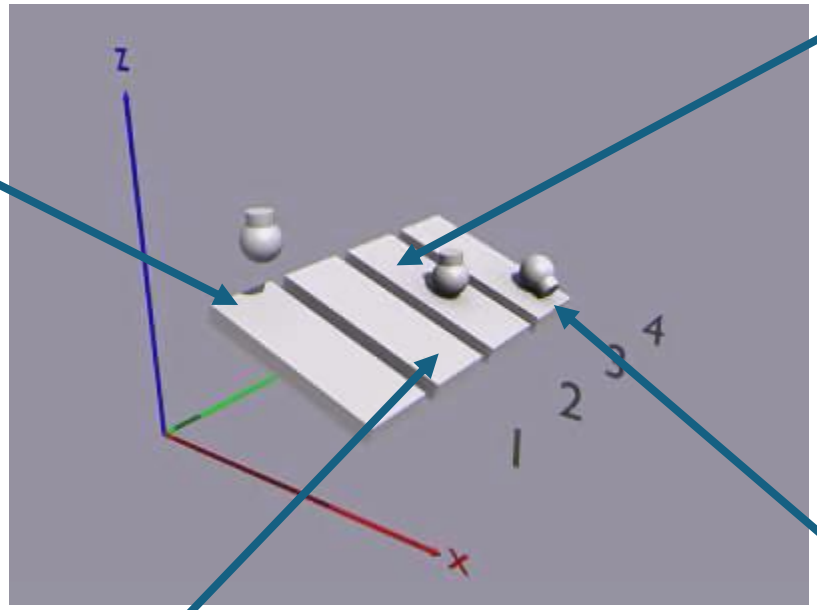
Μηχανές Φυσικής

Physics Engines

Προσομοίωση από Μηχανές Φυσικής

(1) Καμία φυσική

(3) Βαρύτητα και ανίχνευση σύγκρουσης χωρίς δυναμική άκαμπτων σωμάτων



Προσομοιώνοντας ένα αντικείμενο καθώς πέφτει σε μία πλαγιά

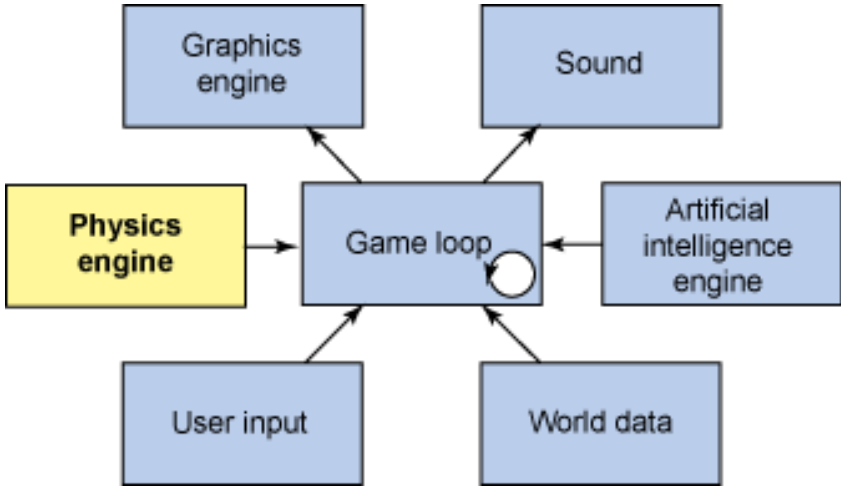
(2) Βαρύτητα αλλά όχι ανίχνευση σύγκρουσης

(4) Βαρύτητα, ανίχνευση σύγκρουσης και υπολογισμοί περιστροφής

Πραγματικού Χρόνου έναντι Υψηλής Ακριβείας Προσομοίωση Φυσικής

Σε πραγματικό χρόνο:

- βιντεοπαιχνίδια και διαδραστικός υπολογισμός
- απλοποιημένοι υπολογισμοί
- μειωμένη ακρίβεια



Υψηλής ακρίβειας:

- Επιστημονικές προσομοιώσεις & ταινίες κινουμένων σχεδίων σε υπολογιστή
- Περισσότερη επεξεργαστική ισχύ
- Ακριβής

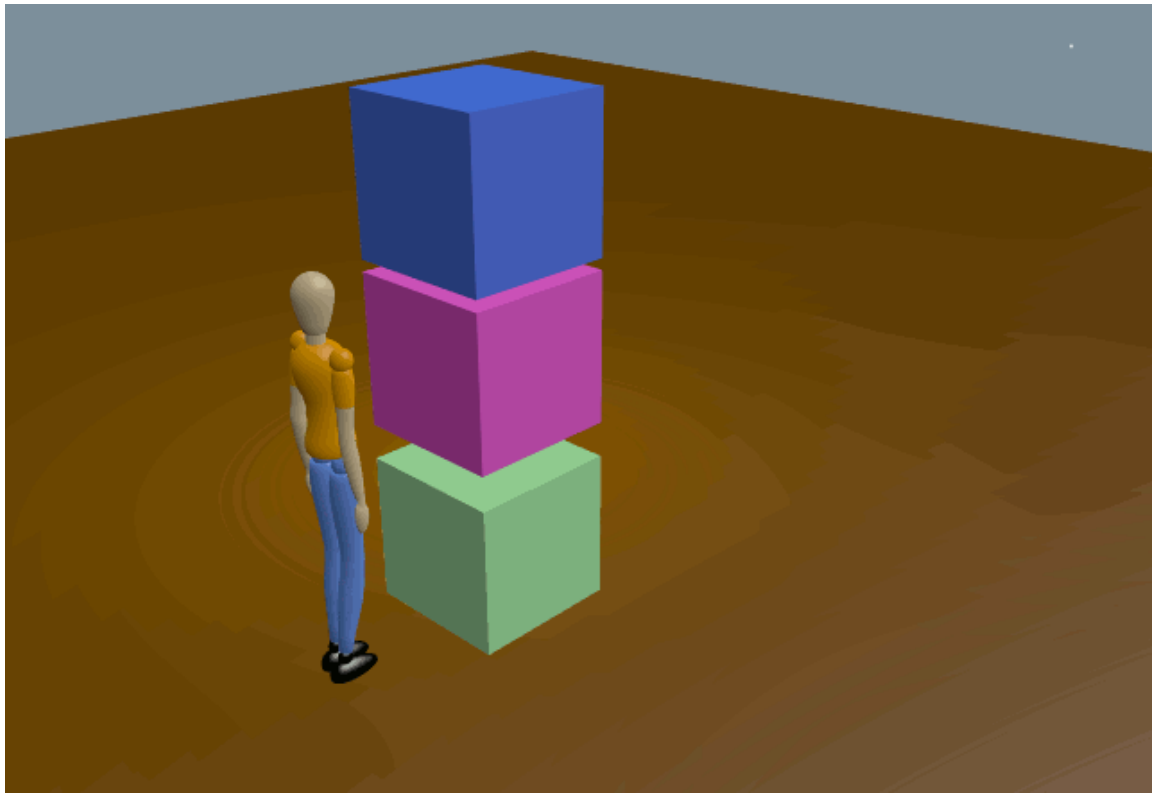
Βασικά Στοιχεία Μηχανής Φυσικής

- Προσδιορίζει και παρέχει κατά προσέγγιση προσομοίωση ορισμένων απλών φυσικών συστημάτων.
- Δυναμική άκαμπτων σωμάτων
- Δυναμική μαλακών σωμάτων
- Προσομοίωση ρευστών
- Ανίχνευση και απόκριση σύγκρουσης
- …

Δυναμική Άκαμπτου Σώματος

- Χρήση:
  - η περιστροφική κίνηση είναι σημαντική
  - η *παραμόρφωση του υλικού δεν έχει σημαντική επίδραση* στην κίνηση του σώματος.

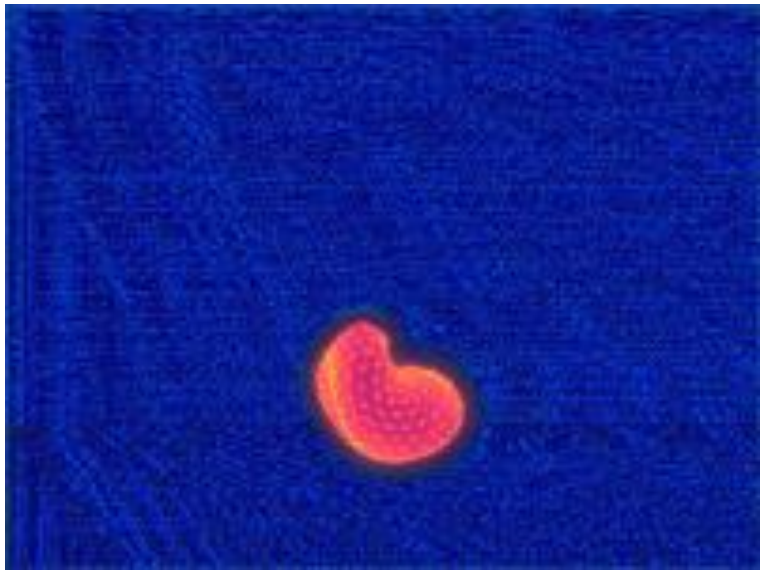
- Αφορά την:
  - γραμμική ορμή του άκαμπτου σώματος
  - γωνιακή ορμή του άκαμπτου σώματος



Δυναμική Μαλακού Σώματος

- Εστιάζει σε οπτικά ρεαλιστικές φυσικές προσομοιώσεις.

- Αυτή η προσομοίωση μπορεί να γίνει χρησιμοποιώντας μια ποικιλία προσεγγίσεων:
  - Μοντέλα μάζας-ελατηρίου
  - Προσομοίωση πεπερασμένων στοιχείων (FEM)
  - Μέθοδοι ελαχιστοποίησης ενέργειας
  - Ταίριασμα σχήματος



Προσομοίωση Υγρών

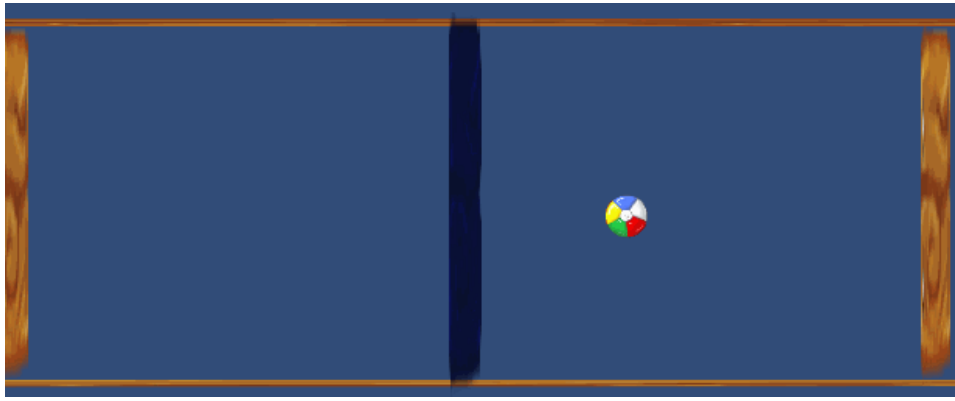
- Όλο και πιο δημοφιλής
- Εξισώσεις Navier-Stokes
- Οι προσομοιώσεις ποικίλλουν σε πολυπλοκότητα από *εξαιρετικά χρονοβόρα κινούμενα σχέδια υψηλής ποιότητας για φιλμ και οπτικά εφέ*, έως *απλά συστήματα σωματιδίων σε πραγματικό χρόνο* που χρησιμοποιούνται σε σύγχρονα παιχνίδια.

Μερικές τεχνικές για την προσομοίωση υγρών είναι οι εξής:

- Μέθοδοι που βασίζονται σε πλέγμα Euler
- Μέθοδοι Lattice Boltzmann
- Μέθοδοι που βασίζονται σε Νευρωνικά Δίκτυα
- …

Ανίχνευση Σύγκρουσης και Απόκριση

- Ανίχνευσης της τομής δύο ή περισσότερων αντικειμένων.
- Η απόκριση σύγκρουσης ασχολείται με την προσομοίωση του τι συμβαίνει όταν ανιχνεύεται μια σύγκρουση.
- Η επίλυση προβλημάτων ανίχνευσης σύγκρουσης απαιτεί εκτενή χρήση εννοιών από τη γραμμική άλγεβρα και υπολογιστική γεωμετρία:
  - Ανίχνευση σύγκρουσης για εύκαμπτες επιφάνειες
  - Ανίχνευση σύγκρουσης για κυρτά πολύεδρα
  - Απόκριση σύγκρουσης με χρήση ελατηρίων
  - Απόκριση σύγκρουσης με χρήση αναλυτικής λύσης
  - Συγκρούσεις δυναμικών αντικειμένων με μη δυναμικά αντικείμενα



Εμπορικές Πραγματικού-Χρόνου Μηχανές Φυσικής

Havok	3d	free for non-commercial and inexpensive commercial PC games, fee-based for games above a price limit, non-game applications, and consoles.
NVIDIA PhysX	3d	free for PC-based games, free for PS3 through Sony pre-purchase, fee-based for other consoles
Digital Molecular Matter (DMM)	3d	fee-based licensing
SPE Simple Physics Engine	3d	free for noncommercial, fee-based for commercial
Chrono :: Engine	3d	free for noncommercial use, fee-based for commercial
True Axis	3d	free for noncommercial, fee-based for commercial with a reasonable indie license option
Gino van den Bergen's SOLID collision detection library	3d	fee-based for commercial, open sourced for GPL/QPL projects
SimVex	3d	Addon library for Havok enabling game-quality vehicle creation of any configuration, cars, motorcycles, aircraft, etc
CarX	3d	C++, fee-based licensing
Matali Physics	3d	commercial physics engine, free for use in non-commercial games

- Συνήθως σε C++
- Πολλές έχουν τιμολογιακές πολιτικές που είναι προσιτές σε Indie ή χομπίστες

Ανοικτού Κώδικα και Δωρεάν Πραγματικού-Χρόνου Μηχανές Φυσικής

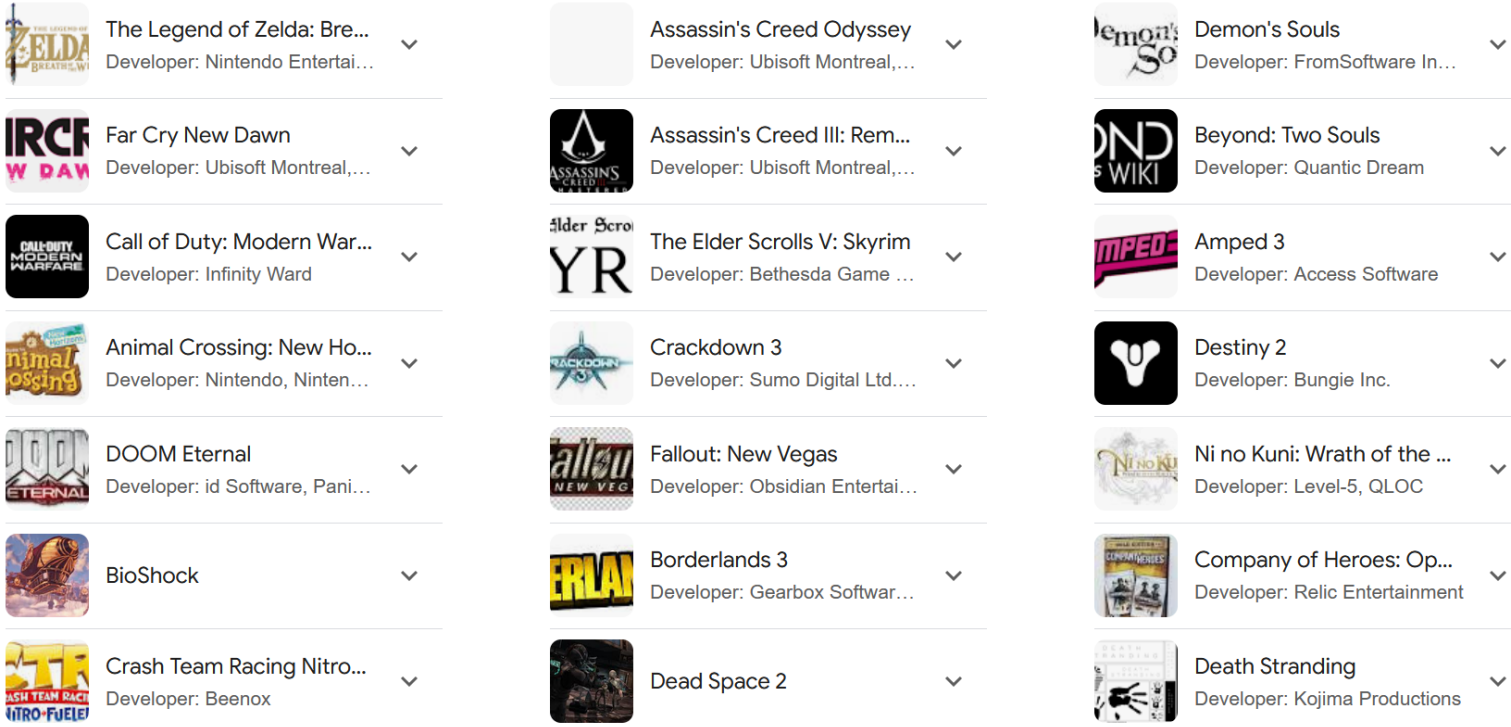
Bullet	3d	open source Zlib license
Open Dynamics Engine	3d	open source BSD license
Box2D	2d	open source permissive Zlib license
Tokamak Physics	3d	open source BSD license - Sourceforge Link
Newton Game Dynamics	3d	custom free-use license
JigLib	3d	open source Zlib license
Chipmunk	2d	open source unrestrictive MIT license
Open Tissue	various	open source Zlib license

Havok

Havok middleware suite consists of the following modules:

- Havok Physics (video games mainly: collision detection/response, rigid body dynamics, continuous physics, vehicle dynamics)
- Havok Cloth (character garments and soft body dynamics)
- Havok Navigation (AI)

Ευρεία χρήση...

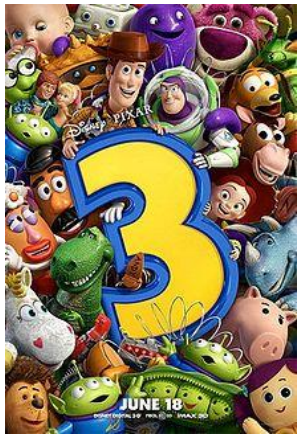


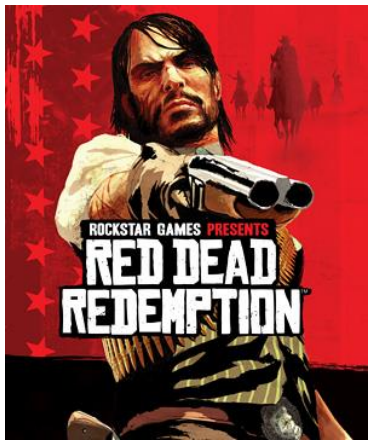
PhysX

- PhysX is a proprietary real-time physics engine middleware SDK.
- PhysX SDK has fully-functional GPU/PPU physics acceleration pipeline.
- Supports:
  - rigid body dynamics
  - soft body dynamics (like cloth simulation, including tearing and pressurized cloth)
  - ragdolls and character controllers
  - vehicle dynamics
  - particles and volumetric fluid simulation

Bullet

- Bullet is an open source physics engine featuring 3D collision detection, soft body dynamics, and rigid body dynamics.
- It is used in games, and in visual effects in movies.
- Features:
  - Rigid body and soft body simulation with discrete and continuous collision detection
  - Multiple collision shapes
  - Soft body support: cloth, rope and deformable objects
  - A rich set of rigid body and soft body constraints with constraint limits and motors

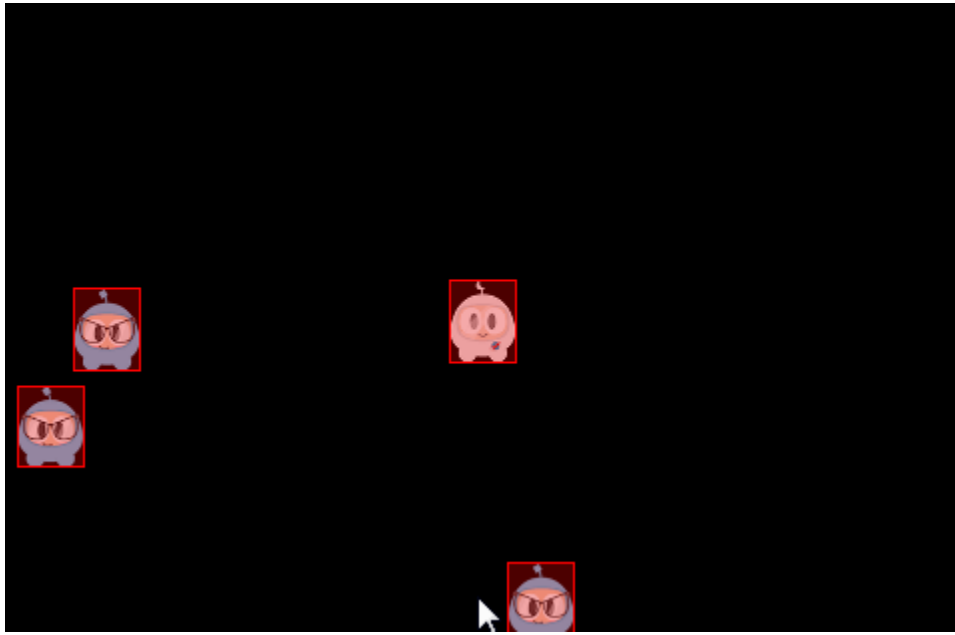




Ανίχνευση Σύγκρουσης & Απόκριση

Συγκρούσεις

- Ανίχνευση σύγκρουσης:
  - Συγκρούονται οι κινούμενες οντότητες;
  - Κυρίως υπολογιστική γεωμετρία

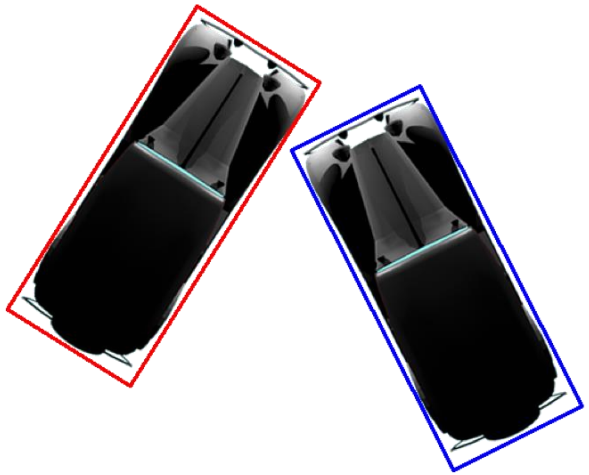


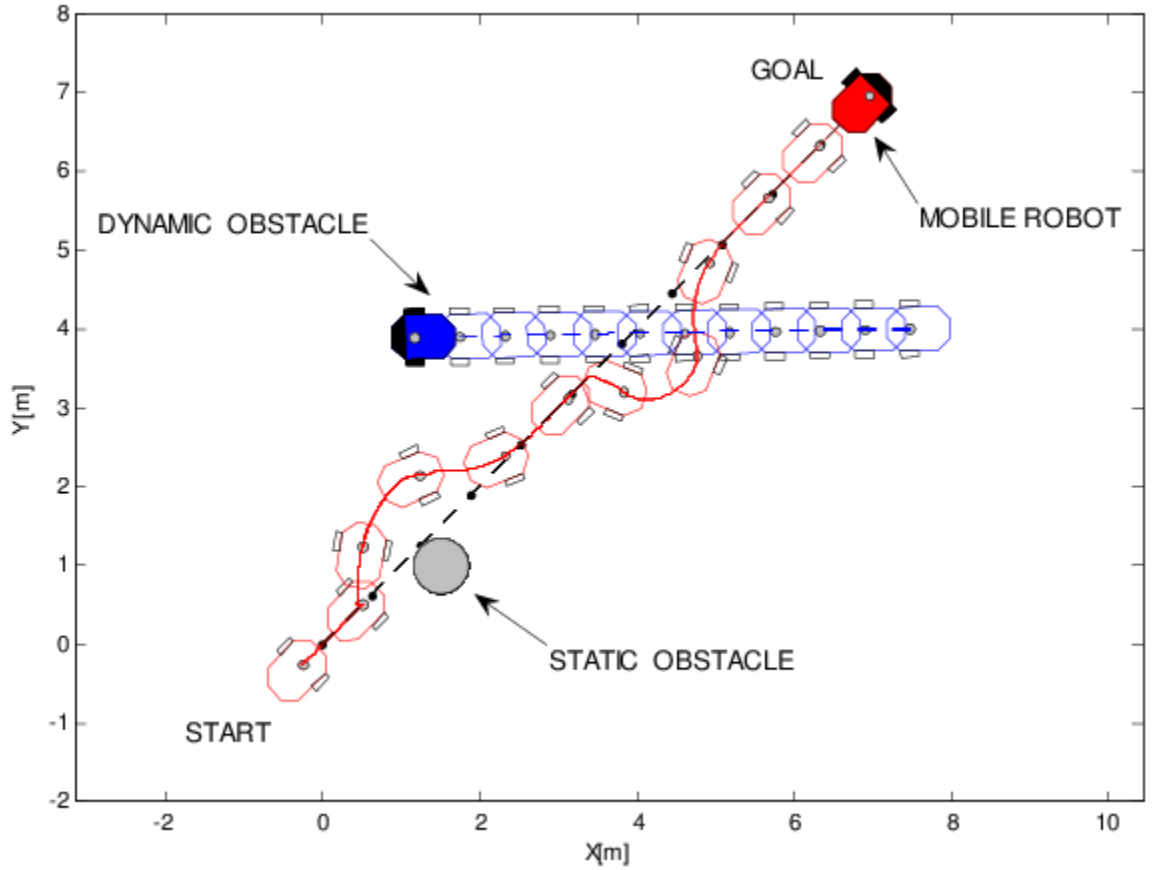
- Απόκριση σύγκρουσης
  - Τί γίνεται όταν υπάρχει σύγκρουση
  - Κυρίως φυσική

- Μία από τις πιο κοινές εργασίες στην ανάπτυξη ενός παιχνιδιού
  - Συνήθης πηγή λαθών

Στατικά έναντι Δυναμικών Αντικειμένων

- Τα στατικά αντικείμενα δεν κινούνται – τα δυναμικά κινούνται
- Σύγκρουση μεταξύ στατικών και δυναμικών αντικειμένων: Πιο εύκολο
- Σύγκρουση μεταξύ δύο (ή περισσότερων) δυναμικών αντικειμένων: Πιο δύσκολο





Βρόχος Παιχνιδιού

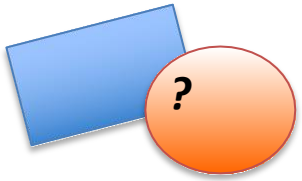
Αφελής προσέγγιση:

```
for(i=0;i<num_obj-1;i++)
  for(j=i+1;j<num_obj;j++)
     if(collide(i,j)){
        react;
     }
```

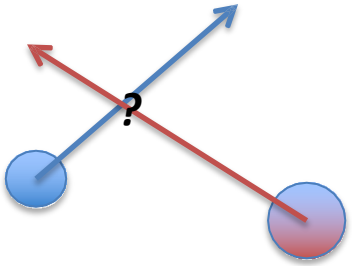
- Θέματα:
  - Πώς γίνεται ο έλεγχος και η απόκριση;
  - Πολύ αργό…

Τεχνικές Ανίχνευσης Σύγκρουσης

- Έλεγχος επικάλυψης
  - Ανίχνευση εάν έχει ήδη συμβεί σύγκρουση
  - Η πιο κοινή τεχνική



- Έλεγχος τομής
  - Πρόβλεψη σύγκρουσης



Διαφορετικές Οπτικές Γωνίες

- Λεπτομερής οπτική γωνία (μικροσκοπική άποψη)
  - Σύγκρουση τριγώνων και πλεγμάτων
- Μεσαίου επιπέδου οπτική γωνία (μεσοσκοπική άποψη)
  - Απλουστευμένη γεωμετρία
- Καθολική οπτική γωνία (μακροσκοπική άποψη)
  - Δομές δεδομένων διαμέρισης του χώρου

Έλεγχος Τομής

Κοιτώντας σε λεπτομέρεια

(Μικρο/Μεσο-σκοπική Άποψη)

Έλεγχος Τομής

- Πρόβλεψη μελλοντικών συγκρούσεων

- Όταν προβλεφθεί μία τέτοια σύγκρουση:
  - Προχωρήστε την προσομοίωση στη στιγμή της σύγκρουσης
  - Επίλυση σύγκρουσης
  - Προσομοίωση υπολειπόμενου χρονικού βήματος

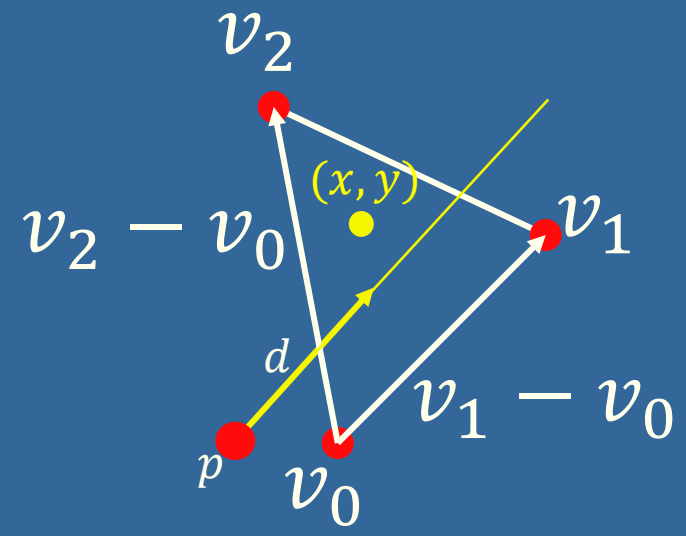
- Υποθέστε σταθερή ταχύτητα (για κάποιο χρονικό διάστημα)
  - Ιδανικό για σύγκρουση μεταξύ δυναμικού και στατικού αντικειμένου

Τέσσερις Διαφορετικές Τεχνικές

- Αναλυτικές
- Γεωμετρικές
- Το Θεώρημα των Διαχωριστικών Αξόνων (Separating Axis Theorem – SAT)
- Δυναμικοί έλεγχοι

  – *Πολλές φορές όμως χρειάζονται «κόλπα» για να τρέξουν γρήγορα*

Αναλυτικές: Τομή Ακτίνας/Τριγώνου



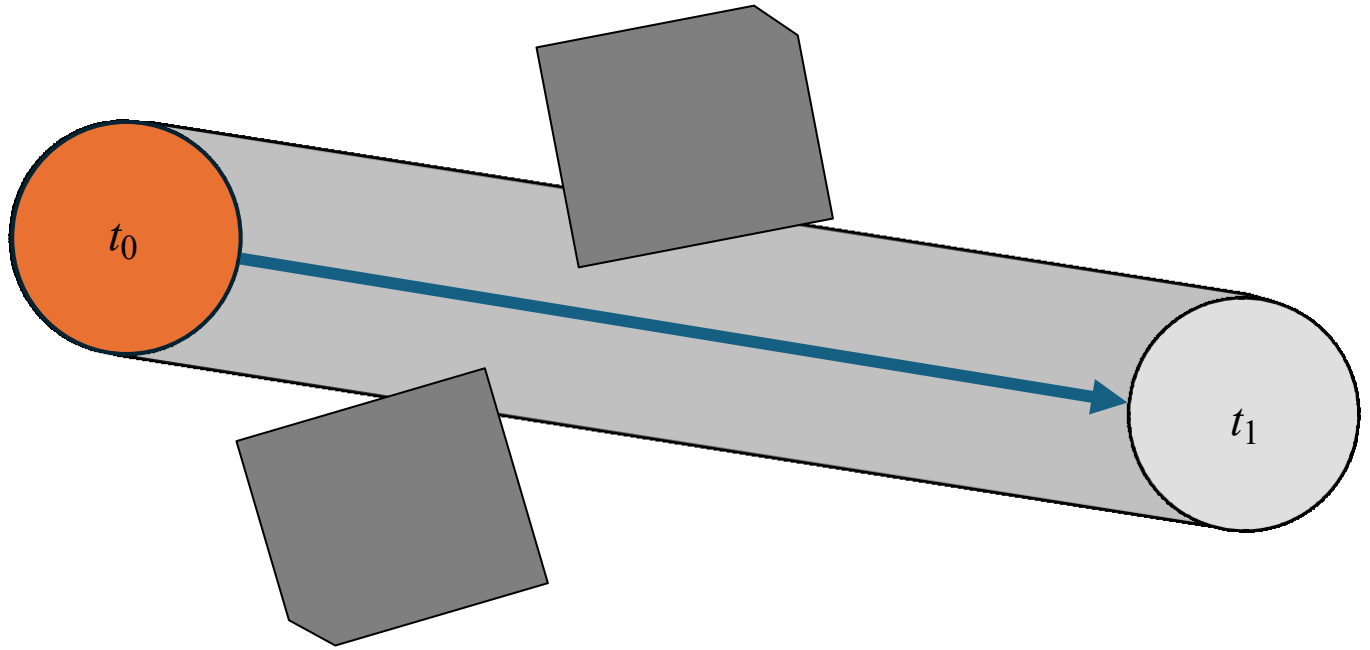
- Ακτίνα: $r(t) = p + td$
- Κορυφές τριγώνου: v_0, v_1, v_2
- Ένα αυθαίρετο σημείο πάνω στο τρίγωνο:

$$t(x, y) = v_0 + x(v_1 - v_0) + y(v_2 - v_0)$$
$$= (1 - x - y)v_0 + xv_1 + yv_2$$
$$u, v \geq 0, u + v \leq 1$$

- Θέτουμε $t(x, y) = r(t)$, και λύνουμε!

Γεωμετρικές: Κινούμενη Σφαίρα

- **Επέκταση** της γεωμετρίας κατά τη κατεύθυνση της κίνησης
  - η σφαίρα μετατρέπεται σε «κάψουλα»

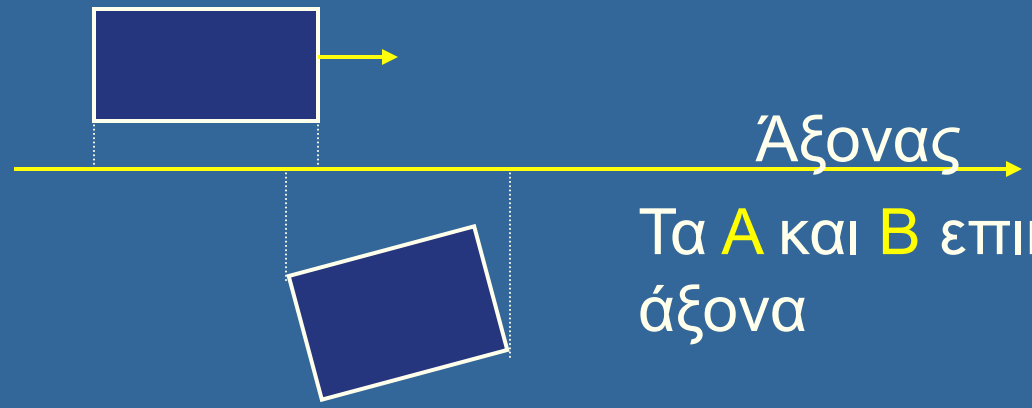


- Έπειτα, έλεγχος για επικάλυψη!

SAT Θεώρημα (Όχι Sipser ☺)

Δύο κυρτά πολύεδρα A και B είναι ξένα εάν κάποιος από τους παρακάτω άξονες χωρίζει τα αντικείμενα:

- Ένας άξονας ορθογώνιος προς μια όψη του A
- Ένας άξονας ορθογώνιος προς μια όψη του B
- Ένας άξονας που σχηματίζεται από το εξωτερικό γινόμενο μιας ακμής από καθένα από τα A και B

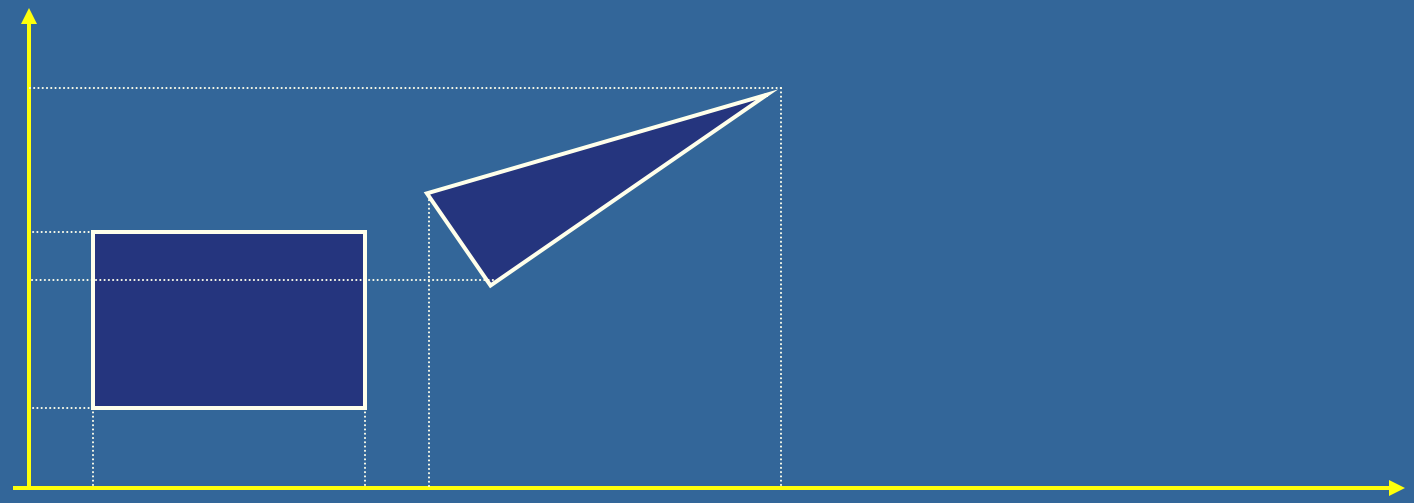


Τα A και B επικαλύπτονται σε αυτόν τον άξονα

Τρίγωνο – Ορθογώνιο

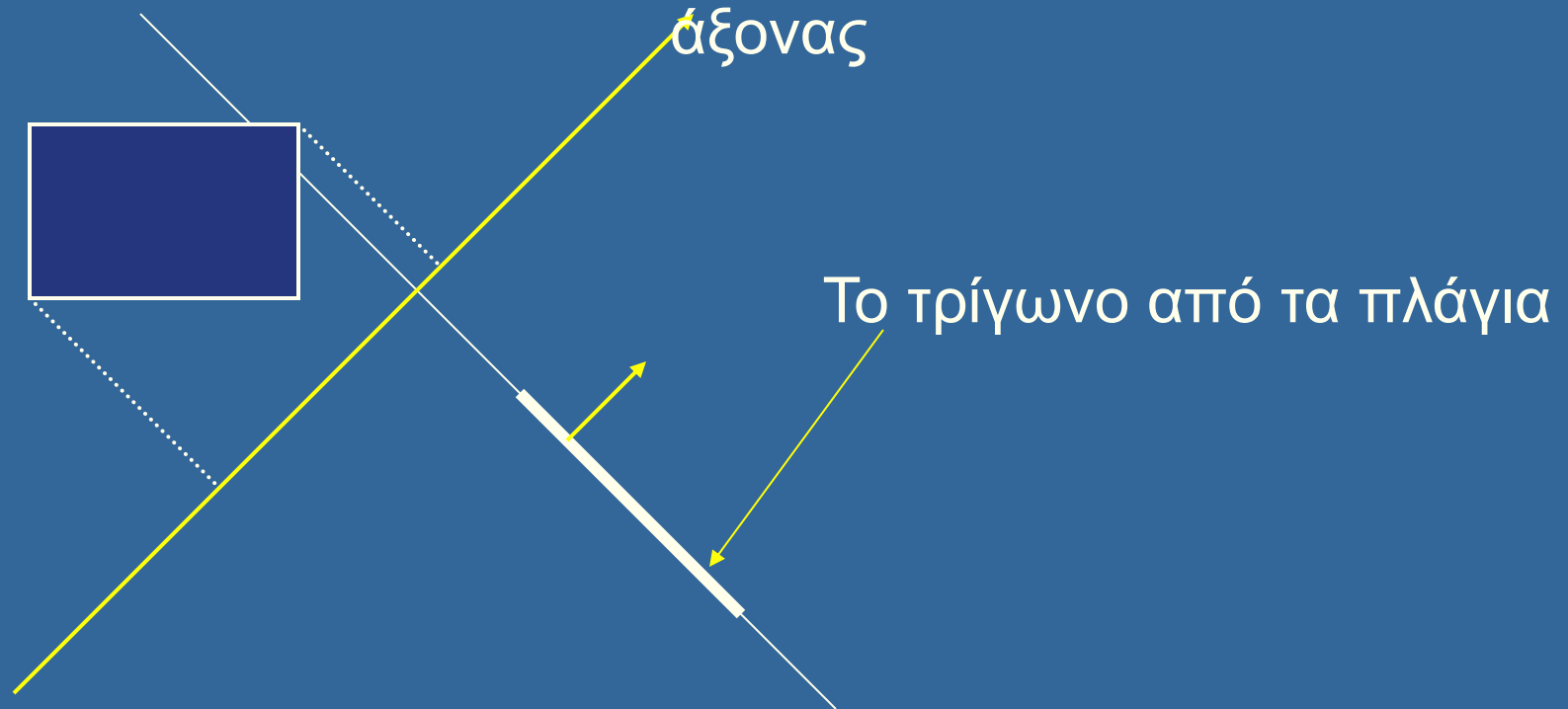
Το ορθογώνιο είναι ορθοκανονικό

1) Ελέγχουμε τους άξονες που είναι ορθογώνιο προς τις όψεις του ορθογωνίου (ως προς x, y, z)



Τρίγωνο – Ορθογώνιο

- Έστω ότι επικαλύπτονται σε x, y, z
- Συνεχίζουμε τον έλεγχο με τον άξονα που είναι ορθογώνιος στην όψη του τριγώνου



Τρίγωνο – Ορθογώνιο

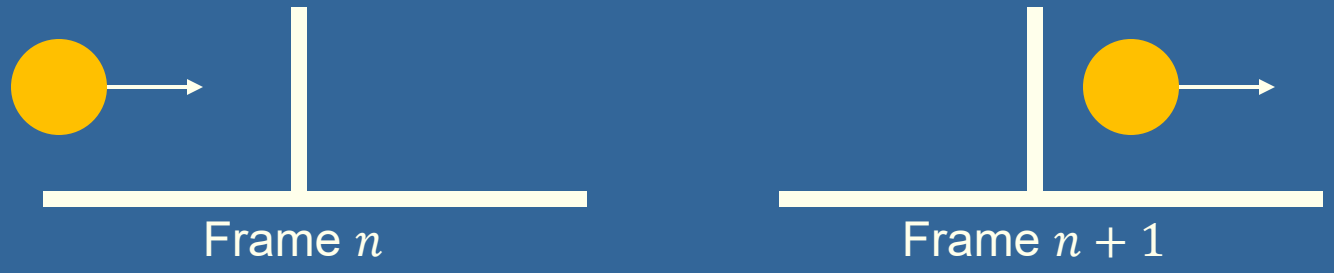
Αν ακόμα δεν βρέθηκε διαχωριστικός άξονας τότε

Έλεγχος άξονα: $t = e_{box} \times e_{triangle}$

- Παράδειγμα:
  - x-άξονας από το ορθογώνιο: $e_{box} = (1,0,0)$
  - $e_{triangle} = v_1 - v_0$
- Έλεγχος όλων των συνδυασμών
- Αν υπάρχει τουλάχιστον ένας διαχωριστικός άξονας τότε τα αντικείμενα δεν επικαλύπτονται
- Αλλιώς, επικαλύπτονται

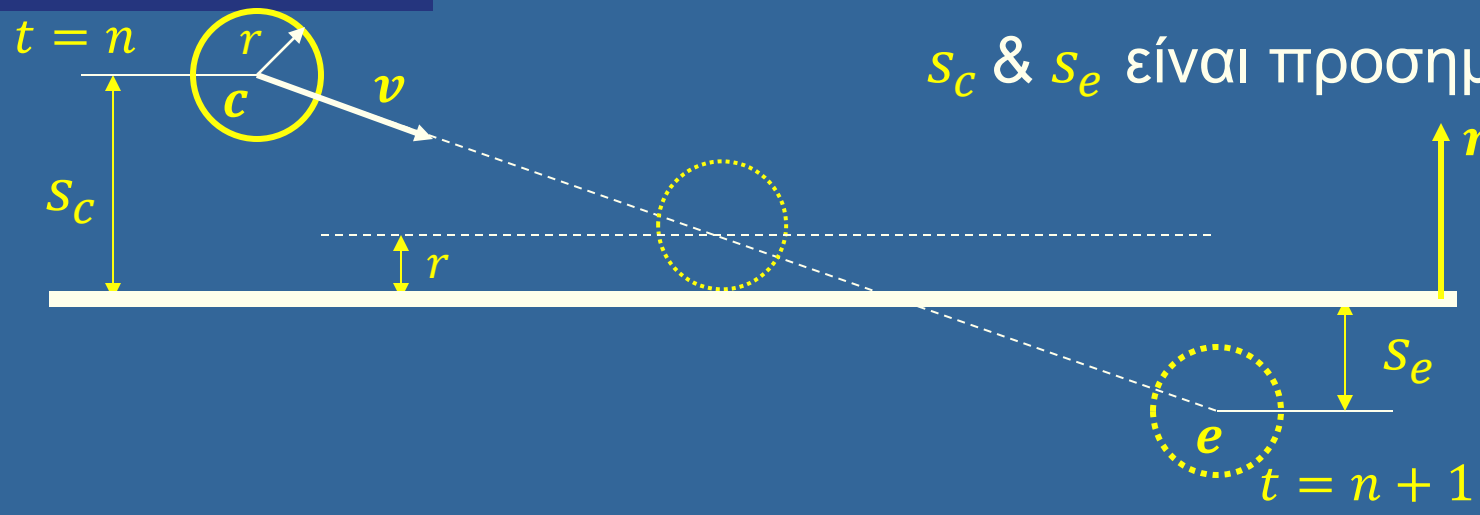
Δυναμικός Έλεγχος Τομής

- Ο έλεγχος γίνεται σε κάθε frame (διακριτός χρόνος)
- Προκύπτουν «κβαντικά» φαινόμενα



- Ο δυναμικός έλεγχος το λύνει
  - Ασχολείται με το χρονικό διάστημα μεταξύ των δύο frame
  - Πιο ακριβό

Παράδειγμα: Δυναμικός Έλεγχος Σφαίρας/Επιπέδου



s_c & s_e είναι προσημασμένες αποστάσεις

- Δεν υπάρχει σύγκρουση:
  - Αν είναι από την ίδια πλευρά του επιπέδου ($s_c s_e > 0$)
  - Επιπλέον: $|s_c| > r$ και $|s_e| > r$
- Διαφορετικά, η σφαίρα θα μετακινηθεί κατά $|s_c| - r$
- Χρόνος σύγκρουσης: $t_{cd} = n + \frac{s_c - r}{s_c - s_e}$
- Απόκριση: ανάκλαση $\boldsymbol{v}$ γύρω από το $\boldsymbol{n}$, και μετακίνηση κατά $(1 - t_{cd})r$

Έλεγχος Τομής: Περιορισμοί

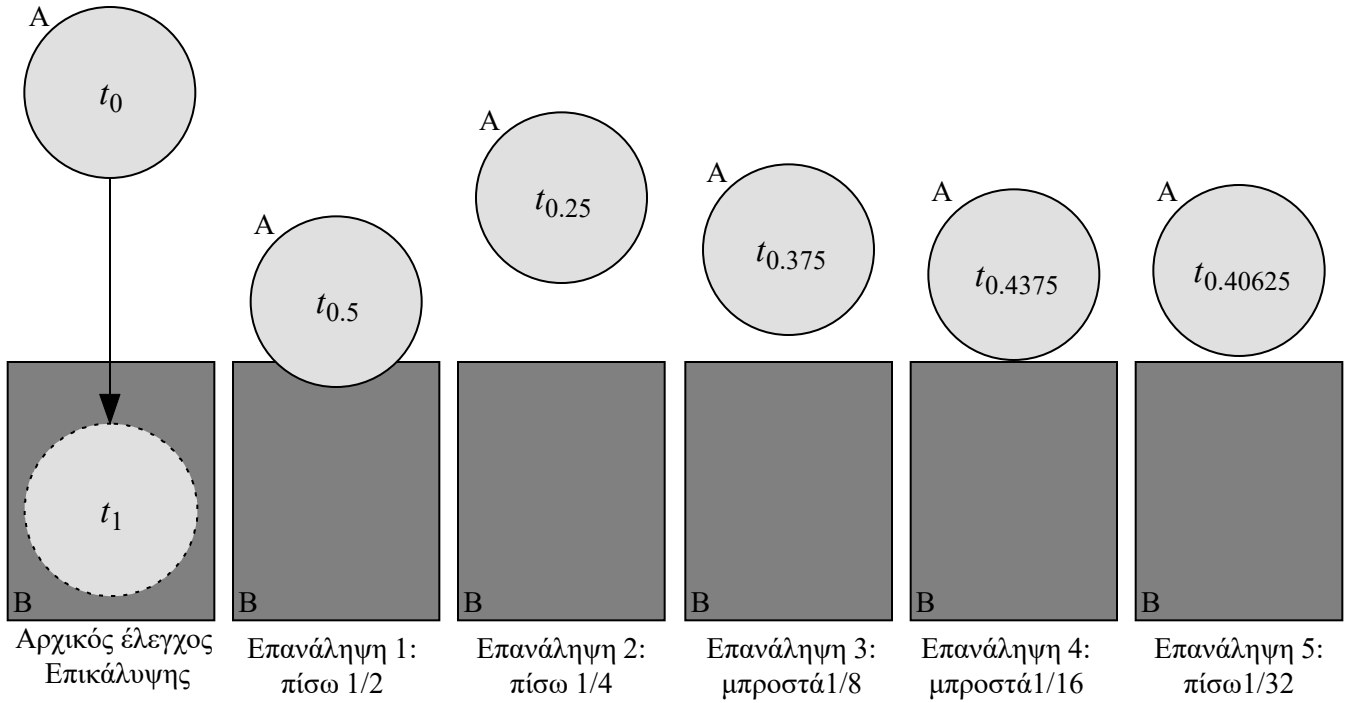
- Μεγαλύτερο κόστος από έλεγχος επικάλυψης (σε λίγο)
- Θέμα με τα διαδικτυακά παιχνίδια
  - Οι μελλοντικές προβλέψεις βασίζονται στη διαμόρφωση του κόσμου στο παρόν
  - Λόγω καθυστέρησης των πακέτων η διαμόρφωση δεν είναι πάντα συνεπής
- Υποθέτει σταθερή ταχύτητα και μηδενική επιτάχυνση στη διάρκεια ενός βήματος προσομοίωσης

Έλεγχος Επικάλυψης

Έλεγχος Επικάλυψης: Στιγμή Σύγκρουσης

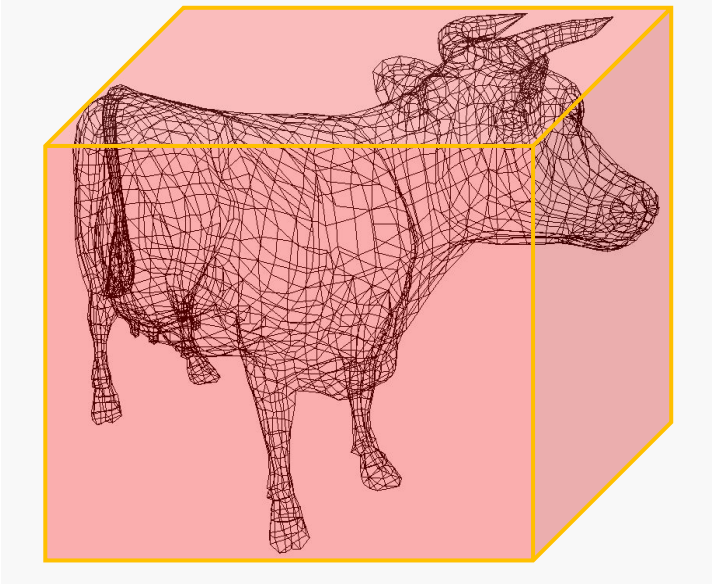
Ο χρόνος σύγκρουσης υπολογίζεται μετακινώντας το αντικείμενο πίσω στο χρόνο μέχρι ακριβώς πριν από τη σύγκρουση

- Μέθοδος διχοτόμησης



Θέματα Πολυπλοκότητας

Μεσοσκοπική Άποψη

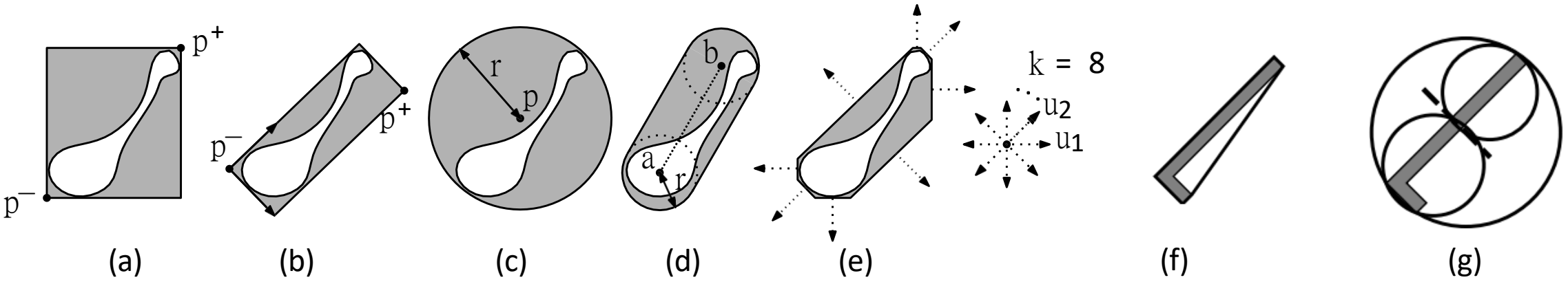


1. Η πολύπλοκη γεωμετρία πρέπει να απλοποιηθεί για να γίνει αποδοτικά ο έλεγχος επικάλυψης
2. Μείωση του πλήθους των ζευγών αντικειμένων προς έλεγχο

- Ο οριοθετημένος όγκος είναι ένα απλό γεωμετρικό σχήμα
  - Ενθυλακώνει πλήρως το αντικείμενο
  - Εάν δεν υπάρχει σύγκρουση με τον οριοθετημένο όγκο δεν απαιτείται άλλη δοκιμή

- Κοινοί οριοθετημένοι όγκοι
  - Σφαίρα
  - Κουτί

Άλλα Κοινά Σχήματα



(a) Ορθοκανονικά κουτιά

(b) Γενικά κουτιά

(c) Σφαίρες (ελλειψοειδή)

(d) Κάψουλα

(e) *k*-DOPs (*k*-Discrete Oriented Polytopes)

(f) Κυρτό περίβλημα

(g) Ιεραρχία οριοθετημένων όγκων

Colliders

Η καλή προσέγγιση πρέπει να είναι:

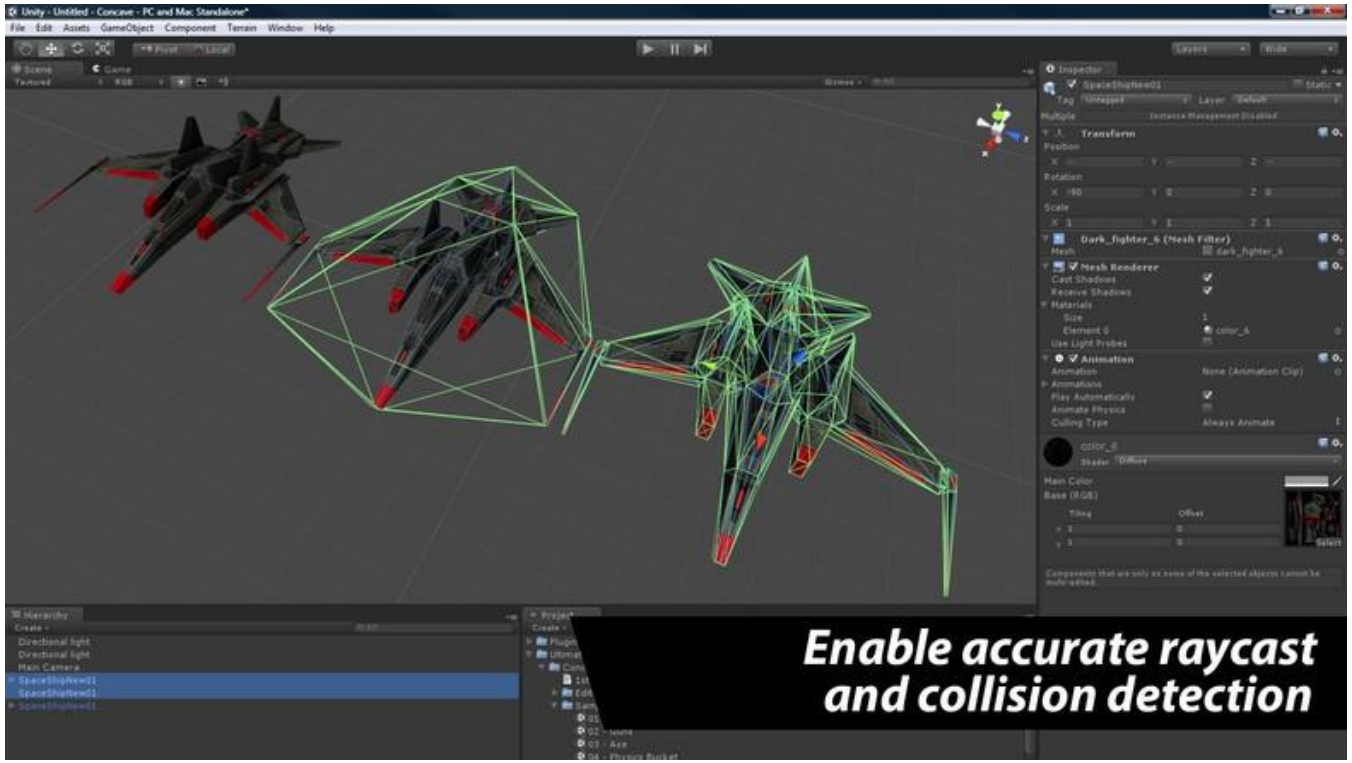
- Αρκετά ακριβής
- Γρήγορη

Αν είναι ανακριβής:

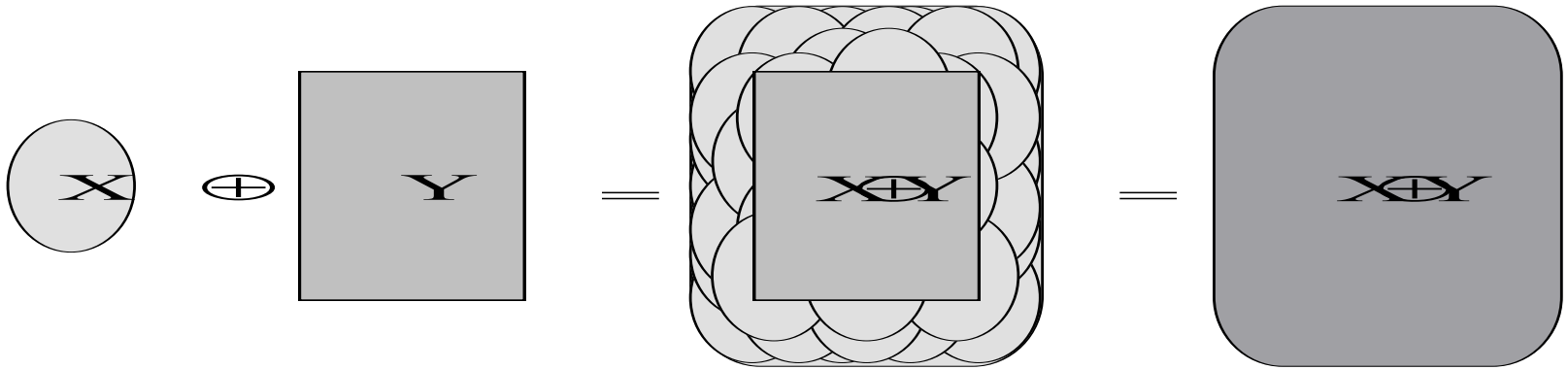
- Συγκρούσεις φαντασμάτων
  - Το σχήμα οριοθέτησης είναι πολύ μεγάλο, σηματοδοτεί ψευδή σύγκρουση
- Κακή φυσική
  - Σημείο σύγκρουσης σε λάθος σημείο-γωνία

Αν είναι υπερβολικά ακριβής:

- Πολύ αργό

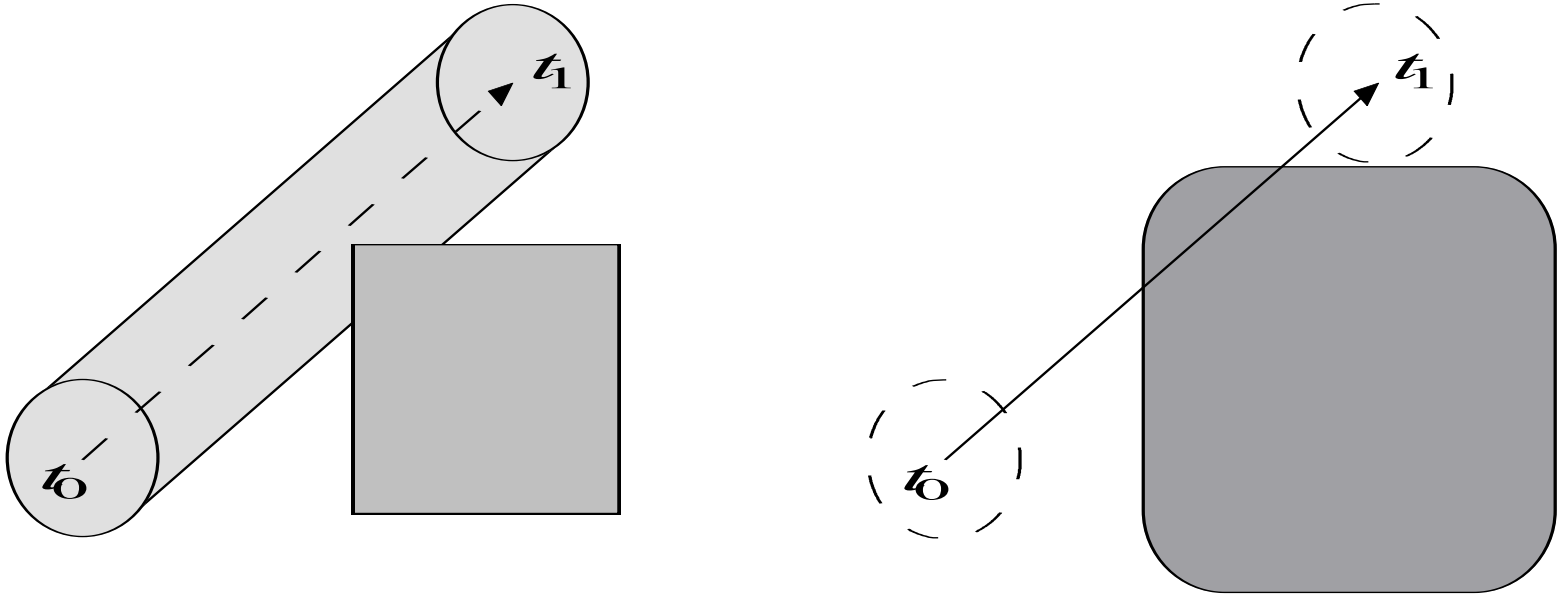


Μειώνοντας την Πολυπλοκότητα: Άθροισμα Minkowski



Άθροισμα Minkowski

Ο έλεγχος μεταξύ σχημάτων μετατρέπεται σε έλεγχο μεταξύ σημείου και σχήματος.

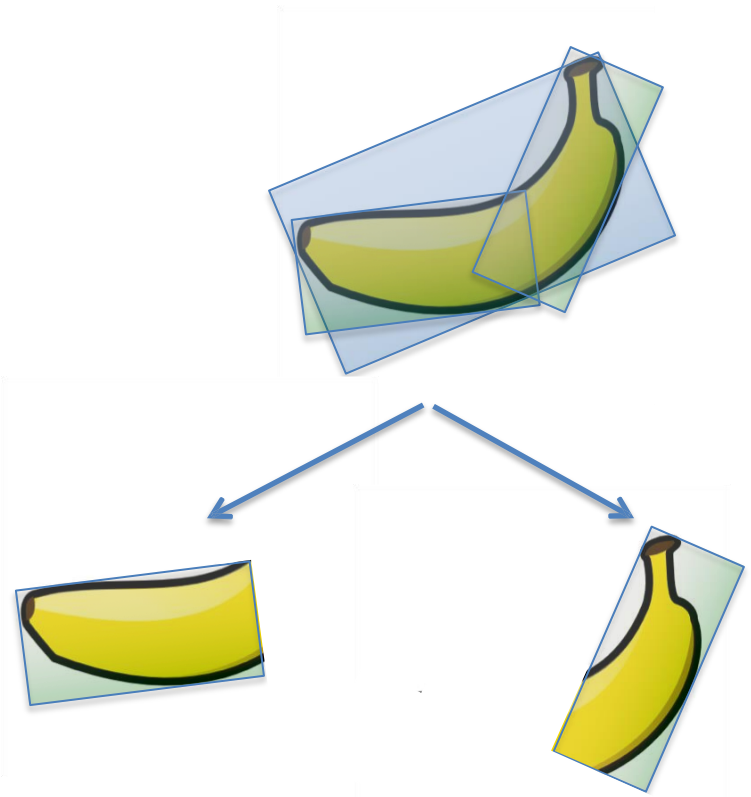


Ιεραρχία Οριοθετημένων Όγκων

(Μακροσκοπική Άποψη)

Οι οριοθετημένοι όγκοι είναι πολλοί. Τα αντικείμενα επίσης μπορούν να έχουν πολλούς οριοθετημένους όγκους. Πώς θα γίνει αποδοτικά η σύγκριση;

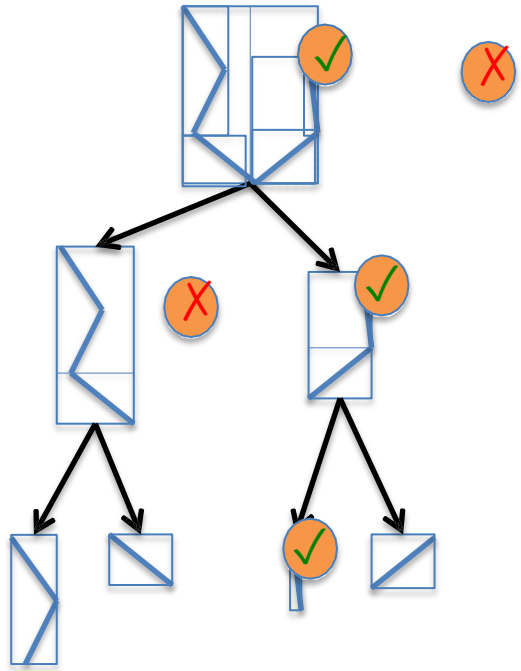
- Divide & Conquer
- Ιεραρχική Δομή (BVH)
  - Η ρίζα ενθυλακώνει πλήρως το αντικείμενο
  - Τα παιδιά αντιστοιχούν σε όγκους που το ενθυλακώνουν καλύτερα



Ανίχνευση Σύγκρουσης βασισμένη σε BVH

Δοθέντων δύο BVH

- Αν οι ρίζες δεν επικαλύπτονται
  - Επέστρεψε **Ψευδές**
- Αλλιώς (μπορεί να επικαλύπτονται)
  - Έλεγξε αναδρομικά όλα τα ζεύγη παιδιών

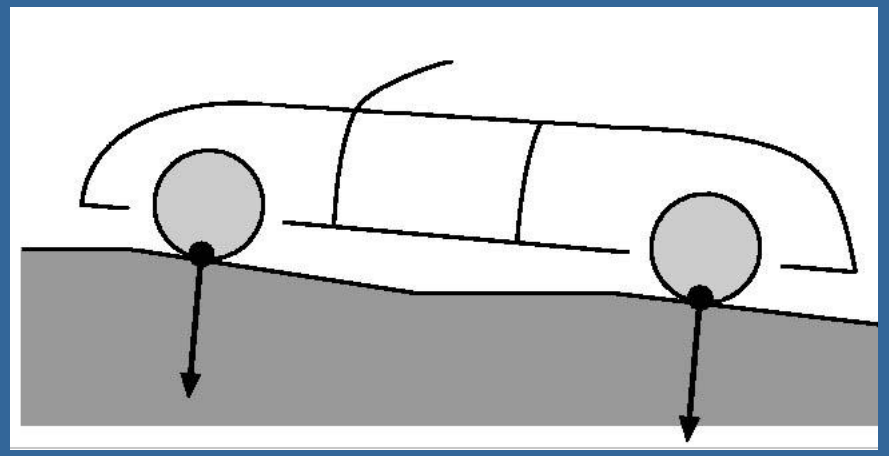


(Το δεύτερο σχήμα είναι απλά μία σφαίρα)

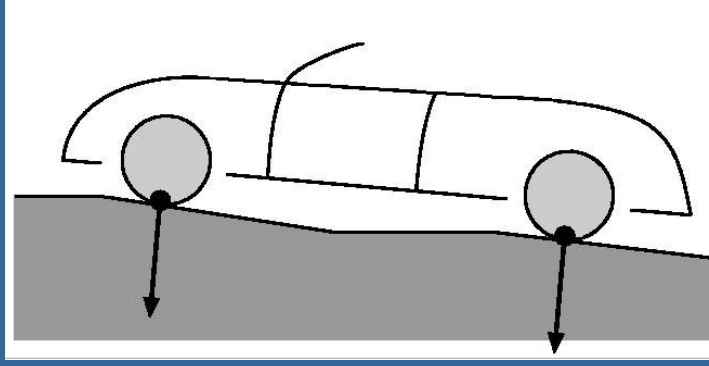
Τί θα Χρησιμοποιούσατε για Ανίχνευση Σύγκρουσης;

Ένα Παράδειγμα: Ανίχνευση Σύγκρουσης με Ακτίνες

- Έστω ότι ένα αυτοκίνητο προχωρά σε δρόμο με κλίση προς τα πάνω (ανηφόρα).
  - Πώς γίνεται ο έλεγχος σύγκρουσης ρόδας και εδάφους;
- Ένας τρόπος: Δοκιμή όλων των τριγώνων όλων των τροχών σε σχέση με τη γεωμετρία του δρόμου
- Για ορισμένες εφαρμογές, μπορούμε να προσεγγίσουμε και να έχουμε καλό αποτέλεσμα
  - Ιδέα: προσεγγίστε ένα σύνθετο αντικείμενο με ένα σύνολο ακτίνων



Ανίχνευση Σύγκρουσης με Ακτίνες



- Βάλτε μια ακτίνα σε κάθε τροχό
- Υπολογίστε την πλησιέστερη απόσταση τομής d, μεταξύ ακτίνας και γεωμετρίας εδάφους
  - Αν $d = 0$, τότε το αυτοκίνητο είναι στο δρόμο
  - Αν $d > 0$, τότε το αυτοκίνητο πετά πάνω από το δρόμο
  - Αν $d < 0$, τότε το αυτοκίνητο είναι βαθιά το δρόμο
- Χρησιμοποιήστε τις τιμές του d για να υπολογίσετε μια απλή απόκριση σύγκρουσης

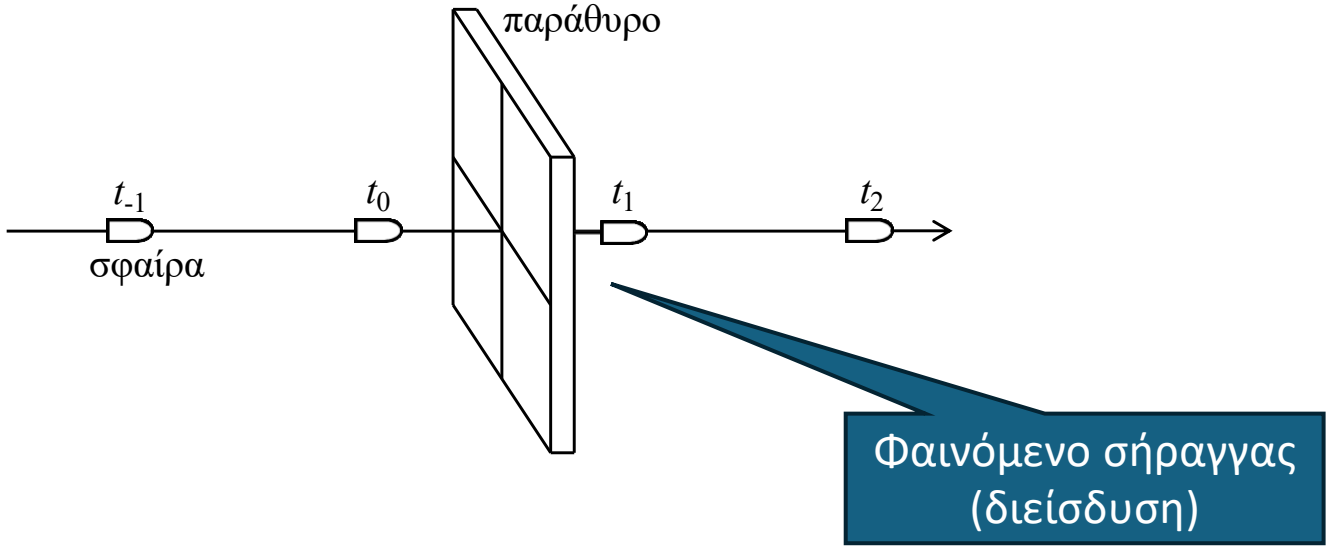
Το Έδαφος;;;;

- Έχουμε απλοποιήσει το αυτοκίνητο, αλλά όχι το δρόμο
- Χρήση χωρικών δομών δεδομένων για το δρόμο
  - BVH ή BSP
- Η απόσταση κατά μήκος της ακτίνας μπορεί να είναι αρνητική
  - Αναζήτηση προς θετική και αρνητική κατεύθυνση
  - Ή μετακινήστε την ακτίνα προς τα πίσω, έως ότου βρίσκεται εκτός του BV της γεωμετρίας του δρόμου

Όρια της Μεθόδου

Αποτυγχάνει με αντικείμενα που κινούνται γρήγορα

- Απίθανο να εντοπίσει την παρακάτω επικάλυψη σε κάποιο frame

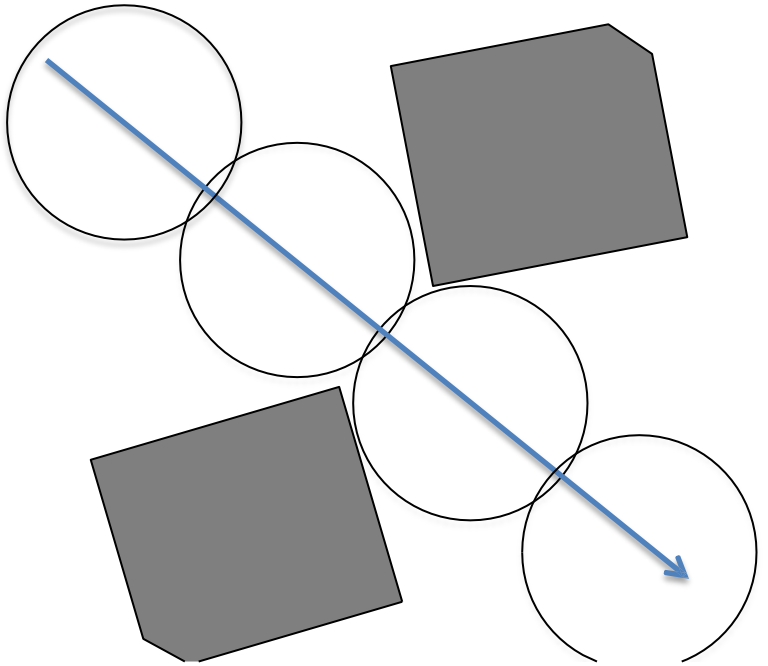


Σφάλματα (Glitches)

- Παίκτες/αντικείμενα που πέφτουν έξω από τον εικονικό κόσμο
- Βλήματα που περνούν μέσα από στόχους
- Οι παίκτες πάνε εκεί που δεν πρέπει

Προέρχονται από σφάλματα στην ανίχνευση συγκρούσεων

- Δύσκολο να αποφευχθεί λόγω της διακριτής κίνησης



Πιθανές Λύσεις

- Σχεδιαστικοί περιορισμοί στην ταχύτητα των αντικειμένων
  - Δεν είναι πάντοτε εφικτό (π.χ., σφαίρες)
- Μείωση του βήματος προσομοίωσης
  - Περιορισμοί από το υλικό
- Έλεγχος τομής (το είδαμε)

Απόκριση Σύγκρουσης

Απόκριση Σύγκρουσης

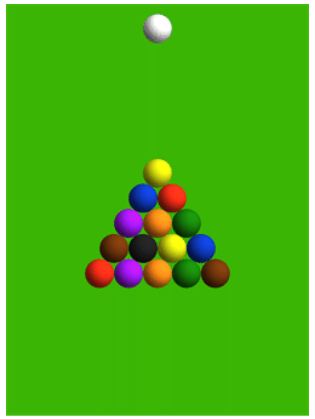
Τρία στάδια:

1. Έλεγξε αν η σύγκρουση μπορεί να αγνοηθεί
   - Κάποια γεγονότα μπορεί να προκληθούν (ηχητικά εφέ, μηνύματα ειδοποίησης σύγκρουσης)

2. Τοποθέτηση των αντικειμένων στη θέση σύγκρουσης. Ανάθεση νέων ταχυτήτων στα αντικείμενα με χρήση:
   - Φυσικών νόμων
   - Διανυσματικών μαθηματικών
   - Άλλης λογικής

3. Διάδοση αποτελεσμάτων σύγκρουσης. Πιθανά αποτελέσματα:
   - Καταστροφή κάποιου αντικειμένου
   - Ηχητικά εφέ
   - Απώλεια ζωής

Παραδείγματα



Κρούση δύο σφαιρών μπιλιάρδου

- Υπολογίστε τις θέσεις των σφαιρών τη στιγμή της πρόσκρουσης
- Δώστε νέες ταχύτητες στις μπάλες
- Παίξτε ηχητικό εφέ «κρούσης»

Η βόμβα χτυπά τους βράχους

- Η βόμβα εξαφανίζεται
- Ηχητικό εφέ έκρηξης
- Ο βράχος καταστρέφεται και γίνεται ζημιά στην τριγύρω περιοχή και σε όσους χαρακτήρες βρίσκονται εντός της

Ο χαρακτήρας διασχίζει τον τοίχο

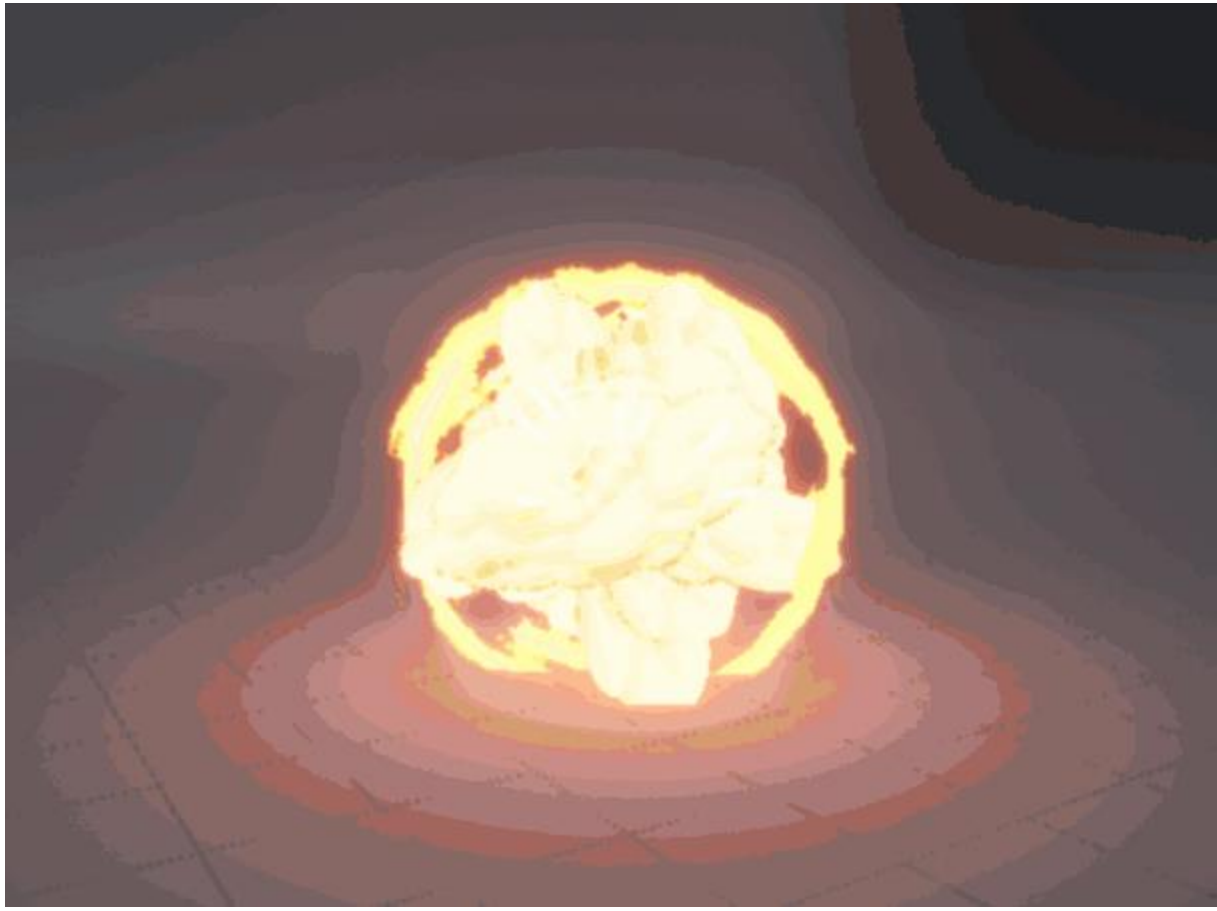
- «Μαγικό» ηχητικό εφέ
- Δεν επηρεάζονται τροχιές ή ταχύτητες
- Ώχ!!! ξέχασα τον collider. Τελικά σπάει τα μούτρα του ☺

Ανακεφαλαιώνοντας

- Εντοπίστε ότι έχει συμβεί σύγκρουση (οριοθετημένοι όγκοι)
- Προσδιορισμός του χρόνου σύγκρουσης (μπορεί να χρειαστεί να γυρίσουμε το χρόνο πίσω στο σημείο σύγκρουσης)
- Προσδιορισμός της θέσης των αντικειμένων όταν αγγίζονται
- Προσδιορισμός του κάθετου διανύσματος σύγκρουσης (γωνία πρόσπτωσης = γωνία ανάκλασης)
- Προσδιορισμός του διανύσματος ταχύτητας μετά τη σύγκρουση
- Προσδιορισμός τυχόν αλλαγών στην κίνηση

Συστήματα Σωματιδίων για Έκρηξη

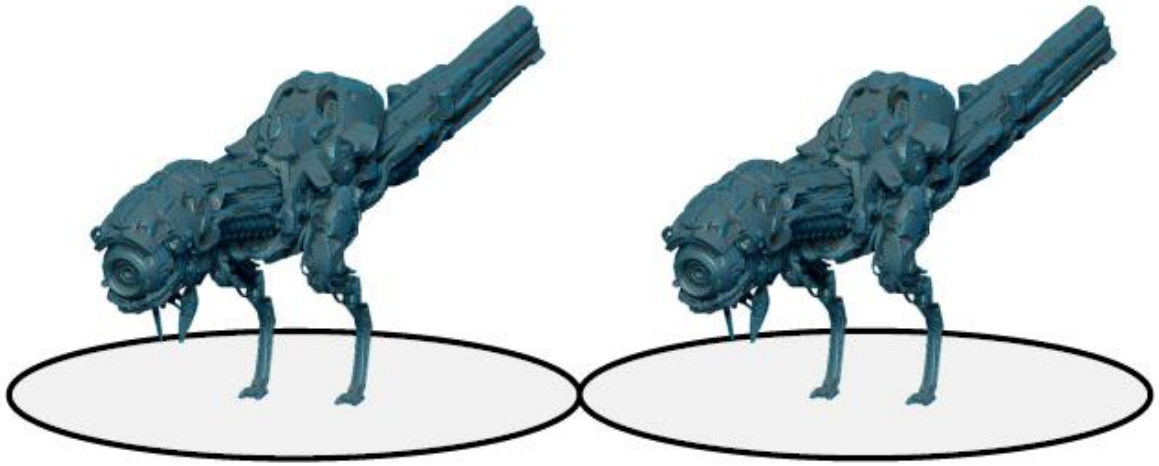
- Εκκίνηση με πολλά μικρά αντικείμενα (1 έως 4 pixel)
- Αρχικοποίηση σωματιδίων με τυχαίες ταχύτητες με βάση την ταχύτητα του αντικειμένου που εκρήγνυται
- Εφαρμογή βαρύτητας
- Μεταβολή έντασης χρώματος ως συνάρτηση του χρόνου
- Καταστροφή αντικειμένων όταν συγκρούονται ή μετά από καθορισμένο χρονικό διάστημα



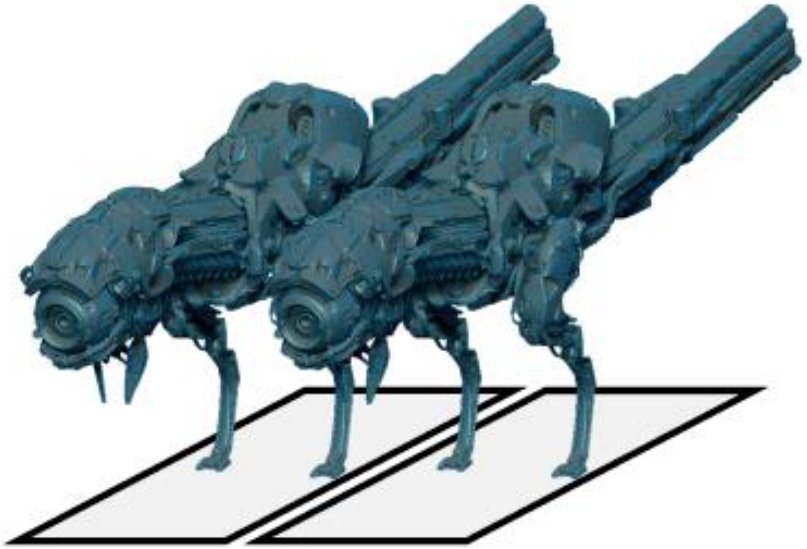
Αποφυγή Εμποδίων

Θα μπορούσαμε να το συζητήσουμε και στην AI διάλεξη

Αποφυγή Εμποδίων



- Παραδοσιακά τα εμπόδια αναπαρίστανται με κύκλους
  - Είναι αναλλοίωτα ως προς την περιστροφή
  - Ανεπαρκή για επιμηκυμένα εμπόδια όπως τα διπλανά robot
  - Μεγάλες αποστάσεις μεταξύ τους



- Λύση στο Zero Dawn:
  - Αναπαράσταση με ορθογώνια
  - Χρήση της τεχνικής Velocity Obstacles (VO)

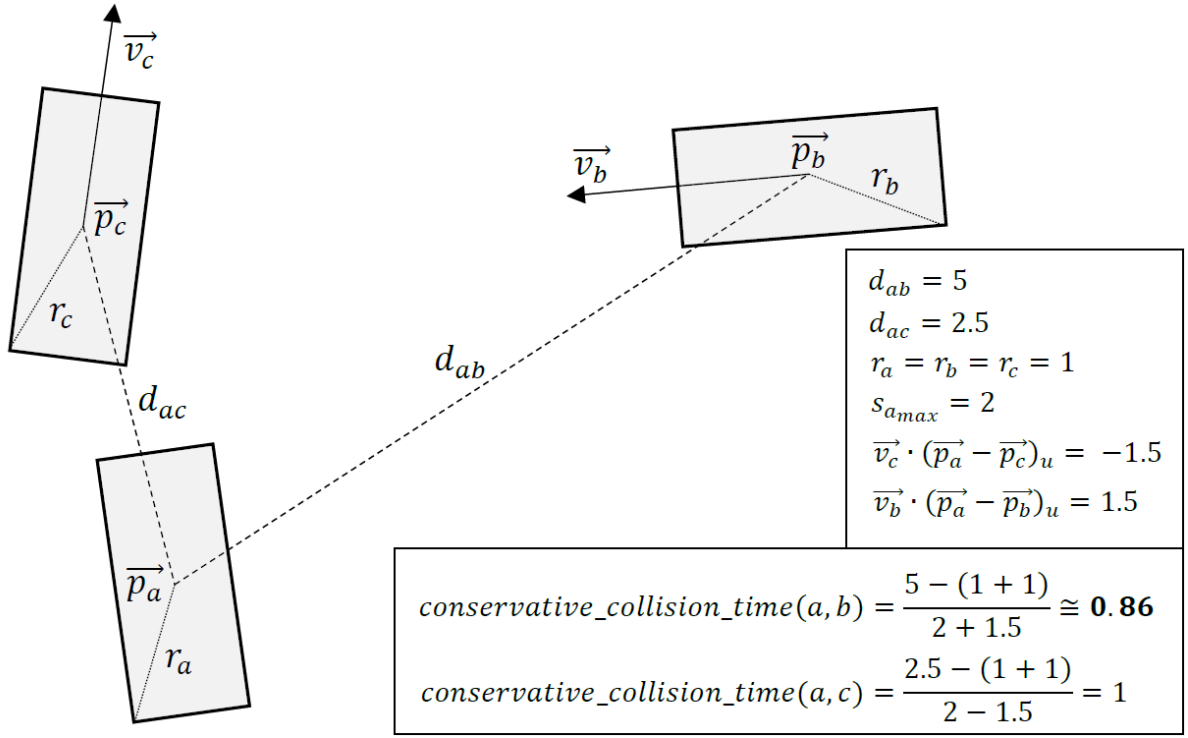
Αποφυγή Εμποδίων: Βήματα

Για κάθε robot κατασκεύασε τα VO

- Περιόρισε το πλήθος σε 5
- Αυτά επιλέγονται με βάση μία μετρική εκτιμώμενου χρόνου σύγκρουσης

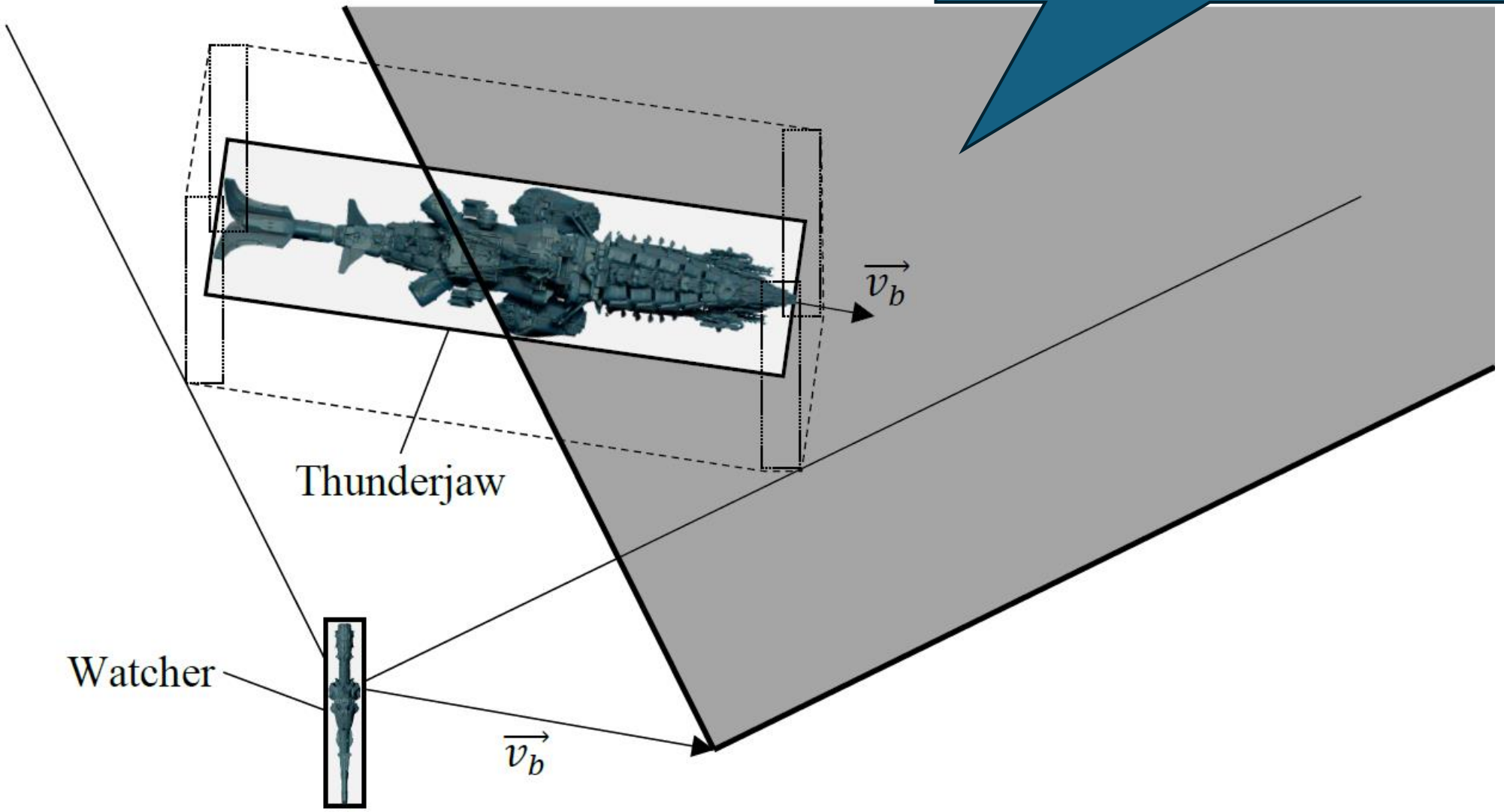
Σε 3 περιπτώσεις δεν λαμβάνονται υπόψη:

- Το εμπόδιο είναι πίσω μας (τόξο 120^o πίσω)
- Το εμπόδιο έχει μικρότερη προτεραιότητα αποφυγής (τα μεγάλα robot δεν κάνουν στην άκρη για τα μικρότερα)
- Το εμπόδιο είναι πέρα από τον προορισμό μας

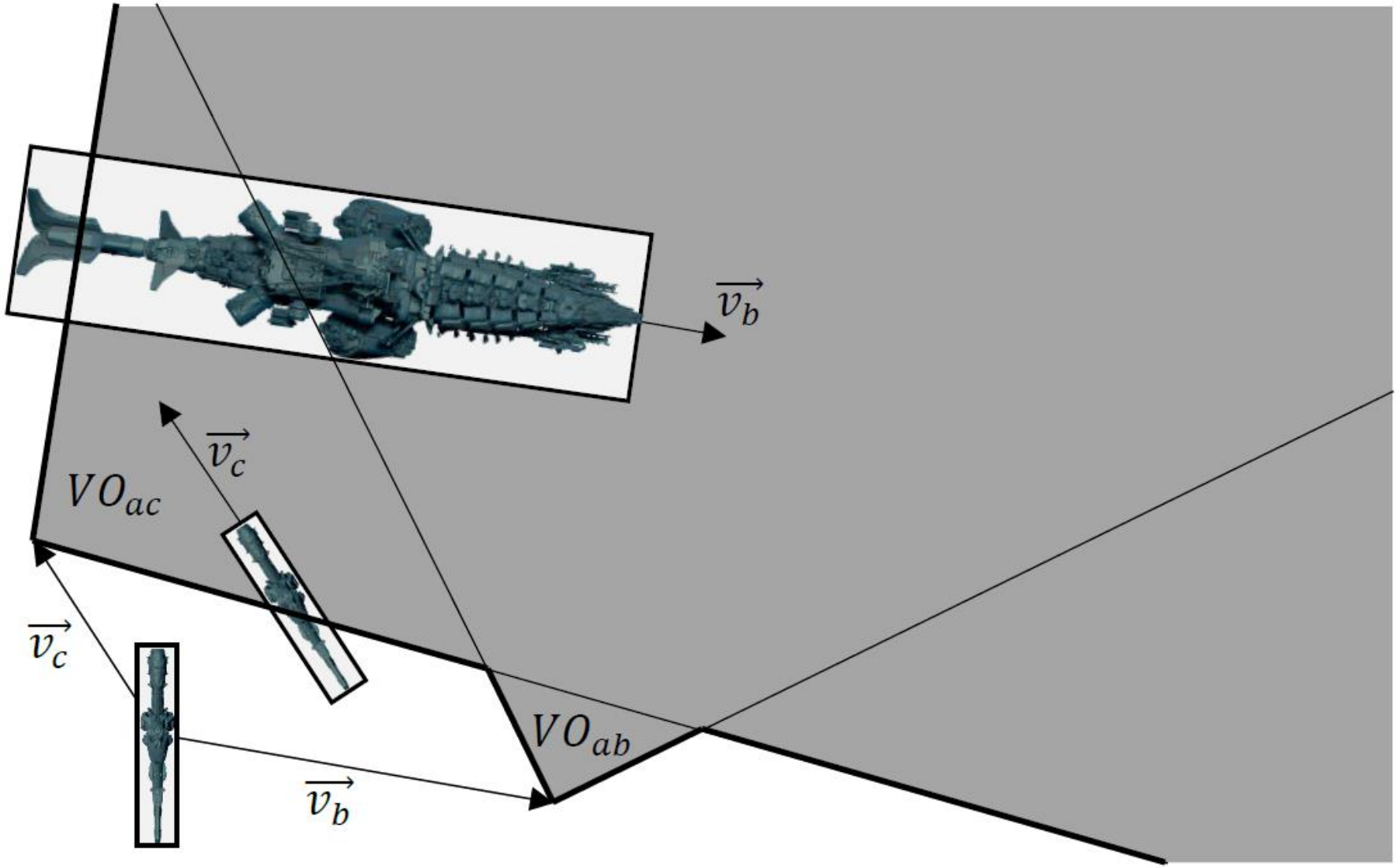


Velocity Obstacles

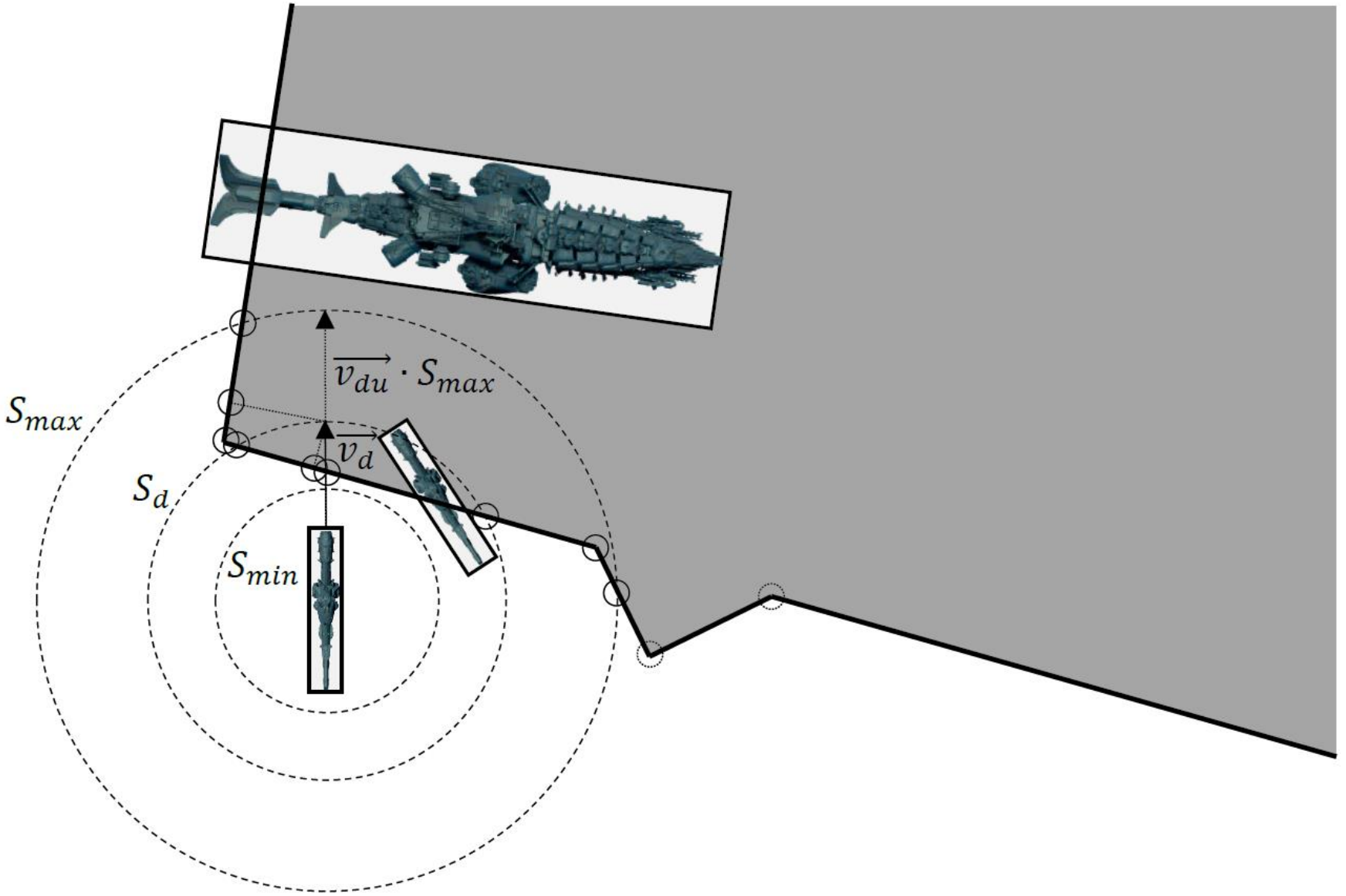
Αν το διάνυσμα ταχύτητας του Watcher πέσει μέσα στην σκιασμένη περιοχή τότε έχουμε πιθανή σύγκρουση



Συνδυάζοντας VO

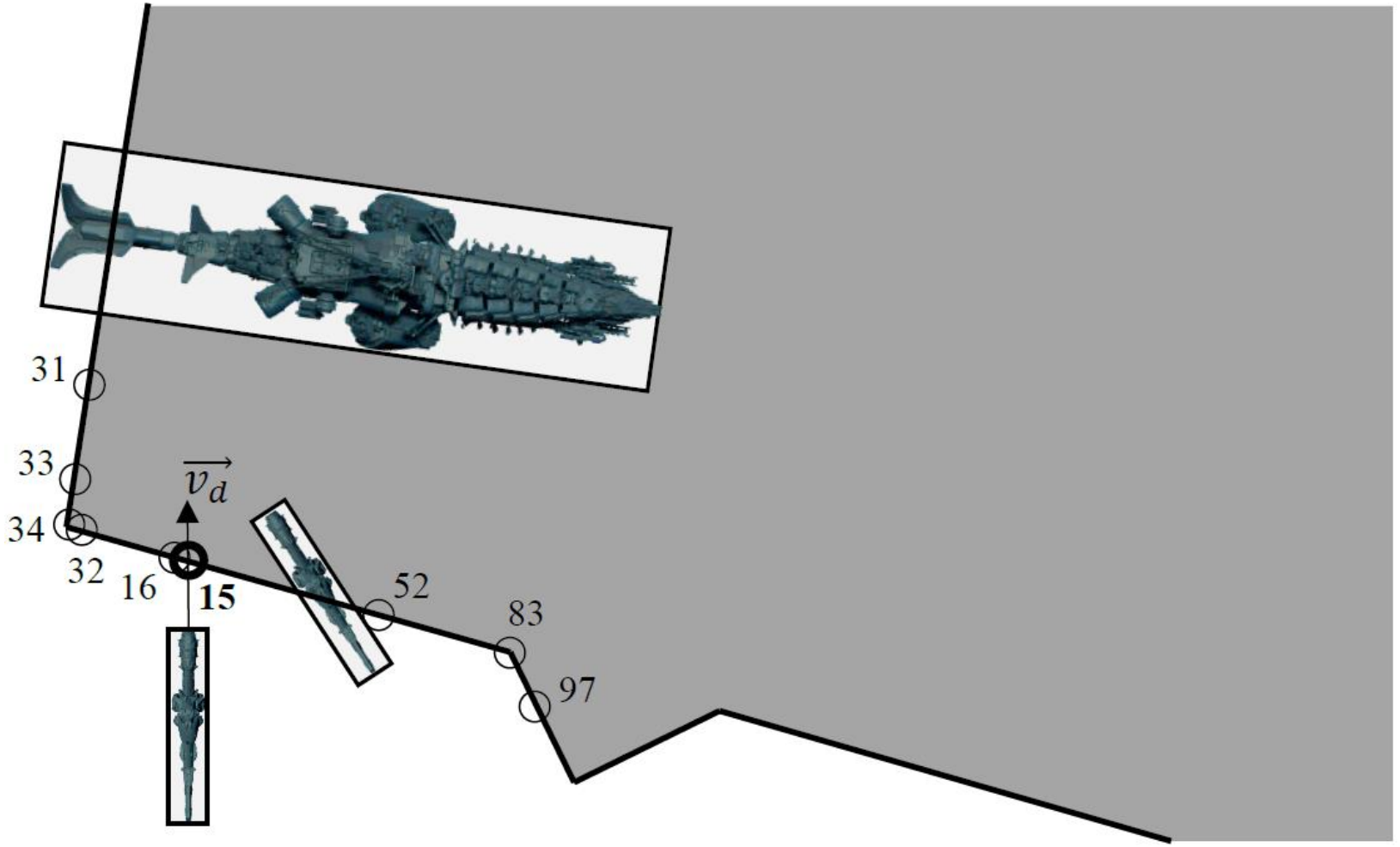


Υπολογισμός Νέου Διανύσματος Ταχύτητας



Βαθμολογώντας τις Διαφορετικές Ταχύτητες

Συνάρτηση βαθμολόγησης: $e = (1 - p_0)(400a_d + 50\beta_d + 70\gamma) + 0.5(1 - p_t)(400a_c + 50\beta_c)$



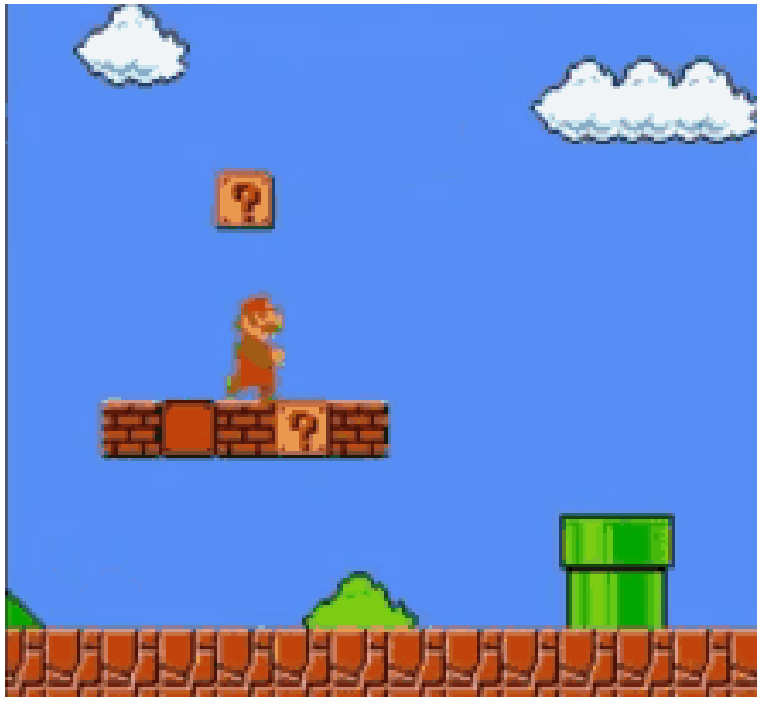
Κάποια Επιπλέον Σημεία

- Οι ταχύτητες είναι μέσοι όροι από τα frames του τελευταίου μισού δευτερόλεπτου
- Οι άνθρωποι μένουν στάσιμοι όταν δεν βρίσκουν κάποιο διάνυσμα ταχύτητας
- Μερικές φορές οι άνθρωποι έκαναν στροφές 360^o – αντιμετωπίστηκε με ανίχνευση (οι πολίτες μένουν στάσιμοι)
- Κάποια robot και οι άνθρωποι χρησιμοποιούσαν κύκλους για αποφυγή σύγκρουσης. Όταν γινόταν έλεγχος μεταξύ κύκλου και ορθογωνίου το πρώτο μετατρεπόταν σε ορθογώνιο.
- Σε σχέση με τη στατική γεωμετρία, υπήρχαν κάποια προβλήματα που δεν τα έλυσαν (όπως όταν ένας άνθρωπος κολλάει σε κάποιο τοίχο) μιας και ήταν σπάνια.
- Κόστος: 60 μs για κάθε οντότητα σε περιοχές με πολλά robot στο PS4. Αυτό αντιστοιχεί στο 1/3 του χρόνου της CPU για κίνηση χωρίς το animation.
- Η συγκεκριμένη προσέγγιση δεν ήταν τόσο ρεαλιστική για πλήθος ανθρώπων στις πόλεις (ήταν κάπως μηχανικό).

Μετακίνηση Σωμάτων

Η Μετακίνηση στα Παιχνίδια

- Κινηματική
  - Κίνηση σωμάτων χωρίς δυνάμεις, τριβή, επιτάχυνση,…
  - Δεν είναι ρεαλιστικό αλλά δεν σημαίνει ότι δεν είναι διασκεδαστικό…
- Δυναμική
  - Αλληλεπίδραση με δυνάμεις και ροπές



Σημείωση: Δύναμη (Force) και Ώθηση (Impulse)

Δύναμη

- Αλλάζει την ταχύτητα κατά force/mass*Time.fixedDeltaTime
  - Εφαρμόζεται σε πολλά frames
  - Αλλάζει σταδιακά την ορμή

Παράδειγμα: Βαρύτητα

Ώθηση

- Αλλάζει την ταχύτητα κατά force/mass
  - Εφαρμόζεται σε ένα frame
  - Αλλάζει στιγμιαία την ορμή (δεν έχει φυσικό αντίστοιχο εκτός από την περίπτωση απότομων μεταβολών σε μεγάλη χρονική κλίμακα)

Παράδειγμα: Άλμα

Φυσική Σωματιδίων

(βασικά σημειακών μαζών)

- Τι είναι ένα σωματίδιο;
  - Μια σφαίρα πεπερασμένης ακτίνας με τέλεια λεία επιφάνεια χωρίς τριβές
  - Χωρίς περιστροφική κίνηση

- Σωματιδιακή Κινηματική
  - Ορίζει τις βασικές ιδιότητες της κίνησης των σωματιδίων
  - Θέση, Ταχύτητα, Επιτάχυνση

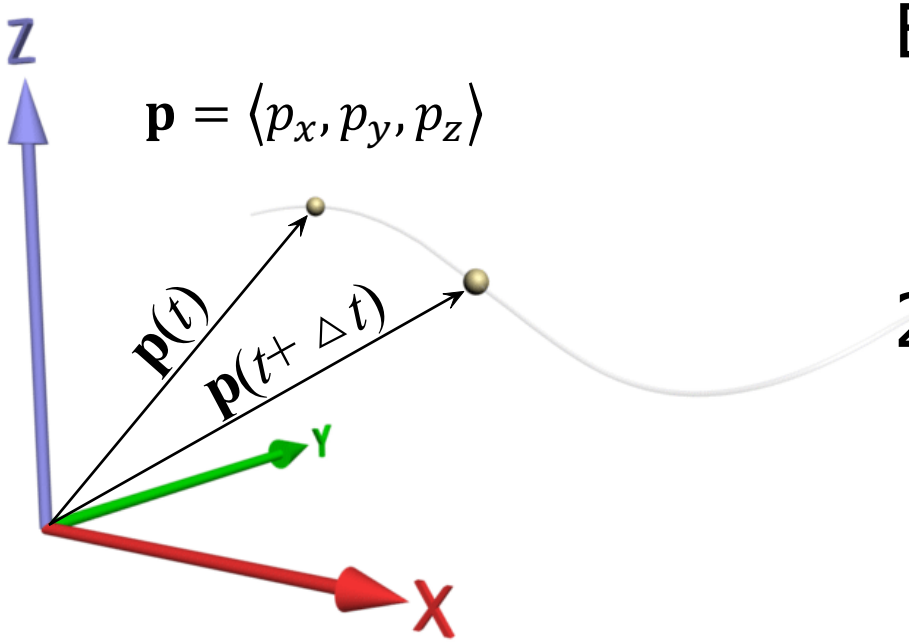
Θέση – Ταχύτητα – Επιτάχυνση

Και κάτι ακόμα…

Ταχύτητα: $\mathbf{V}(t) = \lim_{\Delta t \to 0} \frac{\mathbf{p}(t+\Delta t) - \mathbf{p}(t)}{\Delta t} = \frac{d}{dt}\mathbf{p}(t)$

Επιτάχυνση: $\mathbf{a}(t) = \frac{d}{dt}\mathbf{V}(t) = \frac{d^2}{dt^2}\mathbf{p}(t)$

2ος νόμος Νεύτωνα: $\mathbf{F}(t) = m\mathbf{a}(t)$



Η θέση μεταβάλλεται στο χρόνο

Προσομοίωση Φυσικής

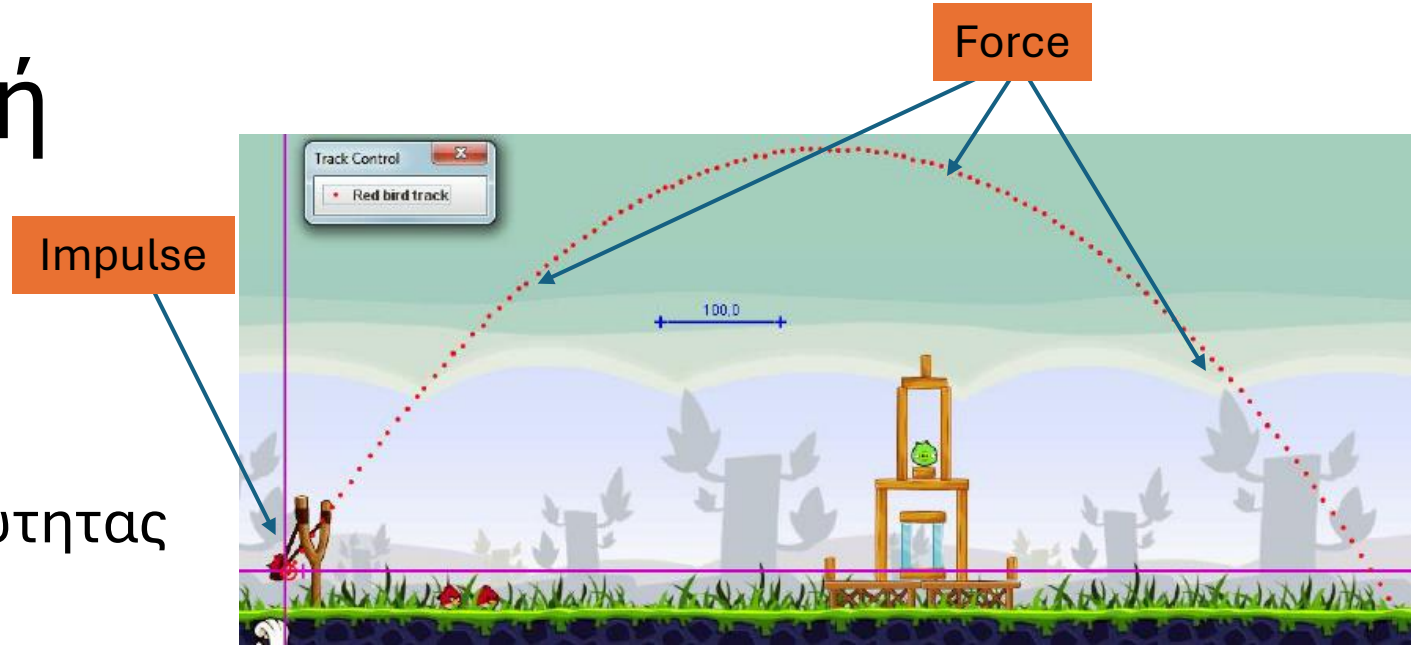
Ο κύκλος της κίνησης:

- Η δύναμη $\boldsymbol{F}(t)$ προκαλεί επιτάχυνση
- Η επιτάχυνση $\boldsymbol{a}(t)$ προκαλεί αλλαγή στην ταχύτητα
- Η ταχύτητα $\boldsymbol{V}(t)$ προκαλεί αλλαγή στη θέση

Προσομοίωση Φυσικής:

- Επίλυση των παραπάνω εξισώσεων για εξομοίωση του κύκλου κίνησης

Παράδειγμα: 2d βολή



- Σταθερή δύναμη
  - Βάρος βλήματος: $\boldsymbol{W = mg}$
  - $\boldsymbol{g}$ είναι η επιτάχυνση της βαρύτητας

- Εξίσωση κίνησης για το βλήμα:

$$\mathbf{V}(t) = \mathbf{V}_{init} + \mathbf{g}(t - t_{init})$$

$$\mathbf{p}(t) = \mathbf{p}_{init} + \mathbf{V}_{init}(t - t_{init}) + \frac{1}{2}\mathbf{g}(t - t_{init})^2$$

Αριθμητική Προσομοίωση

Οι εξισώσεις:

- Κλειστοί τύποι
- Έγκυροι και ακριβείς για τη διαμόρφωση του προβλήματος (σταθερό βάρος/δύναμη – όχι αντίσταση)
- Βάλε το χρόνο και πάρε τη θέση και την ταχύτητα

Δεν δουλεύει όμως:

- Οι δυνάμεις μεταβάλλονται
- Σπάνια υπάρχουν κλειστοί τύποι

Η αριθμητική προσομοίωση αφορά ένα σύνολο τεχνικών αυξητικής επίλυσης των εξισώσεων κίνησης όταν οι εφαρμοζόμενες δυνάμεις δεν είναι σταθερές και δεν υπάρχει λύση κλειστού τύπου.

Μέθοδοι Πεπερασμένων Διαφορών

- Η πιο κοινή οικογένεια αριθμητικών τεχνικών για προσομοίωση δυναμικής άκαμπτου σώματος
- Αυξητική λύση εξισώσεων κίνησης
- Προέρχεται από τους αρχικούς όρους του αναπτύγματος Taylor

- Αριθμητικός Ολοκληρωτής: Αποκαλείται έτσι μία εξίσωση πεπερασμένης διαφοράς που παράγει λύσεις για χρονικές στιγμές

Euler Ολοκληρωτής

$$\underbrace{\mathbf{S}(t+\Delta t)}_{\text{new state}} = \underbrace{\mathbf{S}(t)}_{\text{prior state}} + \Delta t \underbrace{\frac{d}{dt}\mathbf{S}(t)}_{\text{state derivative}}$$

- Οι ιδιότητες του αντικειμένου αποθηκεύονται στο διάνυσμα **S**
- Χρήση του παραπάνω ολοκληρωτή για αυξητική ενημέρωση του **S** (με βάση το χρόνο παιχνιδιού)
- Πρέπει να αποθηκεύει τις προηγούμενες τιμές του **S**
- Για τον Euler, μία επιλογή περιγραφής κατάστασης είναι:

$$\mathbf{S} = \langle m\mathbf{V}, \mathbf{p} \rangle$$
$$d\mathbf{S}/dt = \langle \mathbf{F}, \mathbf{V} \rangle$$

Παράδειγμα

$$\mathbf{S}(t+\Delta t) = \mathbf{S}(t) + \Delta t \frac{d}{dt}\mathbf{S}(t) = \begin{bmatrix} 19.2 \\ 0.0 \\ 72.5 \\ 10.0 \\ 0.0 \\ 2.0 \end{bmatrix} + \Delta t \begin{bmatrix} 0.0 \\ 0.0 \\ -24.53 \\ 7.68 \\ 0.0 \\ 29.0 \end{bmatrix}$$

$\Delta t = .2\ s$	$\Delta t = .1\ s$	$\Delta t = .01\ s$
$= \begin{bmatrix} 19.2025 \\ 0.0 \\ 67.5951 \\ 11.5362 \\ 0.0 \\ 7.8000 \end{bmatrix}$	$= \begin{bmatrix} 19.2025 \\ 0.0 \\ 72.0476 \\ 10.7681 \\ 0.0 \\ 4.9000 \end{bmatrix}$	$= \begin{bmatrix} 19.2025 \\ 0.0 \\ 72.2549 \\ 10.0768 \\ 0.0 \\ 2.2900 \end{bmatrix}$
$= \begin{bmatrix} 19.2 \\ 0.0 \\ 67.5951 \\ 11.5362 \\ 0.0 \\ 7.6038 \end{bmatrix}$	$= \begin{bmatrix} 19.2 \\ 0.0 \\ 72.0476 \\ 10.1536 \\ 0.0 \\ 4.8510 \end{bmatrix}$	$= \begin{bmatrix} 19.2 \\ 0.0 \\ 72.2549 \\ 10.0768 \\ 0.0 \\ 2.2895 \end{bmatrix}$

Exact, Closed-form Solution

$V_{\text{init}} = 30\text{m/s}$
Γωνία εκτόξευσης, $\varphi = 75{,}2^o$
Γωνία εκτόξευσης, $\theta = 0^o$ (κίνηση στο xz επίπεδο)
Μάζα βλήματος, $m = 2.5kg$
Στόχος στο $\langle 50,0,20 \rangle$ (μέτρα)

Time	Position (m)			Linear Momentum (kg-m/s)			Force (N)			Velocity (m/s)		
	p_x	p_y	p_z	mV_x	mV_y	mV_z	F_x	F_y	F_z	V_x	V_y	V_z
5.00	10.00	0.00	2.00	19.20	0.00	72.50	0.00	0.00	-24.53	7.68	0.00	29.00

t_{init} $\quad \mathbf{p}_{init}$ $\quad m\boldsymbol{V}_{init}$ $\quad \boldsymbol{F = Weight = m\mathbf{g}}$ $\quad \boldsymbol{V}_{init}$

$\boldsymbol{S} = \langle m\mathbf{V}_{init}, \mathbf{p}_{init} \rangle$ $\qquad$ $d\boldsymbol{S}/dt = \langle m\boldsymbol{g}, \boldsymbol{V}_{init} \rangle$

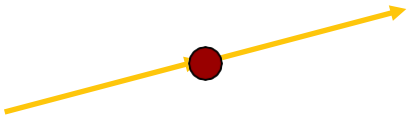
Πιο Γραφικά…

- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$

- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

Πιο Γραφικά…

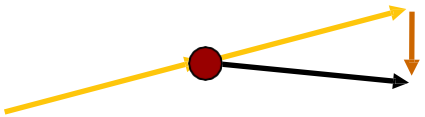
- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$



- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

Πιο Γραφικά…

- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$



- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

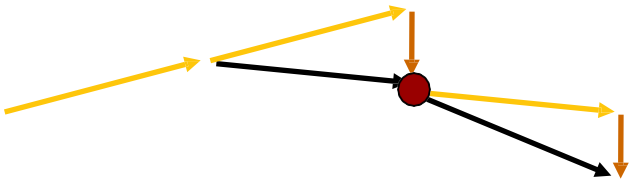
Πιο Γραφικά…

- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$

- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

Πιο Γραφικά…

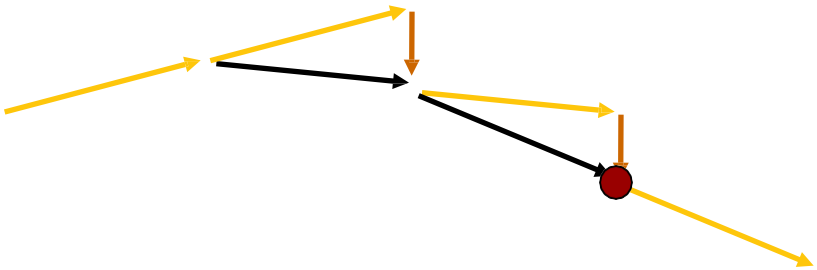
- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$



- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

Πιο Γραφικά…

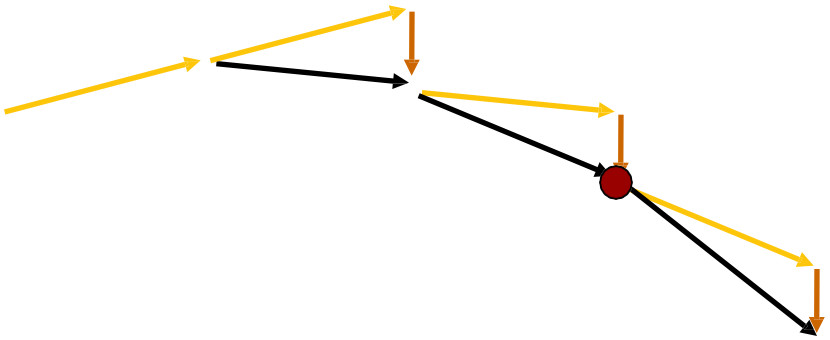
- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$



- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

Πιο Γραφικά…

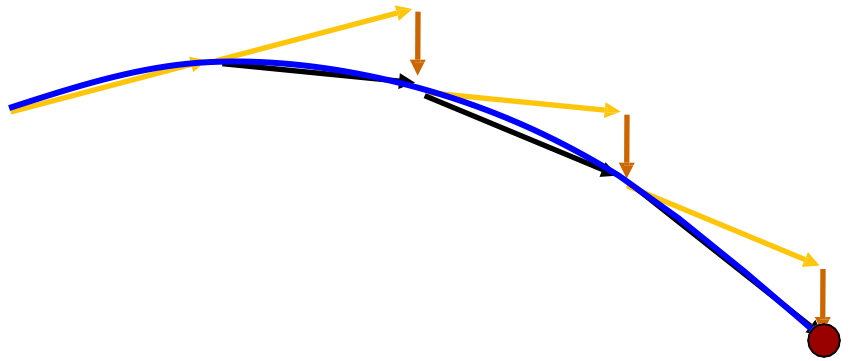
- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$



- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

Πιο Γραφικά…

- Διαφορική εξίσωση
  - $F(p,t) = m\ a(t)$
  - $F(p,t) = m\ p''(t)$



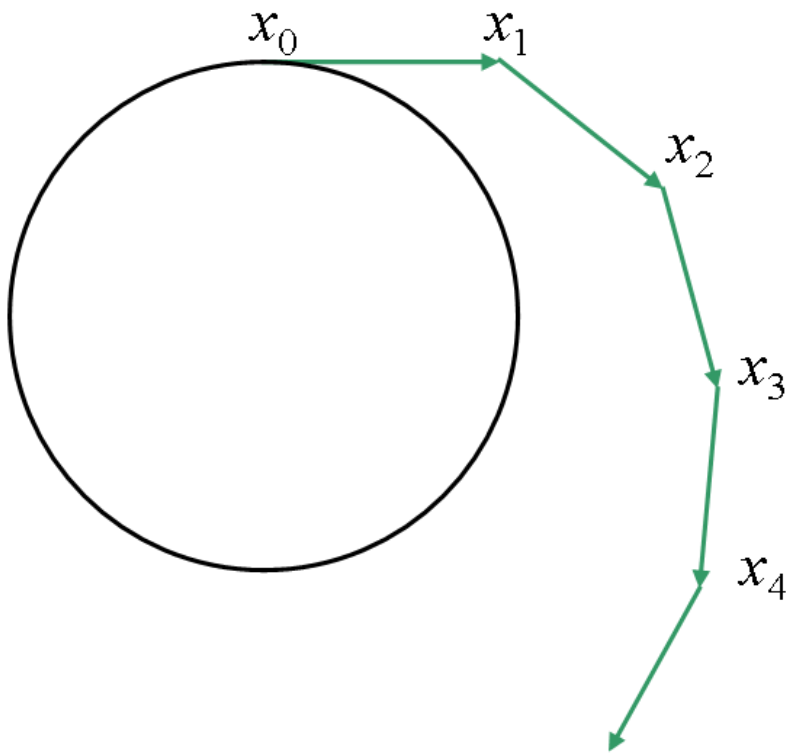
- Μέθοδος Euler:
  - $a_i = F(p_i, i\Delta t)/m$
  - $v_{i+1} = v_i + a_i \Delta t$
  - $p_{i+1} = p_i + v_i \Delta t$

Το Πρόβλημα

- **Συσσώρευση Σφαλμάτων**
  - Παρενέργεια της βηματικής προσέγγισης

- Ιδιαίτερο πρόβλημα στις *τροχιές*
  - Μετακίνηση κατά το διάνυσμα εφαπτομένης
  - Το διάνυσμα σε βγάζει εκτός τροχιάς
  - Χειροτερεύει με το χρόνο

- Πρέπει να *περιοριστεί* ώστε να κρατηθεί στην τροχιά



Αντιμετώπιση Σφαλμάτων

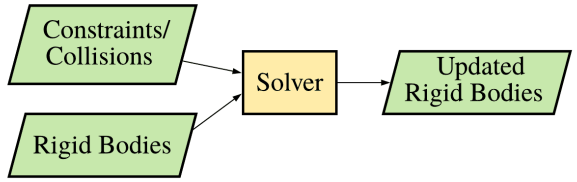
- Κλασσική λύση: μείωση του βήματος Δt
  - Αύξηση των fps (όχι απαραίτητα καλό)
  - Περισσότερα από ένα βήματα (επαναλήψεις) ανά frame

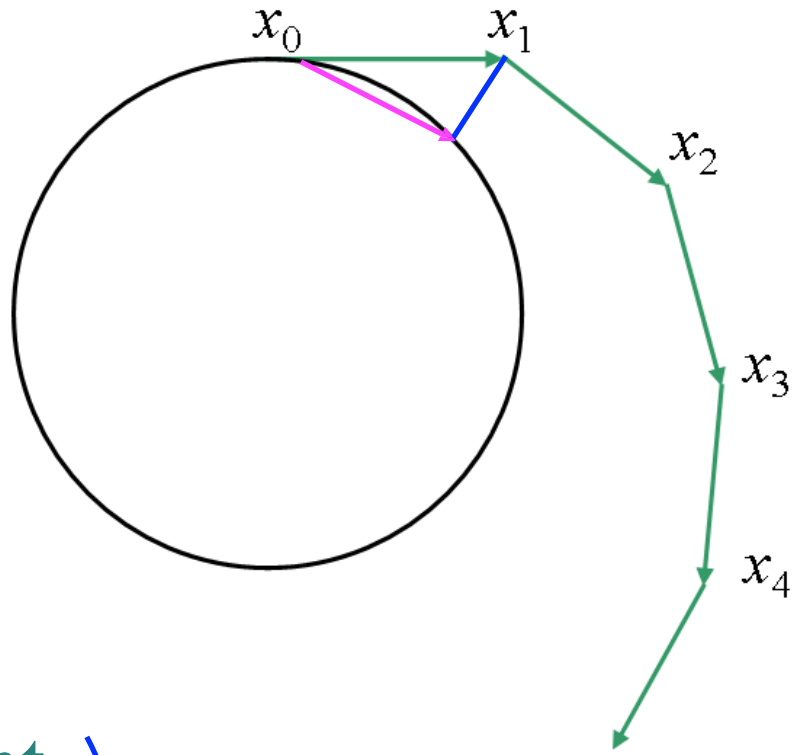
Και πάλι δεν λύνεται πλήρως το πρόβλημα της τροχιάς

- **Πολλαπλές επαναλήψεις ανά frame**
  - Έστω h το μήκος του frame
  - Έστω n το πλήθος των επαναλήψεων
- Τυπική παράμετρος της μηχανής φυσικής

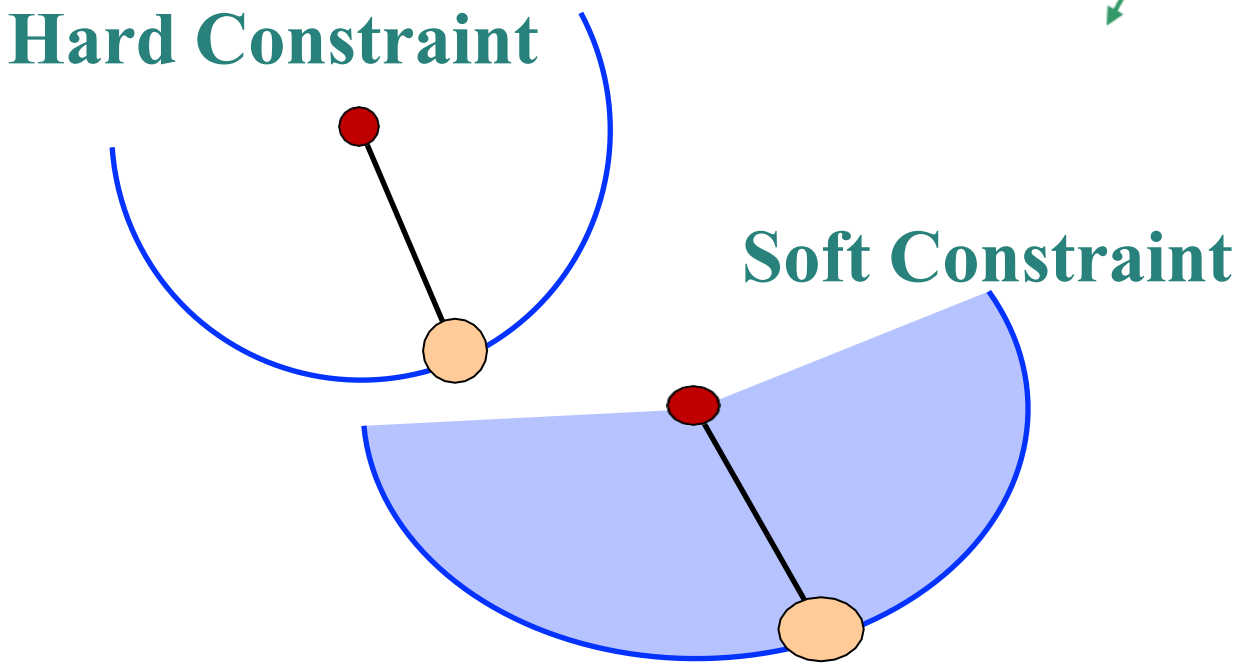
$$\Delta t = h/n$$

Επιλυτές Περιορισμών





- **Περιορισμός** κίνησης αντικειμένων
  - Η θέση πρέπει να ικανοποιεί τον περιορισμό
  - Διόρθωση της θέσης αν δεν ικανοποιεί
- **Παράδειγμα:** Απόσταση
  - Hard: Η απόσταση είναι ακριβής
  - Soft: Πάνω φράγμα στην απόσταση
- Άλλοι περιορισμοί
  - Επαφή: όχι διείσδυση
  - Αποκατάσταση: αναπήδηση
  - Τριβή: ολίσθηση, κόλλημα



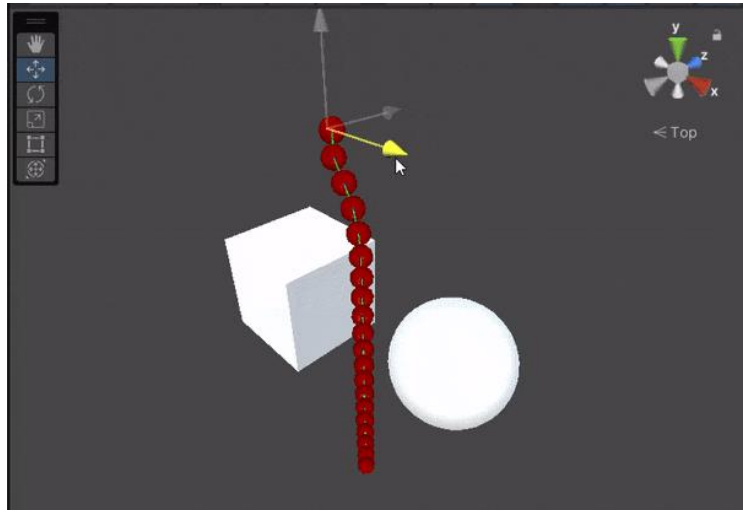
Ο Ολοκληρωτής Verlet

Χρησιμοποιείται περισσότερο στα σύγχρονα παιχνίδια λόγω καλύτερης συμπεριφοράς.

$$\underbrace{\mathbf{S}(t+\Delta t)}_{\text{new state}} = 2 \quad \underbrace{\mathbf{S}(t)}_{\text{prior state 1}} \quad - \quad \underbrace{\mathbf{S}(t-\Delta t)}_{\text{prior state 2}} + (\Delta t)^2 \quad \underbrace{\left(\frac{d^2}{dt^2}\mathbf{S}(t)\right)}_{\text{state derivative}}$$

- Πρέπει να αποθηκεύουμε την κατάσταση σε δύο προηγούμενα βήματα, $\boldsymbol{S}(t)$ και $\boldsymbol{S}(t-\Delta t)$
- Χρησιμοποιεί τη δεύτερη παράγωγο
- Έγκυρη για σταθερό χρονικό βήμα
- Παράδειγμα επιλογής κατάστασης (για σωματίδιο):

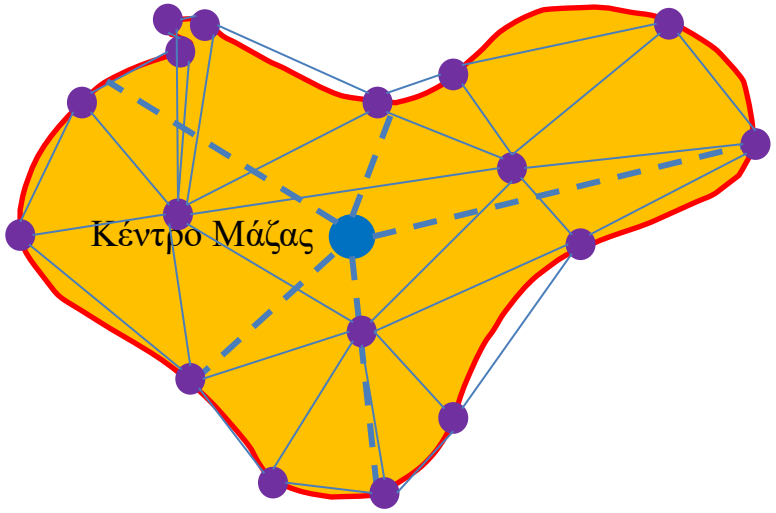
$$\mathbf{S} = \langle \mathbf{p} \rangle$$
$$d^2\mathbf{S}/dt^2 = \langle \mathbf{F}/m \rangle = \langle \mathbf{a} \rangle$$



Άκαμπτα Σώματα

Κίνηση: Μοντελοποίηση Αντικειμένων

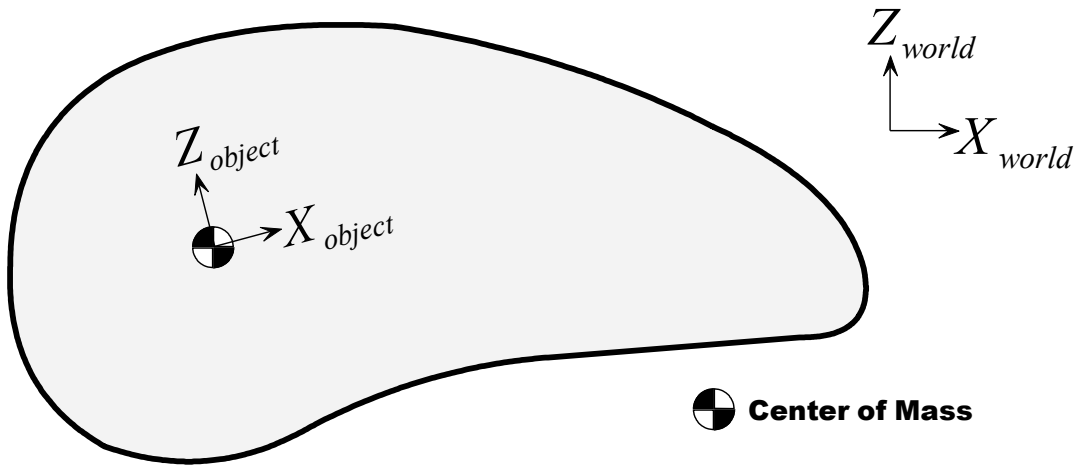
- Συνήθως η γεωμετρία αγνοείται
  - Δεν ενδιαφερόμαστε για το σχήμα
  - Χρειάζεται μόνο για συγκρούσεις
- Κάθε αντικείμενο είναι ένα σημείο
  - *Κεντροειδές*: μέσος όρος σημείων
  - Αποκαλείται επίσης *κέντρο μάζας*
  - Είναι ίδια αν η πυκνότητα είναι ομοιόμορφη
- Χρήση άκαμπτου σώματος αν απαιτείται
  - Πολλαπλά σημεία
  - Όταν κινείται ένα, κινούνται όλα



Άκαμπτο Σώμα

Άκαμπτα Σώματα

- Βασικές διαφορές από σωματίδια
  - Όχι απαραίτητα σφαιρικά
  - Η θέση του $\boldsymbol{p}$ αναπαριστά τη θέση του κέντρου μάζας του αντικειμένου
  - Η επιφάνεια μπορεί να μην είναι λεία
    - Εμφανίζονται δυνάμεις όπως η τριβή
  - Περιστροφική κίνηση εκτός της κίνησης μετακίνησης



Προσομοίωση

- Γωνιακή Κινηματική:
  - Προσανατολισμός (3×3 μητρώο ή κάποιο quaternion)
  - Γωνιακή ταχύτητα $\boldsymbol{\omega}$
  - Σε συντεταγμένες εικονικού κόσμου

- Επιπλέον ιδιότητες
  - Τένσορας αδράνειας (inertia tensor), $\boldsymbol{J}$
  - Κέντρο μάζας

- Επιπλέον ιδιότητες κατάστασης για την προσομοίωση
  - Προσανατολισμός
  - Γωνιακή ορμή, $\boldsymbol{L = J\omega}$
  - Αντίστοιχες παράγωγοι

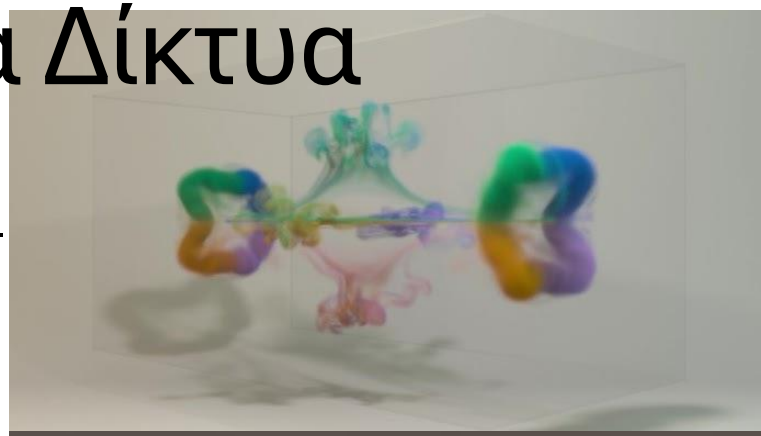
Μία Ματιά πιο Πέρα…: Νευρωνικά Δίκτυα

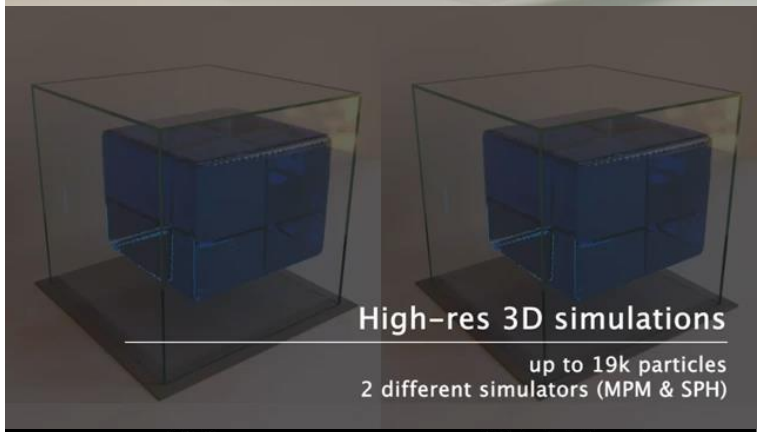
- Hybrid Euler-Lagrangian Neural Pipeline (Neural Flow Maps or Neural Super-Resolution)
  - Coarse Solver
  - Neural Up-Res
  - Correction Step

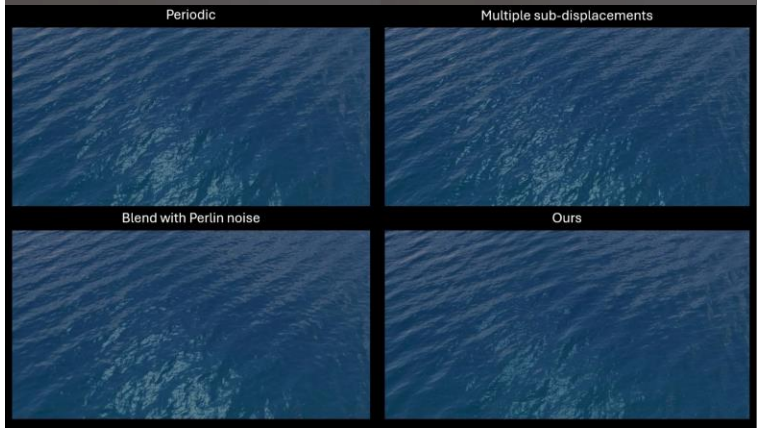
- Graph Neural Networks (GNNs) for Lagrangian Dynamics
  - Nodes (V): Particles (atoms, fluid droplets, cloth vertices).
  - Edges (E): Interactions (springs, gravity, collisions, local fluid pressure).

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \varphi^{(l)}\left(h_i^{(l)}, h_j^{(l)}, e_{ij}\right)\right)$$

- Subspace Simulation and Reduced Order Models
  - Dimensionality Reduction
  - Latent Dynamics
  - Reconstruction





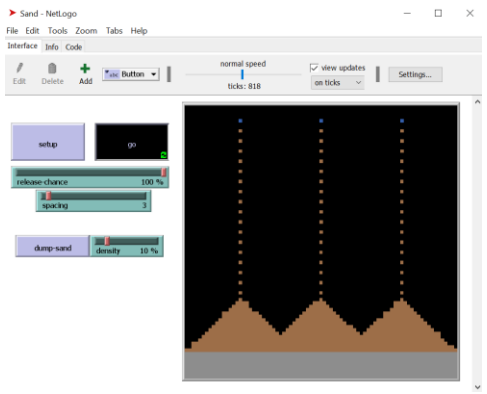


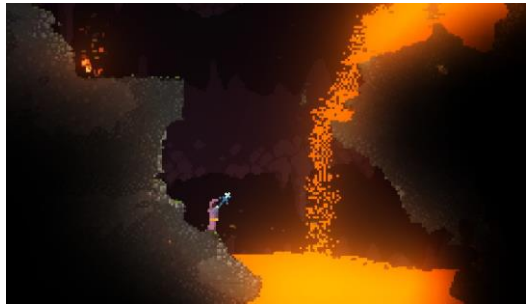
Falling Everything Engine

When pixels become alive...

Η Φυσική του Noita

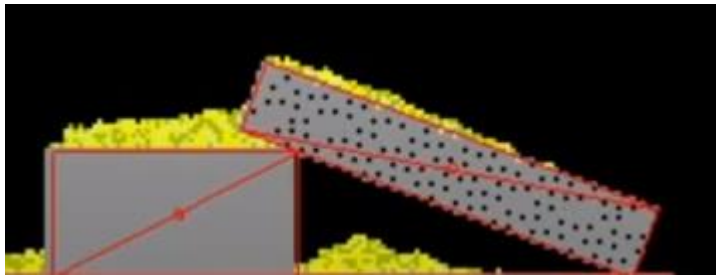
Κυψελωτά Αυτόματα





+

Άκαμπτα Σώματα



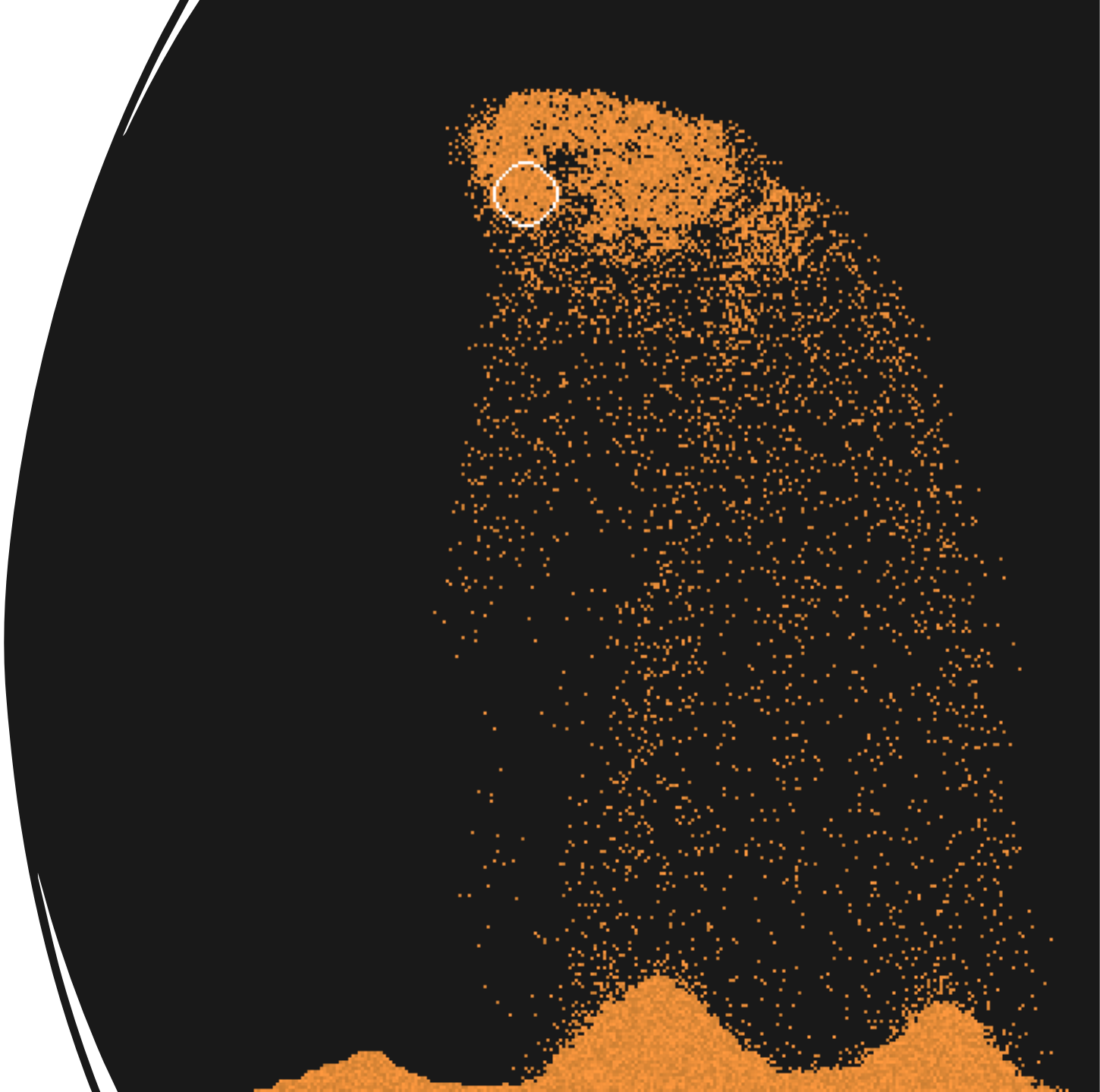
Κυψελωτά Αυτόματα

Πλέγμα: Ο κόσμος χωρίζεται σε ένα πλέγμα από pixel (κελιά)

Κανόνες: Ενημέρωση κάθε pixel λαμβάνοντας υπόψη τους γείτονες.

- Άμμος: Αν το pixel από κάτω μού είναι άδειο προχώρα προς τα κάτω. Αν όχι, τότε αν είναι άδειο το κάτω αριστερά, πήγαινε εκεί.
- Νερό: Αν δεν μπορείς να πας κάτω τότε πήγαινε αριστερά ή δεξιά.

Πυκνότητα υλικών: Αν ένα βαρύτερο pixel είναι πάνω από ένα ελαφρύτερο άλλαξε θέσεις



Φυσική Άκαμπτων Σωμάτων

Αντικείμενα που έχουν:

- Μάζα
- Ταχύτητα
- Περιστροφή

Μετακίνηση και περιστροφή
όπως στις κλασσικές μηχανές

Πώς Συνδυάζονται

- *Marching square αλγόριθμος:* σχεδιάζει συνεχώς ένα collision polygon γύρω από κάθε άκαμπτο σώμα

- *Καταστρεψιμότητα:* Αν χτυπήσετε ένα ξύλινο κουτί τότε αυτό θα σπάσει και ο αλγόριθμος τρέχει για κάθε κομμάτι του που έχει απομείνει.

Αλληλεπιδράσεις:

- Αντικείμενο → Κόσμος
- Κόσμος → Αντικείμενο

Αντιδράσεις

Αντιδράσεις μεταξύ γειτόνων

- Έλεγχος εγγύτητας: Σε κάθε frame έλεγχος για κάθε pixel αν αντιδρά με γείτονες
- Αλλαγή κατάσταση: Αν ένα Pixel «φωτιά» ακουμπήσει ένα pixel «νερό», τότε το «νερό» σβήνει και αντικαθίσταται από ένα pixel «ατμός» που έχει τον κανόνα να ανεβαίνει προς τα πάνω.
- Ετικέτες: Τα υλικά έχουν ετικέτες όπως «εύφλεκτο» ή «ευδιάβρωτο». Ένα «φωτιά» κοιτά αν ο γείτονάς του είναι «εύφλεκτο» και αν είναι το αντικαθιστά με φωτιά.

Βελτιστοποίηση – Η Μαγεία ☺

Εξομοίωση κάθε pixel σε 1080p είναι μάλλον ζόρικο.

- Τμήματα: Ο κόσμος χωρίζεται σε τμήματα, που όσο δεν αλλάζουν, τίποτα δεν υπολογίζεται.

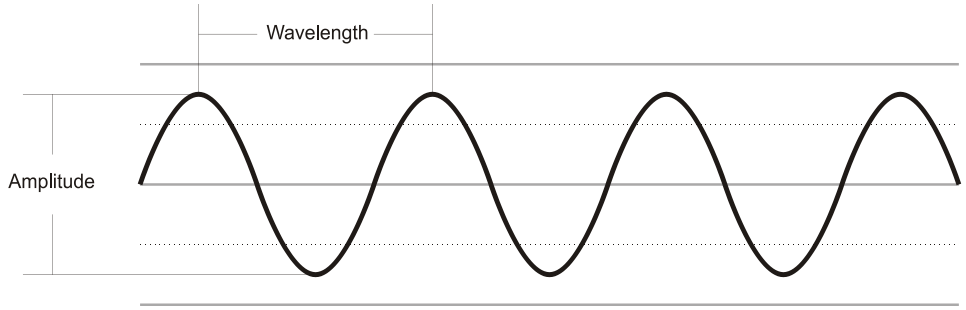
- Νήματα: Διαχωρισμός της οθόνης σε κομμάτια που το καθένα ζωγραφίζεται από διαφορετικό πυρήνα της CPU.

Απόδοση Ήχου σε 3d

Βασικά Στοιχεία

Βασική Ορολογία Ήχου

- Πλάτος (amplitude)
  - Μέτρηση της πίεσης ενός ηχητικού κύματος (θετική ή αρνητική κατεύθυνση)
- Συχνότητα (frequency)
  - Μέτρηση του διαστήματος μεταξύ των κύκλων κυμάτων, συνήθως σε Hertz (κύκλοι/δευτερόλεπτο)
- Ύψος (Pitch)
  - Η αντίληψη της συχνότητας που συνήθως χρησιμοποιείται εναλλακτικά με τη συχνότητα
- Decibel
  - Μετρά το πλάτος του ήχου, μετρά την αντιληπτή διαφορά στην ένταση μεταξύ δύο ήχων



3d Μηχανή Ήχου

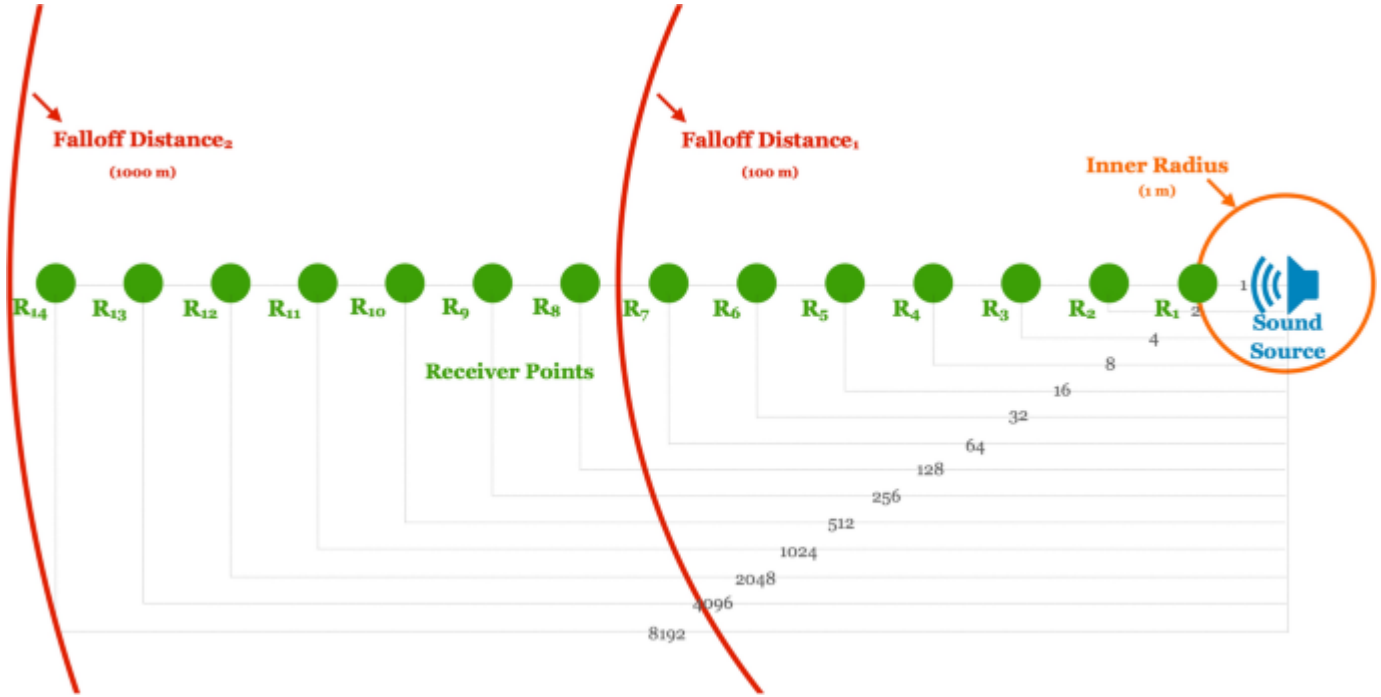
Βασικές εργασίες:

1. Σύνθεση ήχου
2. Χωροταξία ήχου
    1. Εξασθένιση με βάση την απόσταση
    2. Pan
3. Ακουστική μοντελοποίηση
    1. Ανακλάσεις
    2. Αντηχήσεις
    3. Ατμοσφαιρική απορρόφηση
    4. HRTF (Head-Related Transfer Function) φαινόμενα
4. Μετατόπιση Doppler
5. Μίξη

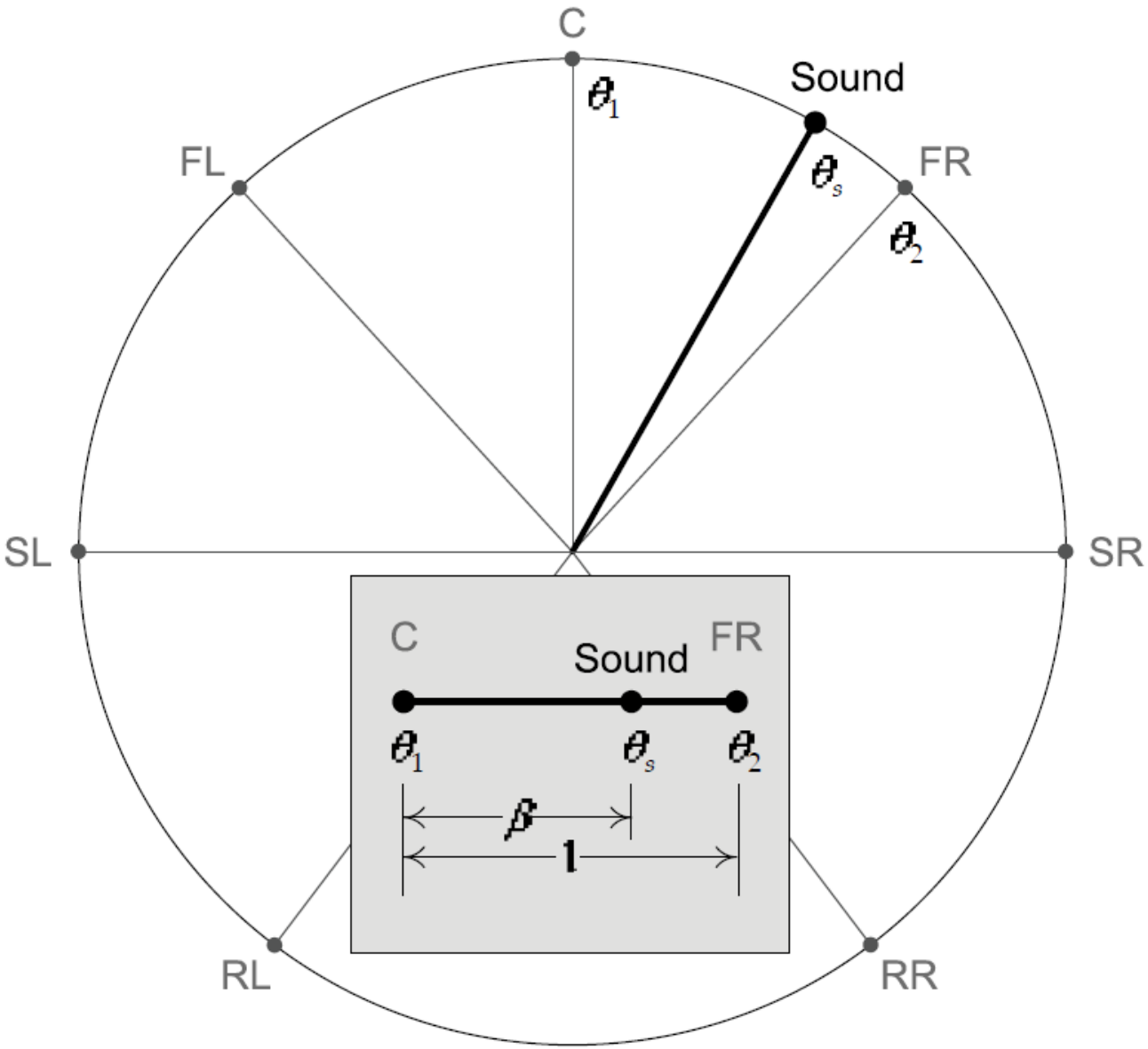
Μοντελοποίηση Ηχητικού Κόσμου

- 3d πηγές ήχου
  - Θέση, ταχύτητα, σχήμα διάδοσης, εμβέλεια
- Ακροατής
  - Εικονικό μικρόφωνο: θέση, ταχύτητα, προσανατολισμός
- Μοντέλο περιβάλλοντος
  - Γεωμετρία, ακουστικές ιδιότητες επιφανειών, αντικειμένων και χώρων

Εξασθένιση με βάση την Απόσταση

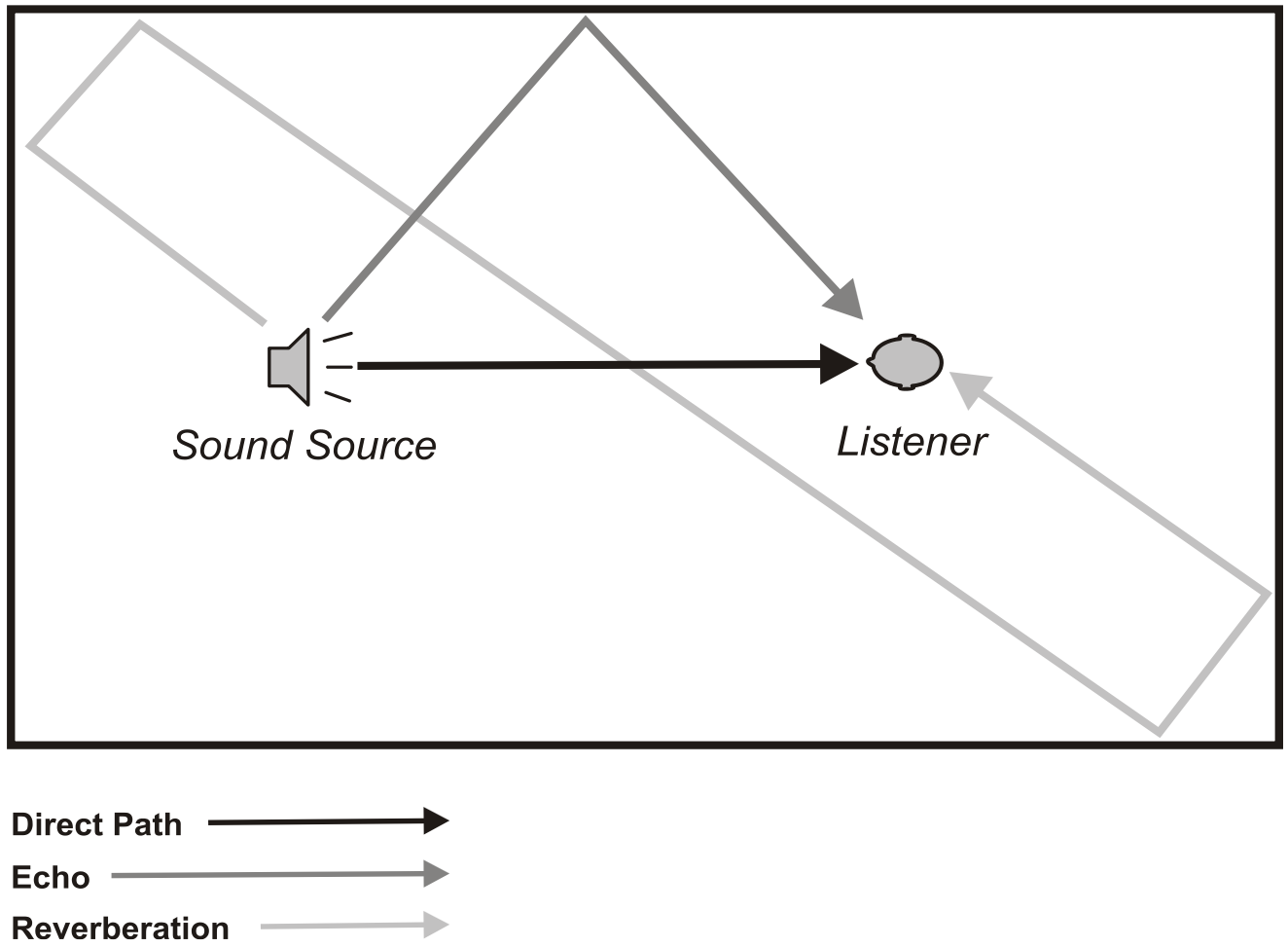


Pan

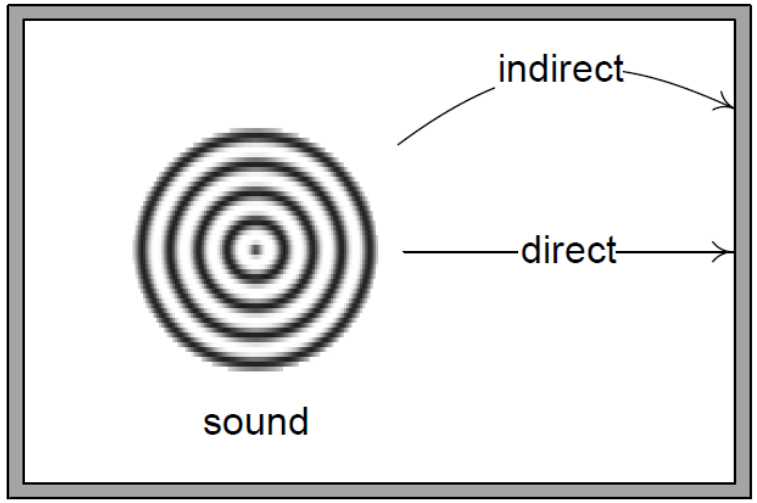

C
Sound
θ_1
θ_s
θ_2
FL
FR
SL
SR
RL
RR
C
Sound
FR
θ_1
θ_s
θ_2
β
1

Τύποι Μετάδοσης Ήχου

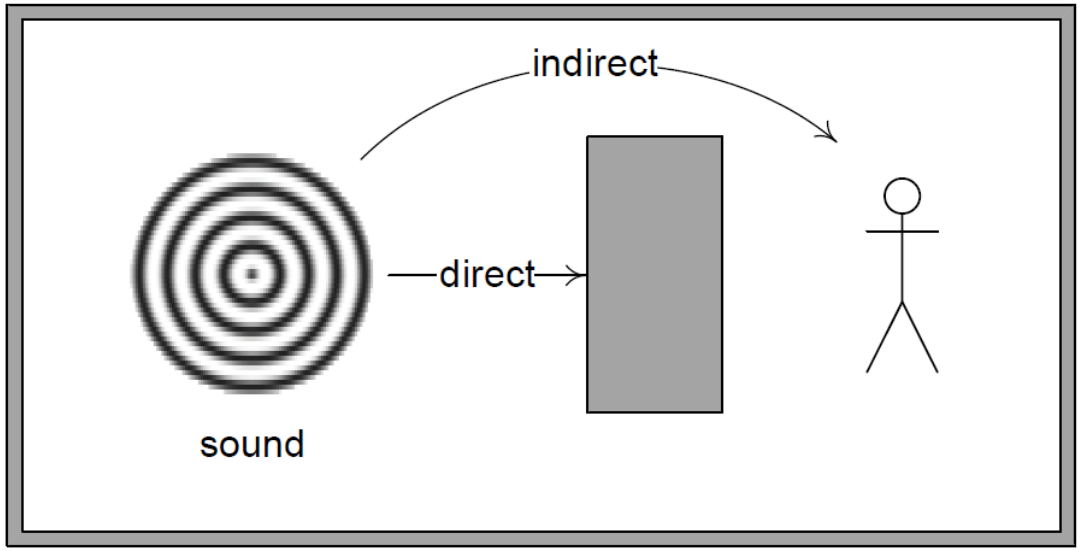
- Άμεση μετάδοση (direct path)
- Πρώιμες αντανακλάσεις (ηχώ – echo)
- Καθυστερημένες αντανακλάσεις (αντήχηση – reverberation)



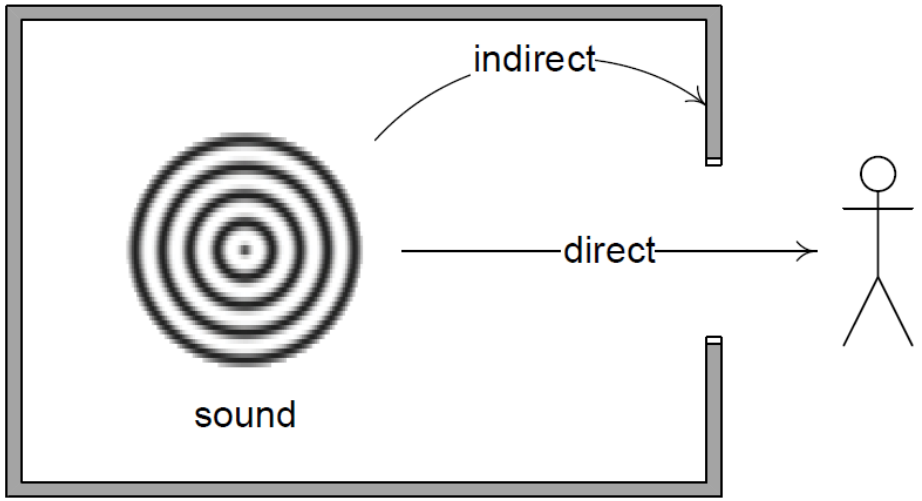
Λαμβάνοντας υπόψη τη Γεωμετρία



Αποκοπή
(occlusion)



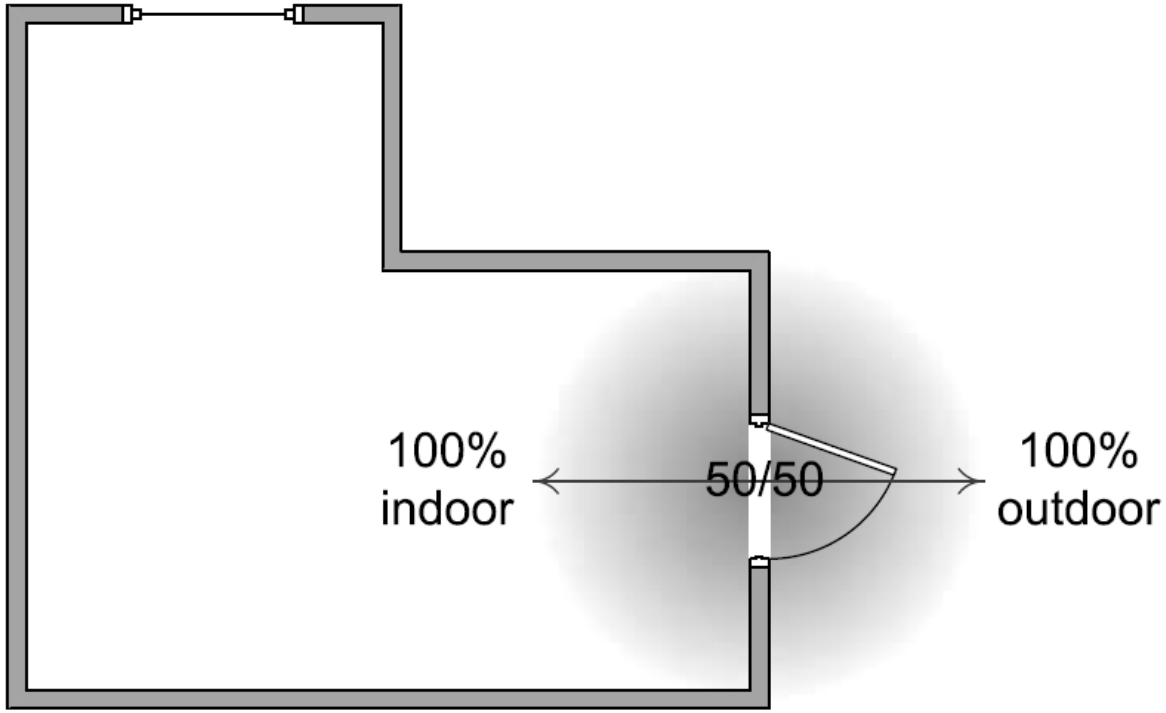
Παρεμπόδιση
(obstruction)



Αποκλεισμός
(exclusion)

Αντήχηση

- Κάθε περιοχή έχει δικά της χαρακτηριστικά αντήχησης
  - Ένα μικρό δωμάτιο δεν έχει σχεδόν καθόλου ηχώ
  - Σε ένα μεγάλο δωμάτιο μπορούμε να ανεβάσουμε την ηχώ
  - Όταν περνάμε μεταξύ περιοχών με διαφορετικά ακουστικά χαρακτηριστικά πρέπει να γίνεται ομαλή μετάβαση (π.χ., με LERP)



Πιο προχωρημένο: Τεχνολογία Συγχρονισμού Χειλιών (Lip-sync)

- Η τεχνολογία Lip-sync είναι ένας συνδυασμός τεχνικών ήχου και εικόνας για τη δημιουργία ρεαλιστικής ομιλίας από τους ηθοποιούς του παιχνιδιού.
  - Παιχνίδια όπως το Mass Effect Andromeda (2017) είχαν κατηγορηθεί για κακή εφαρμογή της τεχνολογίας
  - Unreal 5.5 (Metahumans)