

Σύστημα Τεχνητής Νοημοσύνης (AI)

Global Game Jam (26/1-1/02/2026)

<https://globalgamejam.org/>

1. *Make a game in 48 hours*
2. *On an announced theme*

LOCATION

- **University of Bremen**, Bibliothekstr. 1
- Building: MZH, Floor 0
- Rooms: P1 "Windows", P2 "Apple", P3 "Linux"



REQUIREMENTS

- **Warm clothes** because it can be cold at night!
- If you prefer to stay for the whole night: a **sleeping bag** and a camping mat or similar
- **Food and drinks** for 48 hours!
We probably will be able to provide coffee, tea, softdrinks and snacks but real food has to be brought by the participants or ordered at your own expense.
- For computer game development:
Bring your own **computer and software**. Paper and pens might be helpful in the planning phase.
- For board game development:
Gaming materials (dices, chips, tokens, ...) and craft stuff (paper, pens, cardboard, glue, post-its, ...).
- We will be able to provide some general software as well as some board game materials. But we strongly recommend to install the needed software beforehand and to bring your own materials.
- Bring a switch + cables if you need wired network. We only provide WIFI.

RULES

- **Computer games have to be uploaded** to the globalgamejam.org website either as a package or a browsergame.
- **Board games have to be presented as "print out games"** (pdf format) and uploaded to the globalgamejam.org website.
- **All games have to include certain characteristics** which will be announced in the beginning of the jam, so that every participant has a similar challenge to start off with.

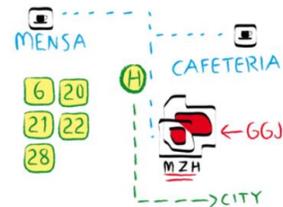
SCHEDULE

Friday
15.00 Check In
15.45 Welcome Presentation
16.00 Start of the Global Game Jam
18.00 Concept Presentations

Saturday
12.00 Game Registration Session
13.00 **Game Registration Deadline**
18.00 **Prototype Presentations**

Sunday
16.00 **Game Update Deadline**
16:05 **Five Minutes Game Presentations**
16:45 Voting for the Games
17:00 End of Global Game Jam
19:00 Jam Site Closes

↑
ANOTHER 48 HRS.
↓



GLOBAL GAME JAM

Get together and create games in one weekend (non-stop) in over 800 locations worldwide.

Themes:

- 2009 – "As long as we have each other, we will never run out of problems"^[17]
- 2010 – "Deception"^[17]
- 2011 – "Extinction"^[17]
- 2012 – An image of [Ouroboros](#)^[19]
- 2013 – Sound of a Heartbeat^[20]
- 2014 – "We don't see things as they are, we see them as we are."^[21]
- 2015 – "What do we do now?"^[22]
- 2016 – "Ritual"^[17]
- 2017 – "Waves"^[17]
- 2018 – "Transmission"^[23]
- 2019 – "What home means to you"^[24]
- 2020 – "Repair"^[25]
- 2021 – "Lost and Found"^[17]
- 2022 – "Duality"^[26]
- 2023 – "Roots"^[27]
- 2024 – "Make Me Laugh"^[28]
- 2025 – "Bubble"^[29]

Τί είναι το AI σε Βιντεοπαιχνίδια;

- Είναι η συμπεριφορά του NPC;
 - Μία απλή δήλωση “if”;
 - Συμπεριφορά από σενάριο;
- Εύρεση διαδρομής;
- Επιλογή animation;
- Αυτόματη παραγωγή περιβάλλοντος;

“Game AI is anything that contributes to the perceived intelligence of an entity, regardless of what’s under the hood.”

Τέσσερις Όψεις της AI

- Συστήματα που **σκέφτονται σαν άνθρωποι**
 - γνωσιακή επιστήμη, έμπειρα συστήματα
- Συστήματα που **δρουν σαν άνθρωποι**
 - Turing test, προγράμματα σκάκι
- Συστήματα που **σκέφτονται λογικά**
 - Προσεγγίσεις βασισμένες σε Λογική και Μαθηματικά
- Συστήματα που **δρουν λογικά**
 - Σύγχρονες προσεγγίσεις βασισμένες σε πράκτορες

Αφορά την **ουσία** των πραγμάτων!

AI Βιντεοπαιχνιδιών

Αφορά την **εμφάνιση** των πραγμάτων!

- Μηχανική: ανάπτυξη αλγορίθμων που *φαίνεται* ότι συμπεριφέρονται έξυπνα και λογικά
- Μερικές φορές αυτές οι δύο προσεγγίσεις επικαλύπτονται

Τρόποι για AI Βιντεοπαιχνιδιών

- “Τυπική”
 - Μηχανές πεπερασμένων καταστάσεων (FSM), σχεδιασμός, εύρεση διαδρομής (pathfinding), αποφυγή σύγκρουσης, έμπειρα συστήματα, ασαφής λογική,...
- “Ad-Hoc”
 - Κόλπα
 - Το animation μπορεί να *δείξει/εμφανίσει* περισσότερη ευφυΐα από έναν αλγόριθμο
 - Ευρετικές τεχνικές

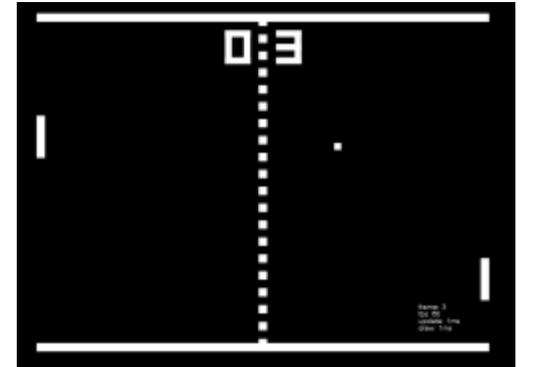
Κάποια (Παλιά) Παιχνίδια

Pong!

Η αρχική έκδοση (1972) ήταν μόνο για δύο παίκτες (δεν υπάρχει AI)

Μετάπειτα εκδοχές είχαν ένα είδος AI:

- *Μοντελοποίηση χρόνου αντίδρασης*: περίμενε κάποιο χρονικό διάστημα πριν μετακινήσει τη πλατφόρμα
- *Μοντελοποίηση ακρίβειας*: αν και ο Η/Υ μπορεί να βρει ακριβώς που θα πέσει η μπάλα, προσθέτει ένα τυχαίο σφάλμα ώστε να προσποιηθεί ότι έκανε λάθος



Πώς θα το
μοντελοποιούσατε;

Pac-Man

ΑΙ Τεχνική: Μηχανή καταστάσεων

Υπάρχουν 4 τέρατα που μπορεί να είναι σε μία από τις 3 καταστάσεις:

- κυνήγι
- διασπορά
- φόβος

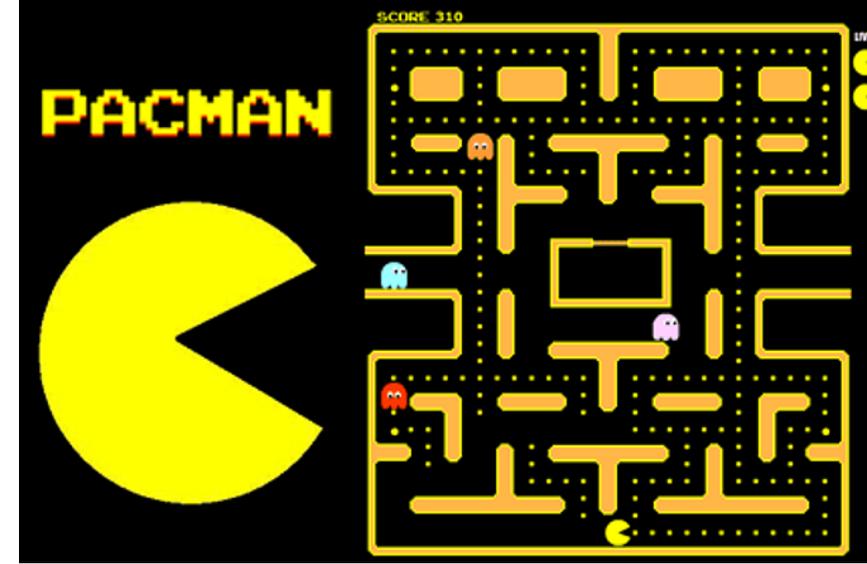
Σε κάθε κατάσταση επιλέγουν ένα κελί ως στόχο και στρίβουν προς τα εκεί σε κάθε διασταύρωση

1. **CHASE** - A ghost's objective in *chase* mode is to find and capture Pac-Man by hunting him down through the maze. Each ghost exhibits unique behavior when chasing Pac-Man, giving them their different personalities: Blinky (red) is very aggressive and hard to shake once he gets behind you, Pinky (pink) tends to get in front of you and cut you off, Inky (light blue) is the least predictable of the bunch, and Clyde (orange) seems to do his own thing and stay out of the way.
2. **SCATTER** - In *scatter* mode, the ghosts give up the chase for a few seconds and head for their respective home corners. It is a welcome but brief rest- soon enough, they will revert to chase mode and be after Pac-Man again.
3. **FRIGHTENED** - Ghosts enter *frightened* mode whenever Pac-Man eats one of the four energizers located in the far corners of the maze. During the early levels, the ghosts will all turn dark blue (meaning they are vulnerable) and aimlessly wander the maze for a few seconds. They will flash moments before returning to their previous mode of behavior.



22/05/1980 στην Ιαπωνία

Οι 3 Καταστάσεις στο Pac-Man



Τα τέρατα εναλλάσσονται μεταξύ των καταστάσεων εξερεύνησης και κυνηγιού σε προκαθορισμένα διαστήματα.

Τα τέρατα πρέπει να αλλάξουν κατεύθυνση όταν έχουμε τις εξής μεταβάσεις καταστάσεων:
κυνήγι→διασπορά, κυνήγι→φόβος, διασπορά→κυνήγι,
διασπορά→φόβος.

Στις άλλες περιπτώσεις μετάβασης τα τέρατα δεν αλλάζουν κατεύθυνση.

| Mode | Level 1 | Levels 2-4 | Levels 5+ |
|---------|------------|------------|------------|
| Scatter | 7 | 7 | 5 |
| Chase | 20 | 20 | 20 |
| Scatter | 7 | 7 | 5 |
| Chase | 20 | 20 | 20 |
| Scatter | 5 | 5 | 5 |
| Chase | 20 | 1033 | 1037 |
| Scatter | 5 | 1/60 | 1/60 |
| Chase | indefinite | indefinite | indefinite |

Προσομοίωση Αισθήσεων

Goldeneye 007 (1997)

- Το AI των χαρακτήρων βασίζονταν σε ένα μικρό πλήθος καταστάσεων
- Καινοτόμησε προσθέτοντας ένα σύστημα οπτικής προσομοίωσης:
 - Οι χαρακτήρες μπορούσαν να δουν τους συντρόφους τους και να παρατηρήσουν ότι έχουν σκοτωθεί

Πολύ δημοφιλής με ολόκληρα παιχνίδια (1998) να ορίζουν το gameplay πάνω σε τέτοιες προσομοιώσεις

- Όχι πολύπλοκο:
 - Έλεγχος με raycast και σκιά για το αν ένα αντικείμενο είναι ορατό από τους NPCs
 - Για τον ήχο γινόταν έλεγχος χωρικής κοντινότητας για το αν τον ακούν οι NPCs



GoldenEye 007

Video game

9.7/10
IGN

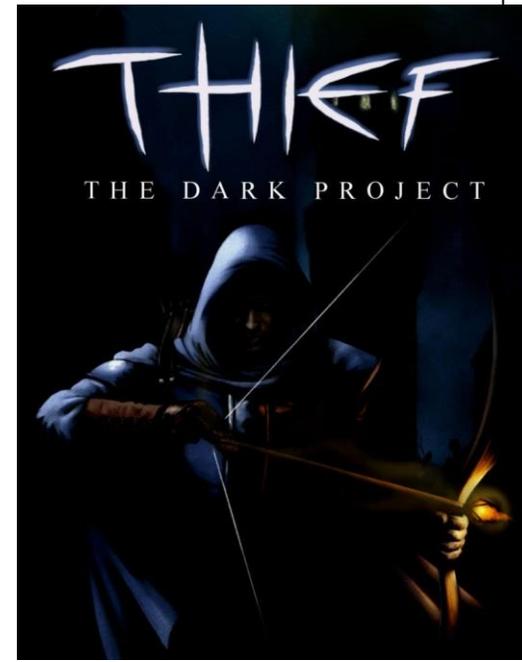
4.5/5
GameStop

95% liked this video game

Google users

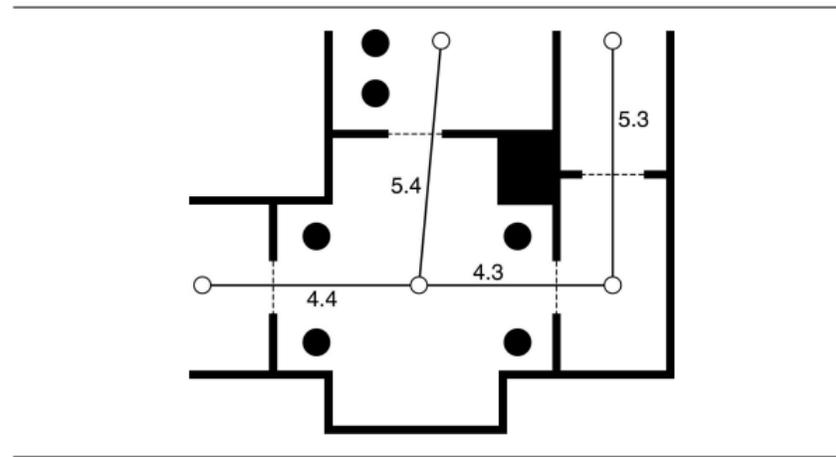
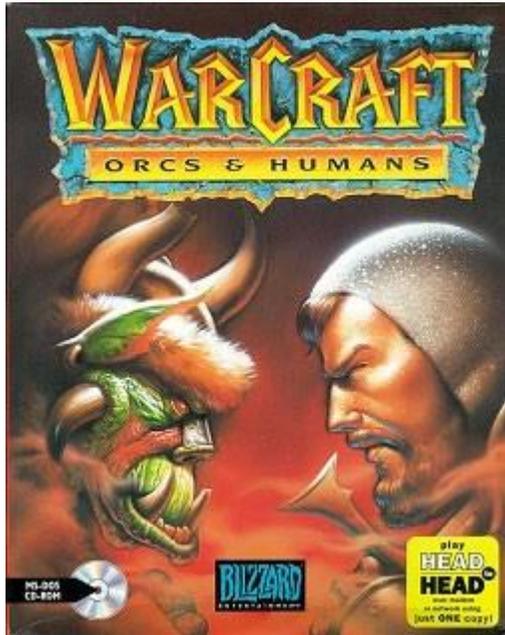
GoldenEye 007 is a 1997 first-person shooter developed by Rare and published by Nintendo for the Nintendo 64. [Wikipedia](#)

Initial release date: August 25, 1997



Warcraft

Στα μέσα της δεκαετίας του 1990 όπου τα RTS – Real Time Strategy – ήταν εξαιρετικά δημοφιλή, χρησιμοποιήθηκαν μέθοδοι **εύρεσης διαδρομής**

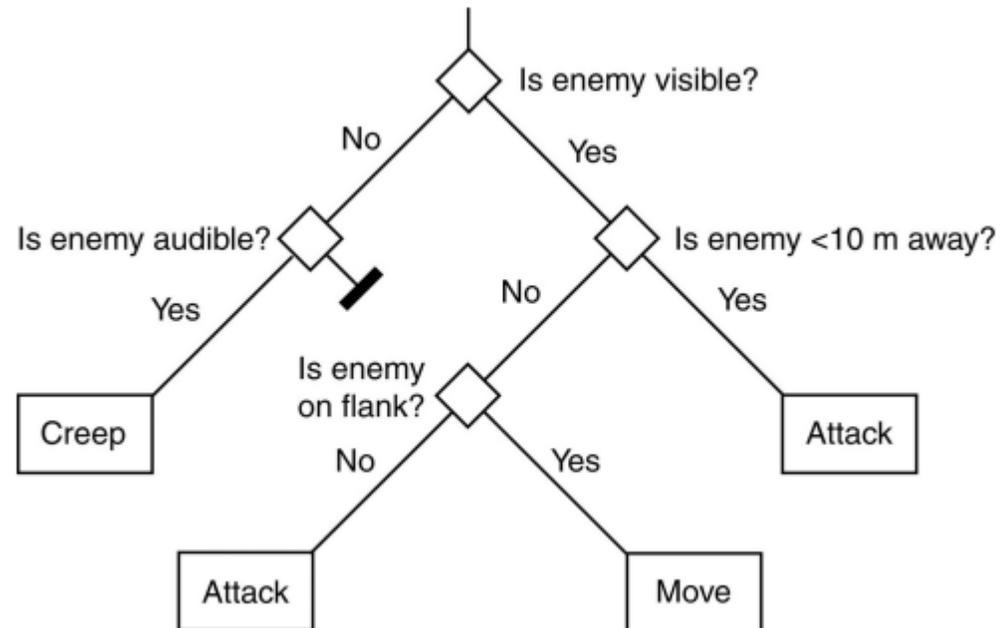


Εύρεση διαδρομής: Συνήθως αναζήτηση συντομότερης διαδρομής σε γράφημα

Halo

Το Halo (2001) εισήγαγε τη χρήση δένδρων απόφασης

Τώρα πια είναι μία συνηθισμένη τεχνική για το τι θα κάνει ένας NPC



Starcraft AI

“First person shooters and RTS games have been subjected to significant academic research (there is an annual competition for Starcraft AI, for example).”

Millington, Ian. AI for Games, Third Edition

NEWS | 30 October 2019

Google AI beats top human players at strategy game *StarCraft II*

DeepMind's AlphaStar beat all but the very best humans at the fast-paced sci-fi video game.

Dan Garisto

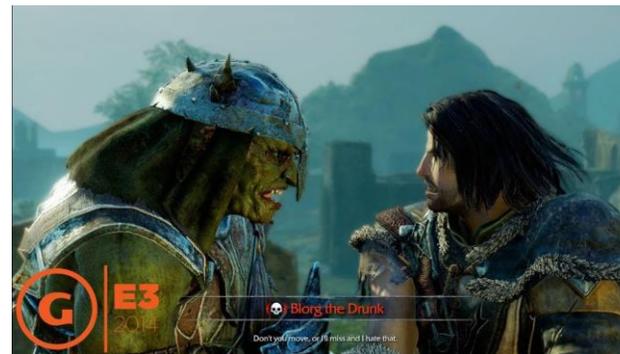


Μία Ματιά πιο Πρόσφατα...



The Last Of Us: Part II

Χρησιμοποιεί προχωρημένο AI για τη συμπεριφορά των NPCs επιδεικνύοντας ρεαλιστική συμπεριφορά τόσο σε σχέση με τον παίκτη όσο και με το περιβάλλον



Middle-Earth: Shadow of Mordor

Nemesis σύστημα: Προσωποποιεί τις αλληλεπιδράσεις με τους NPCs με βάση τις επιλογές του παίκτη



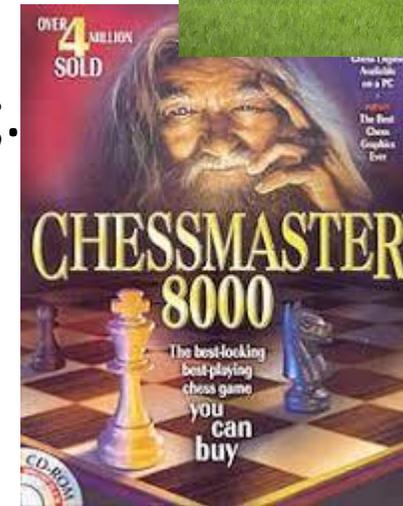
No Man's Sky

Χρησιμοποιεί διαδικαστική παραγωγή για τη δημιουργία ενός τεράστιου σύμπαντος

Βασικά Στοιχεία

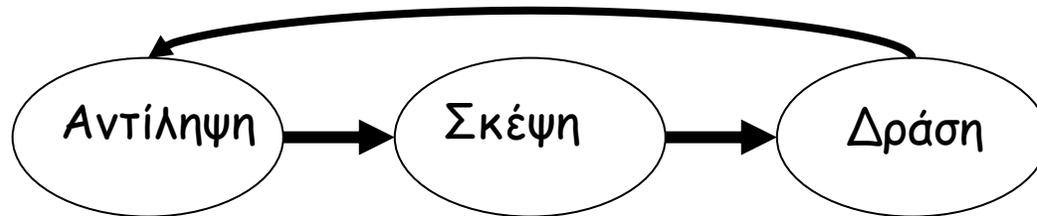
Ευφυείς Οντότητες

- Πράκτορες Παιχνιδιού
 - Αυτόνομες οντότητες που παρατηρούν και επιδρούν στο εικονικό περιβάλλον.
 - NPCs
- Εικονικός παίκτης
 - Κάνει ό,τι κάνει και ο ανθρώπινος παίκτης.
 - Σκάκι



Ο Κύκλος ενός Πράκτορα

- Το μεγαλύτερο μέρος του AI εστιάζει στους πράκτορες
 - εχθρός, σύμμαχος, ουδέτερος NPC
- Κύκλος: Αντίληψη – Σκέψη – Δράση



Όλα Μαζί: Κοινές Τεχνικές AI σε Βιντεοπαιχνίδια (1/4)

Κίνηση

- *Συσώρευση (Flocking)*

- Μετακίνηση ομάδων οντοτήτων με έναν φυσικό τρόπο
- Κάθε οντότητα ακολουθεί τρεις κανόνες
 - *Διαχωρισμός* – αλλαγή κατεύθυνσης για αποφυγή συνωστισμού
 - *Ευθυγράμμιση* – αλλαγή κατεύθυνσης προς τη μέση κατεύθυνση της ομάδας
 - *Συνοχή* – αλλαγή κατεύθυνσης προς τη μέση θέση της ομάδας
- Παράδειγμα: Σμήνη πουλιών/ψαριών. Με κατάλληλη τροποποίηση εφαρμόζεται σε ομάδες εχθρών (Diablo;;;)



- *Σχηματισμοί (Formations)*

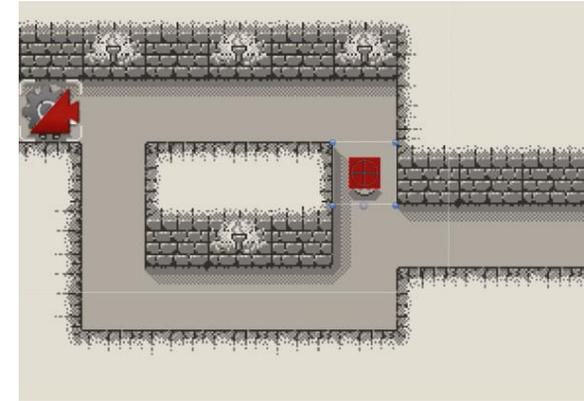
- Όπως η συσώρευση, αλλά οι μονάδες κρατάνε τις σχετικές θέσεις μεταξύ τους
- Παράδειγμα: στρατιωτικοί σχηματισμοί (οι τοξότες πίσω)



Όλα Μαζί: Κοινές Τεχνικές AI σε Βιντεοπαιχνίδια (2/4)

Κίνηση

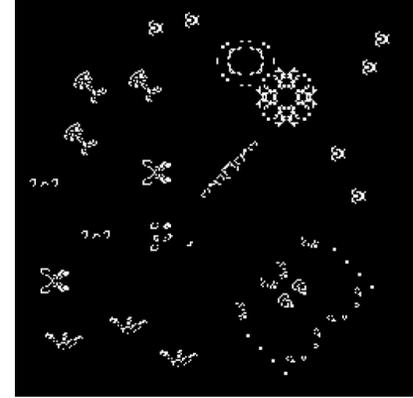
- *A* εύρεση διαδρομής*
 - Φθηνότερη διαδρομή στο εικονικό περιβάλλον
 - Χρήση της γνώσης για τον προορισμό ώστε να γίνει έξυπνα η αναζήτηση της διαδρομής
 - Εξαιρετικά γρήγορος αλγόριθμος, ευρεία χρήση
 - Ενδιάμεση βοηθητική πληροφορία που βοηθά στην αναζήτηση της διαδρομής χωρίς επαναυπολογισμό
- *Αποφυγή εμποδίων*
 - Ο A* είναι καλός για στατικό περιβάλλον αλλά όχι για δυναμικό (άλλοι παίκτες, σημεία συμφόρησης, κ.α.)
 - Παράδειγμα: ίδια διαδρομή για 4 μονάδες όπου γίνεται πρόβλεψη σύγκρουσης ώστε η μονάδα που είναι πιο πίσω να μειώσει την ταχύτητα, κοκ.

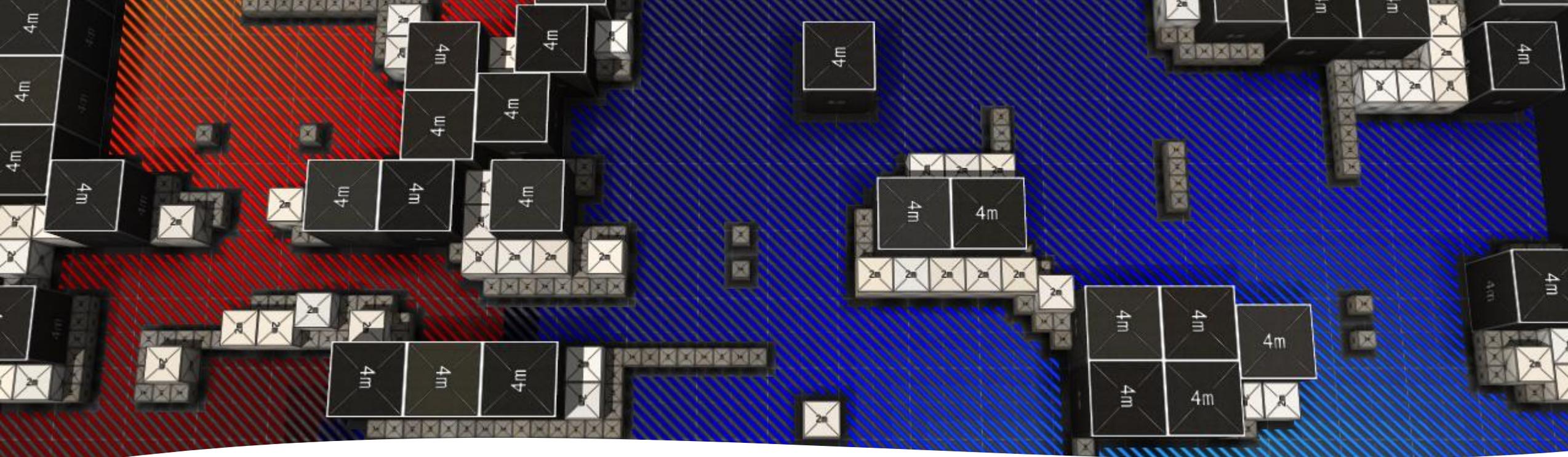


Όλα Μαζί: Κοινές Τεχνικές AI σε Βιντεοπαιχνίδια (3/4)

Οργάνωση Συμπεριφοράς

- *Αναδυόμενη συμπεριφορά*
 - Απλοί κανόνες οδηγούν σε πολύπλοκες αλληλεπιδράσεις
 - Παράδειγμα: παιχνίδι της ζωής, flocking
- *Ιεραρχία εντολών*
 - Η AI παίρνει αποφάσεις σε διαφορετικά επίπεδα
 - Σαν τη στρατιωτική ιεραρχία
 - Παράδειγμα: Παιχνίδια στρατηγικής – συνολική στρατηγική, τακτική λόχου, τακτική στρατιώτη
- *Διαχειριστής ανάθεσης εργασιών*
 - Όταν μονάδες ενεργούν μεμονωμένα τότε μπορεί να έχουν κακή απόδοση
 - Ένας διαχειριστής φτιάχνει εργασίες, δίνει προτεραιότητες και τις αναθέτει σε μονάδες
 - Παράδειγμα: Ποδόσφαιρο – 1^η προτεραιότητα: παίκτης κινείται προς την μπάλα, 2^η προτεραιότητα: τοποθέτηση στο κέντρο της άμυνας, 3^η προτεραιότητα: τοποθέτηση στα δεξιά της άμυνας, 4^η προτεραιότητα: τοποθέτηση στα αριστερά της άμυνας. Ο διαχειριστής αποφασίζει ποιος είναι ο καλύτερος παίκτης σε κάθε περίπτωση.





Όλα Μαζί: Κοινές Τεχνικές AI σε Βιντεοπαιχνίδια (4/4)

- *Χάρτης επιρροής*
 - 2d αναπαράσταση ισχύος στο παιχνίδι
 - Διαχωρισμός σε κελιά, όπου οι μονάδες σε κάθε κελί αθροίζονται ως προς την ισχύ τους
 - Οι μονάδες έχουν επιρροή σε γειτονικά κελιά (μειώνεται σε σχέση με την απόσταση)
 - Πληροφόρηση για τη θέση και την ισχύ των δυνάμεων
 - Παράδειγμα: Σχεδιασμός επιθέσεων και αναγνώριση αδύνατων σημείων του εχθρού ή δικών μας ώστε να ενισχύσουμε την άμυνα
- *LOD (Level of Detail) AI*
 - Ίδια ιδέα με LOD στα γραφικά – μικρότερος υπολογισμός αν δεν φαίνεται
 - Παράδειγμα: αλλαγή συχνότητας ενημέρωσης του NPC με βάση τη σχετική του θέση ως προς τον παίκτη

Υποσχόμενες AI Τεχνικές

Γενικά δεν χρησιμοποιούνται (όχι ευρέως τουλάχιστον) σε παιχνίδια αλλά έχουν τη δυναμική για να το κάνουν κάποια στιγμή

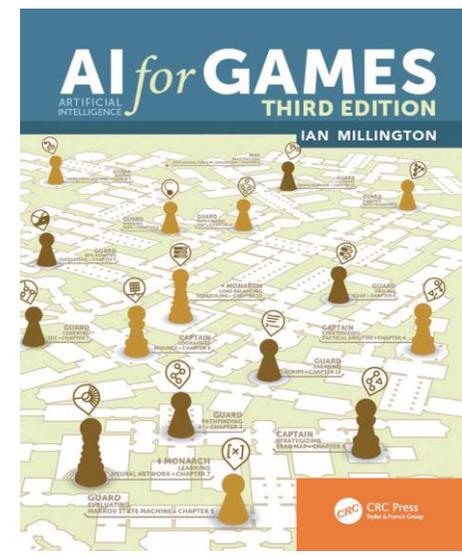
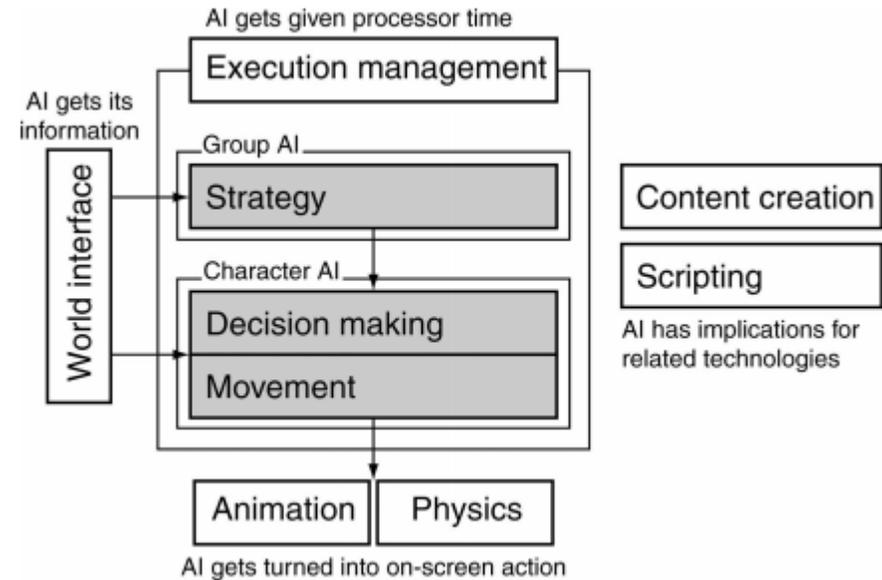
- Γιατί;

- Μπορεί να μην είναι γνωστές
- Μπορεί να είναι δυσκολονόητες
- Μπορεί να έχουν περιορισμένη χρήση
- Μπορεί να απαιτούν πολύ χρόνο ανάπτυξης
- Μπορεί να απαιτούν πολλούς πόρους

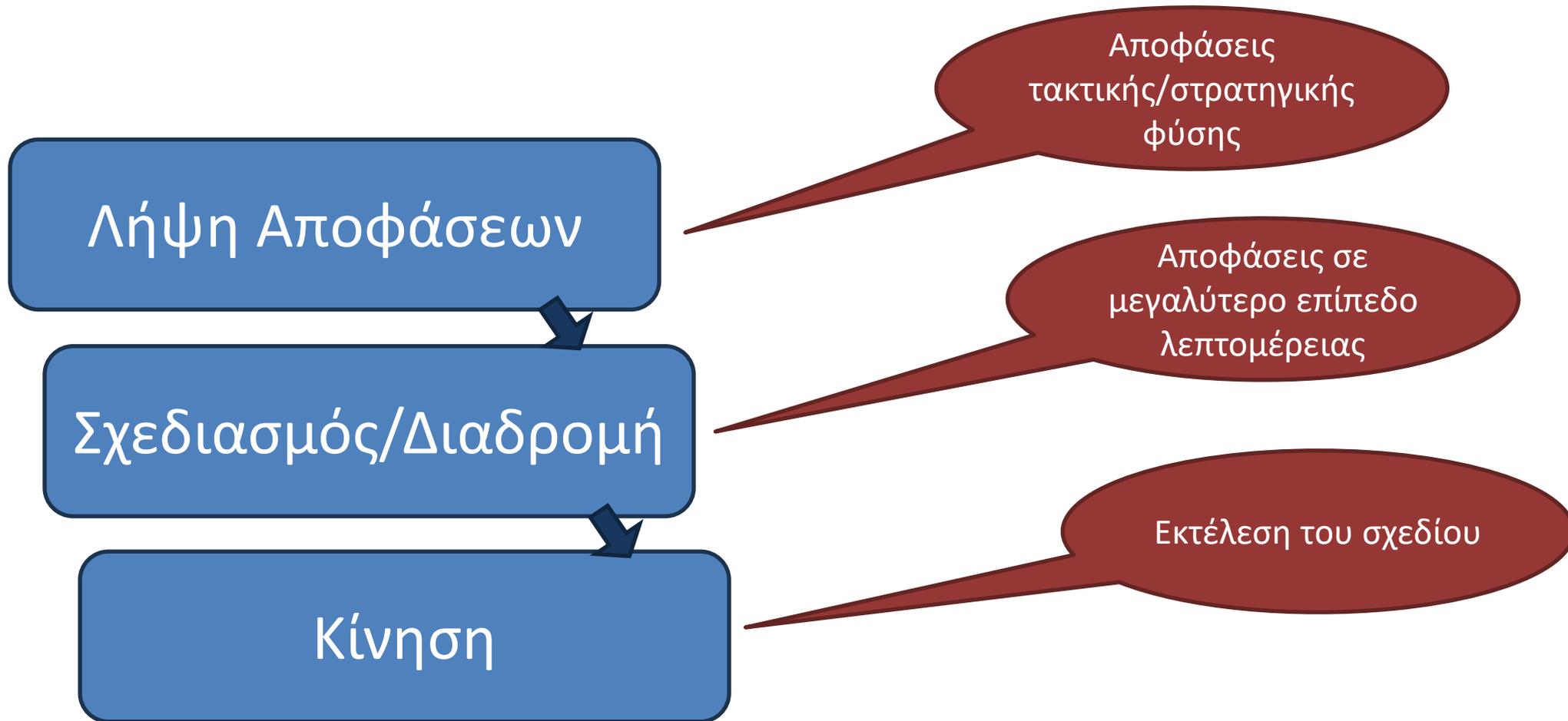
- **Bayesian δίκτυα:** μοντελοποίηση γνώσης του παίκτη από το AI
 - Παράδειγμα: Η AI μπορεί να συμπεράνει σχετικά με την ύπαρξη ή όχι συγκεκριμένων κτιρίων
- **Γενετικοί αλγόριθμοι:** εύρεση λύσης σε «δύσκολο» χώρο αναζήτησης – τυπικά γίνεται offline
 - Παράδειγμα: Ποιες είναι οι βέλτιστες ρυθμίσεις του παιχνιδιού για ένα συγκεκριμένο σύστημα;
- **N-Gram στατιστική πρόβλεψη:** πρόβλεψη επόμενης τιμής σε μία ακολουθία
 - Παράδειγμα: Ξυλίκι... Ο παίκτης κάνει χαμηλή κλωτσιά και έπειτα χαμηλή μπουλιά. Τί ακολουθεί; Από ιστορικά δεδομένα μπορούμε τυχαία με ένα βάρος να κάνουμε σχετική πρόβλεψη.
- **Νευρωνικά δίκτυα**
- **Σύστημα φήμης**
- **Έξυπνο περιβάλλον**
- **Μάθηση**
- **Μοντελοποίηση παίκτη**
- **Σύστημα Πίνακα, ...**

Ένα μοντέλο AI

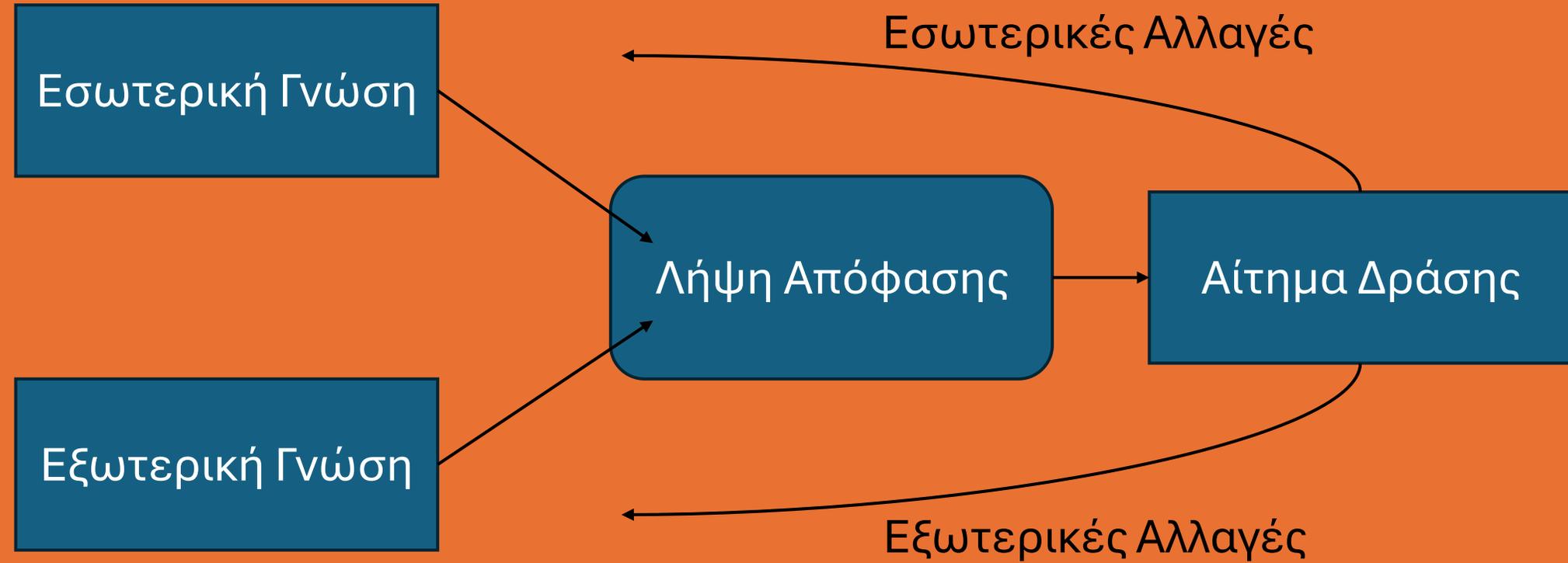
Για βιντεοπαιχνίδια μιλάμε πάντα...



Χονδρική Δομή AI



Δεν είναι πάντα σαφώς διαχωρισμένα



Λήψη Αποφάσεων

- Εξωτερική γνώση: πληροφορία που γνωρίζει ο πράκτορας για το περιβάλλον του (π.χ., κλίμα)
- Εσωτερική γνώση: πληροφορία σχετικά με την κατάστασή του (π.χ., ζωή)

Λήψη Αποφάσεων

- Αντιδραστική: Απόκριση

- Δένδρα απόφασης (Decision Trees)
- Μηχανές Πεπερασμένων Καταστάσεων (Finite State Machines – FSM)
- Δένδρα συμπεριφοράς (Behavior Trees)
- ...

- Προληπτική: Πλάνο

- Δεν θα μας απασχολήσει ιδιαίτερα σε αυτό το μάθημα
- Παραλλαγές του A^* στο χώρο των πράξεων επί της κατάστασης του εικονικού κόσμου
- Γίνεται ουσιαστικά από τους σχεδιαστές παιχνιδιού
- Παράδειγμα: ελεύθερη κατασκευή (Minecraft) ή εξερεύνηση (Skyrim)

Δένδρα Απόφασης

- Πολλές καταστάσεις μπορούν να περιγραφούν ως **if-then-else**
- Συστήματα κανόνων (production/expert) systems

Χαρακτηριστικά:

- Απλή τεχνική
- Εύκολη στην υλοποίηση και στην κατανόηση
- Αποδοτική
- Κυρίως αντιδραστική AI
- Μπορεί να συνδυασθεί με άλλες τεχνικές
- Μπορεί να εφαρμοσθεί Μηχανική Μάθηση για την εκμάθησή των δένδρων

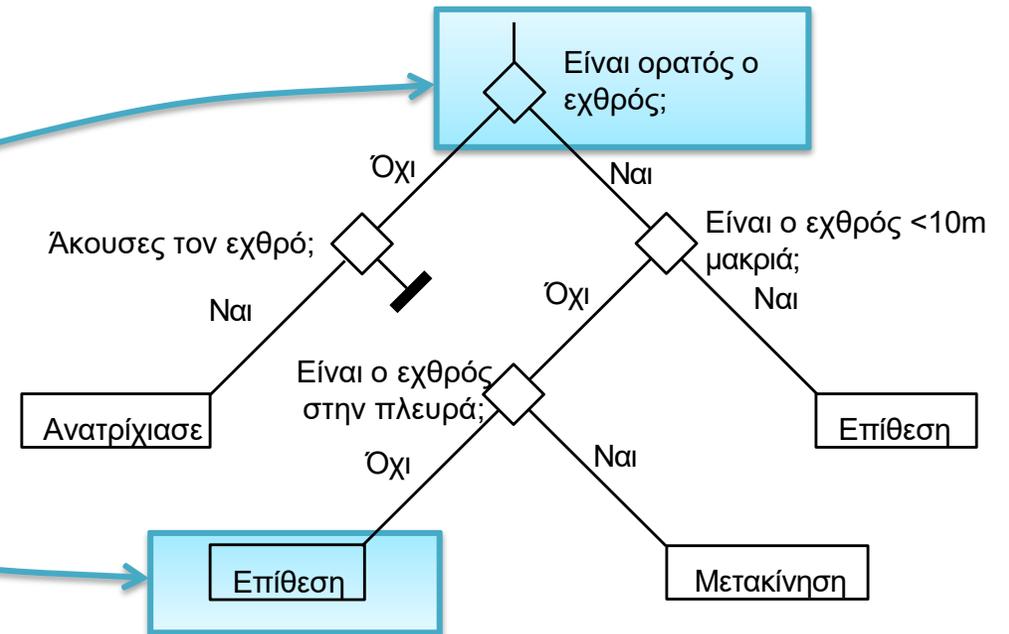
Παράδειγμα

Λήψη απόφασης
στρατιώτη

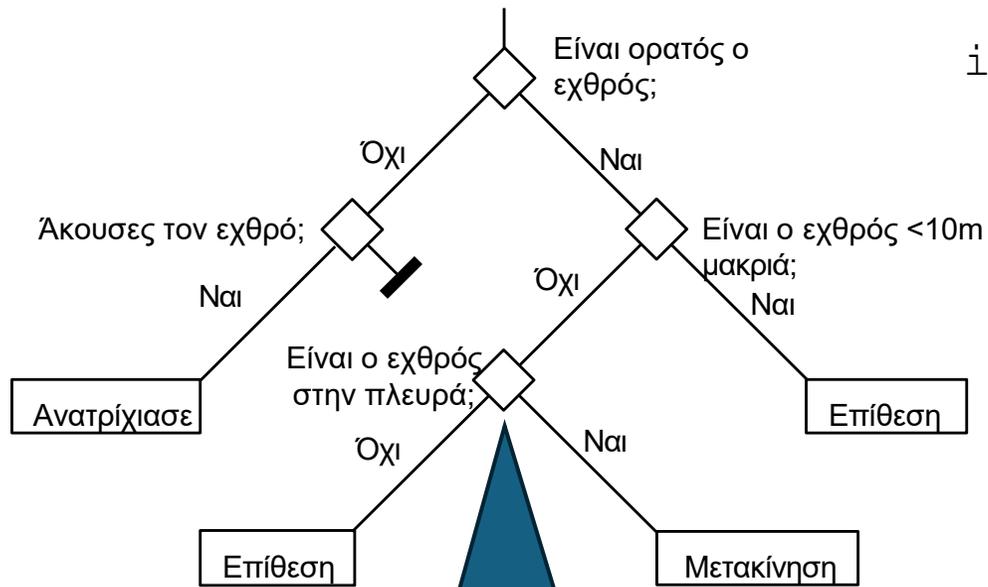
- Βασισμένη στην οπτική/ηχητική αντίληψή του

Έλεγχος
ιδιότητας

Δράση/Απόφαση/
Ταξινόμηση



Εύκολη Υλοποίηση



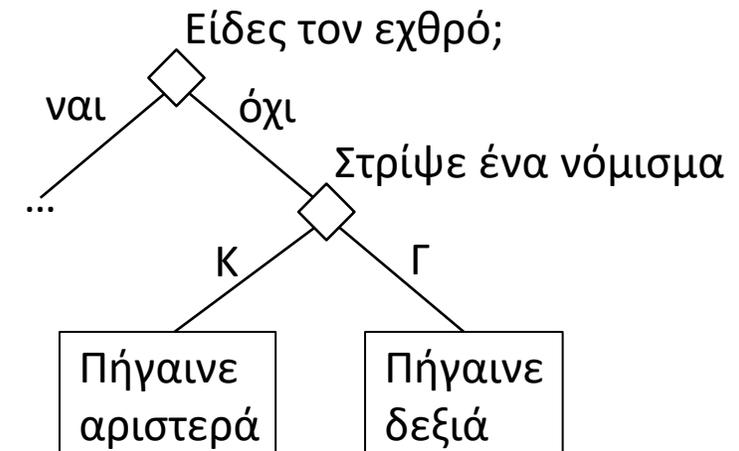
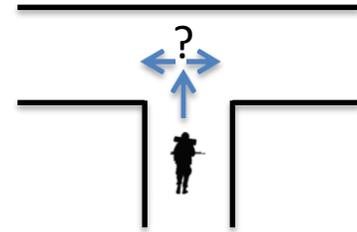
```
if (enemy.isVisible()) {  
    if (distance(player, enemy) < 10) {  
        attack();  
    }  
    else {  
        if (enemy.isOnFlank()) {  
            move();  
        }  
        else {  
            attack();  
        }  
    }  
}  
else { ... .. }
```

Αν θέλω να προσθέσω ένα επιπλέον έλεγχο: Έχει ο εχθρός πυροβόλο;

Η hard-coded γνώση μπορεί να μην είναι καλή ιδέα

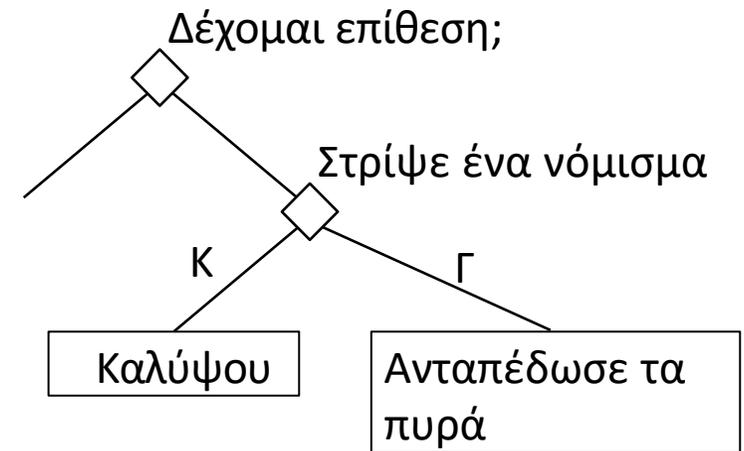
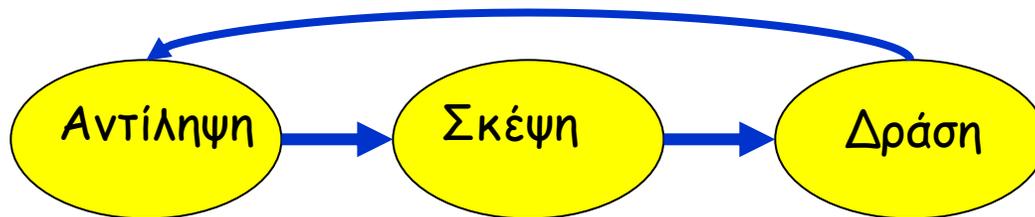
Παραλλαγές: Τυχαίες Αποφάσεις

- Η πλήρως προβλέψιμη συμπεριφορά είναι βαρετή
- Η τυχαιότητα δεν επιτρέπει μοτίβα
- Το τυχαίο νόμισμα μπορεί να είναι προκατειλημμένο (ψυχολογία παίκτη)



Επιμένοντας σε μία Απόφαση

- Συμπεριφορά στρατιώτη
- Ο κύκλος εξερευνά το δένδρο κάθε φορά
- Αν σε κάθε κύκλο παίρνουμε τυχαία απόφαση τότε ο στρατιώτης δεν θα λειτουργεί σωστά.



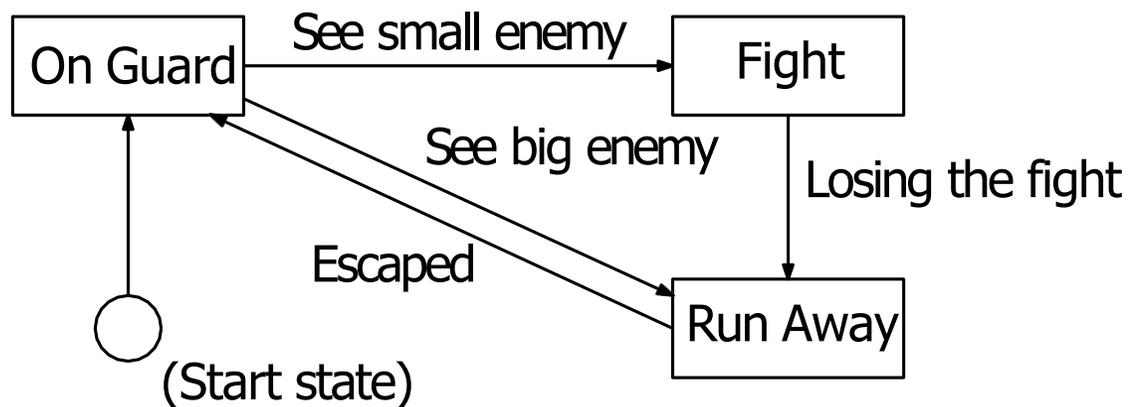
Επέμεινε στην
απόφαση (για λίγο)

Μηχανές Πεπερασμένων Καταστάσεων

FSM (Finite State Machine)

- Οργανώνει τις συμπεριφορές σε ένα γράφημα
- Μεταβάσεις με βάση την κατάσταση του παιχνιδιού

- Οι καταστάσεις μπορεί να είναι:
 - Συμπεριφορά
 - Συναισθηματική κατάσταση
 - Δράσεις
 - Προδιάθεση



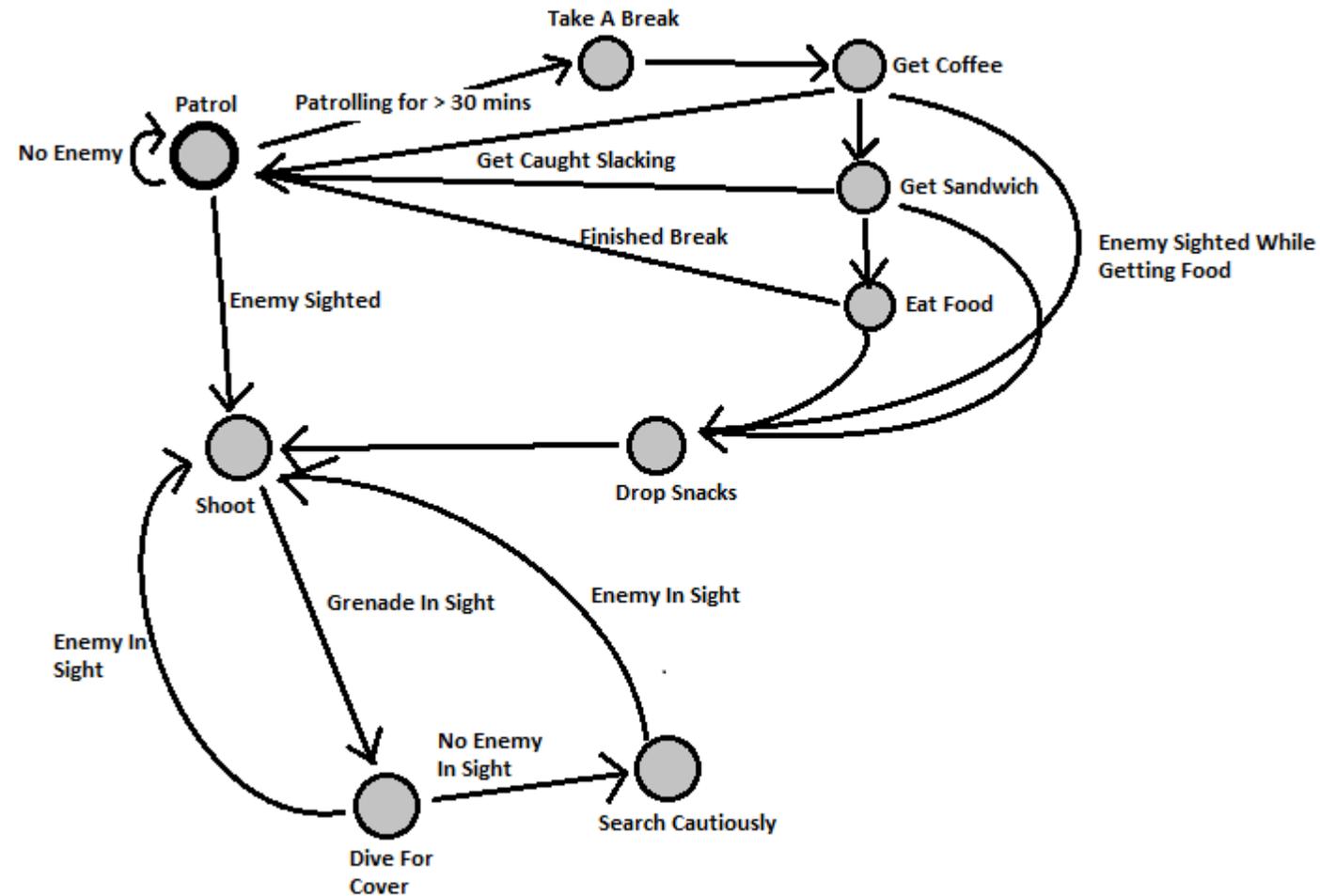
(a)

| Start | Small enemy | Big enemy | Losing | Escaped |
|-------|-------------|-----------|--------|---------|
| Guard | Fight | Run | — | — |
| Fight | — | — | Run | — |
| Run | — | — | — | Guard |

The table is a transition table for the FSM. The columns represent the current state (Start, Guard, Fight, Run) and the rows represent the next state (Guard, Fight, Run). The cells contain the next state or a dash (—) indicating no transition.

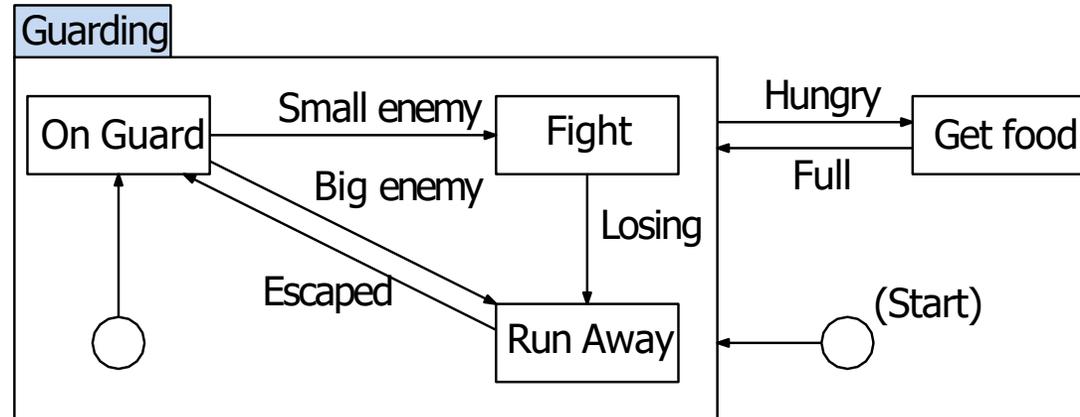
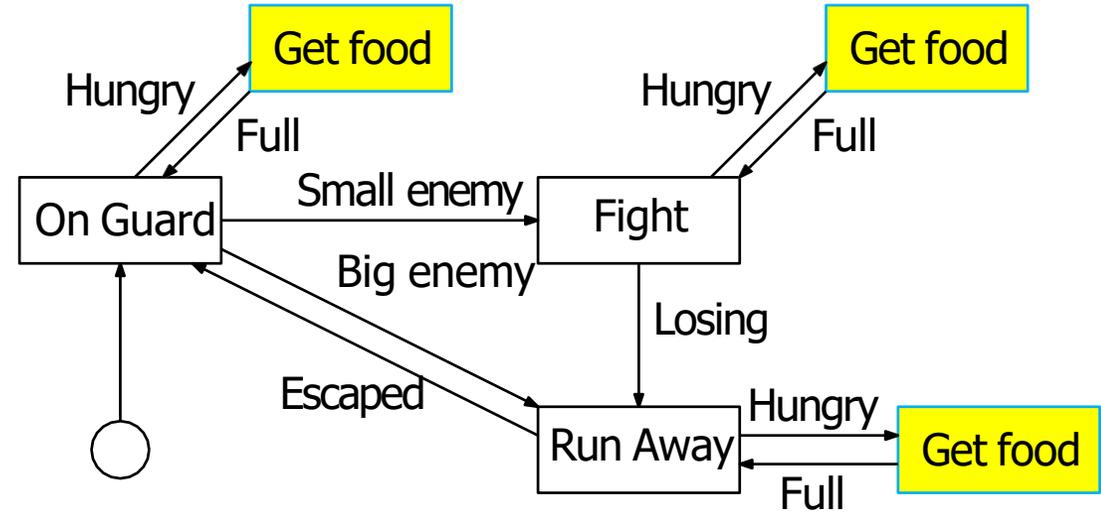
(b)

Παράδειγμα



Μειώνοντας την Πολυπλοκότητα σε FSM

- Κατάσταση με μεταβάσεις προς πολλές άλλες
- Ιεραρχικά FSM

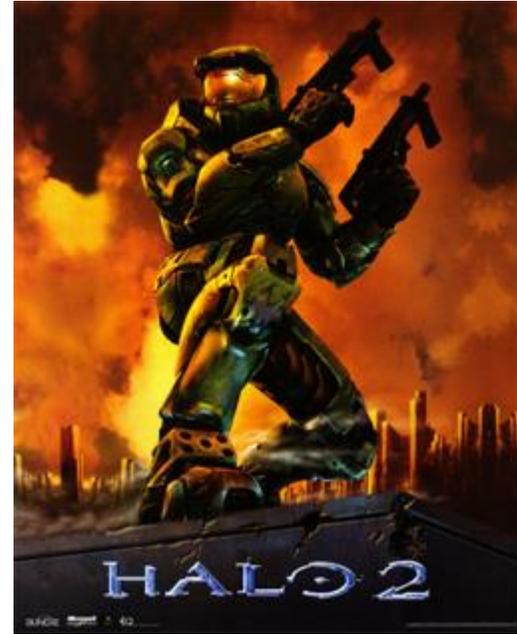


FSM: Γενικά

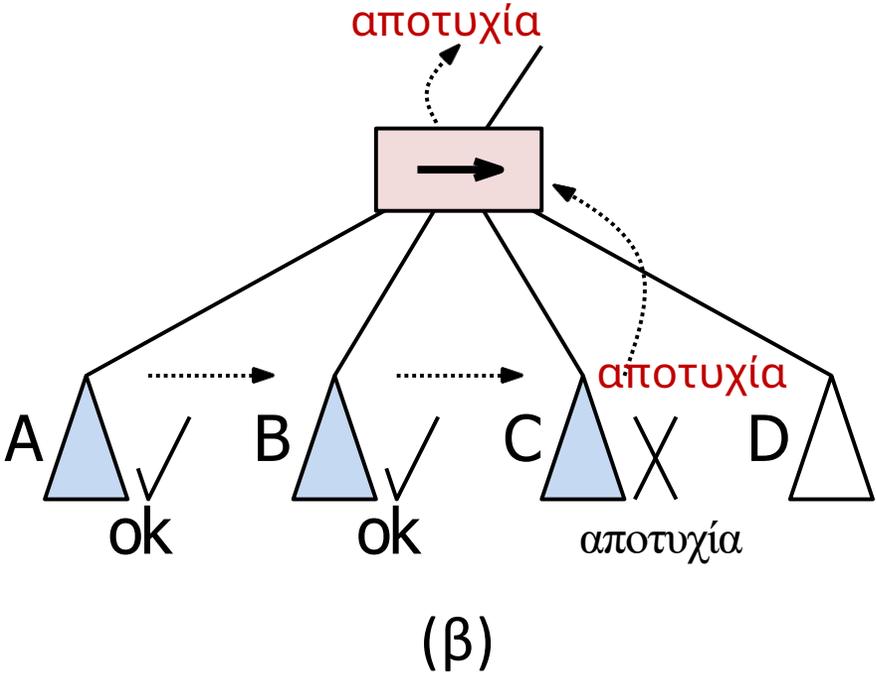
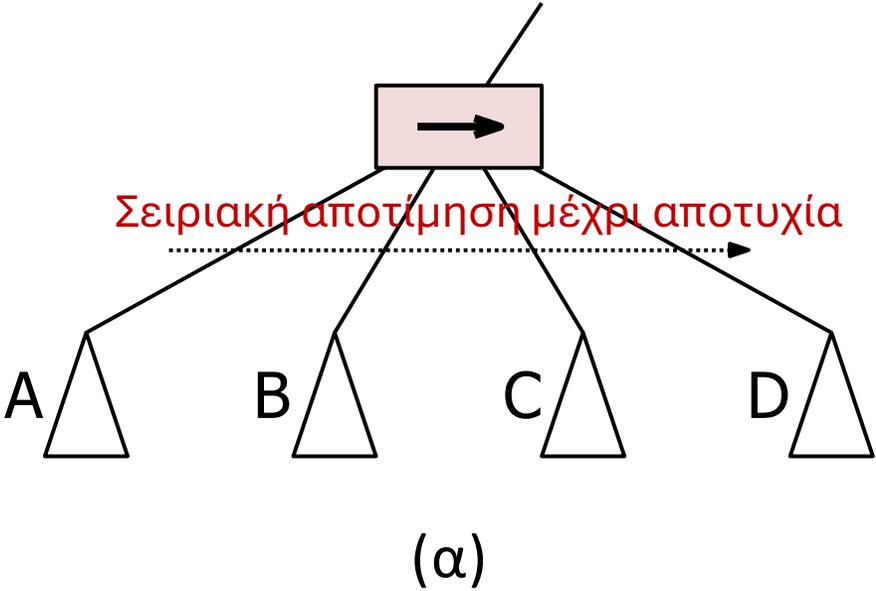
- Το πιο συνηθισμένο μοτίβο λογισμικού AI για βιντεοπαιχνίδια
 - Φυσική αντιστοιχία μεταξύ καταστάσεων και συμπεριφορών
 - Εύκολη δημιουργία διαγράμματος
 - Ευκολία στον προγραμματισμό
 - Ευκολία στον εντοπισμό σφαλμάτων
 - Εντελώς γενικό για οποιοδήποτε πρόβλημα
- Προβλήματα
 - Πολλές καταστάσεις
 - Υπερβολικά προβλέψιμο
 - Συχνά δημιουργείται με ad hoc δομή

Δένδρα Συμπεριφοράς

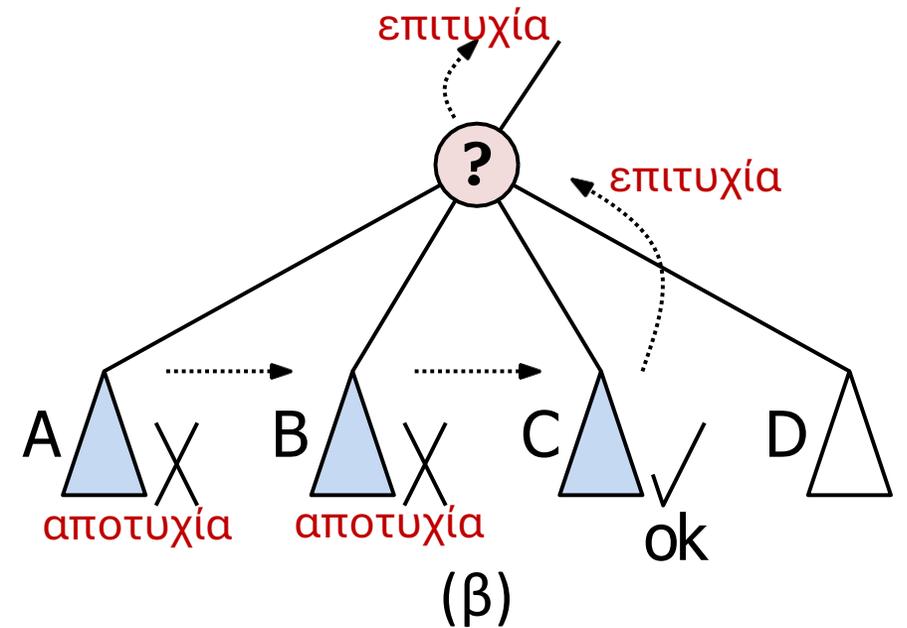
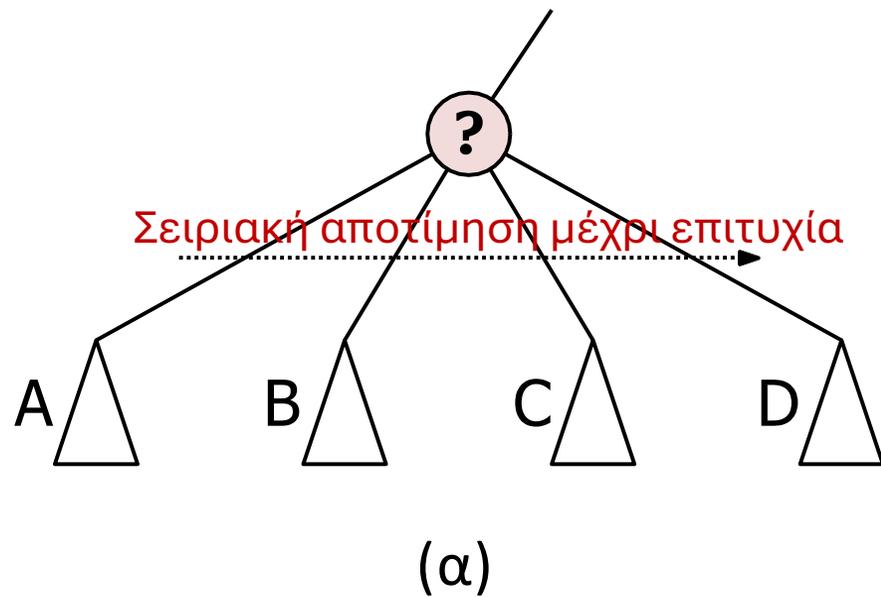
- Χρησιμοποιήθηκε πρώτη φορά στο Halo 2 (2004)
- Ένας ελαφρύς τρόπος σχεδίασης πλάνων δράσης
- Πλάνο:
 - Ακολουθία δράσεων:
 - 1) Πήγαινε στη πόρτα
 - 2) Χρήση κλειδιού για άνοιγμα της πόρτας
 - 3) Πέρασε μέσα από την πόρτα
 - Με προϋποθέσεις:
 - 2)* Πρέπει να έχει το κλειδί



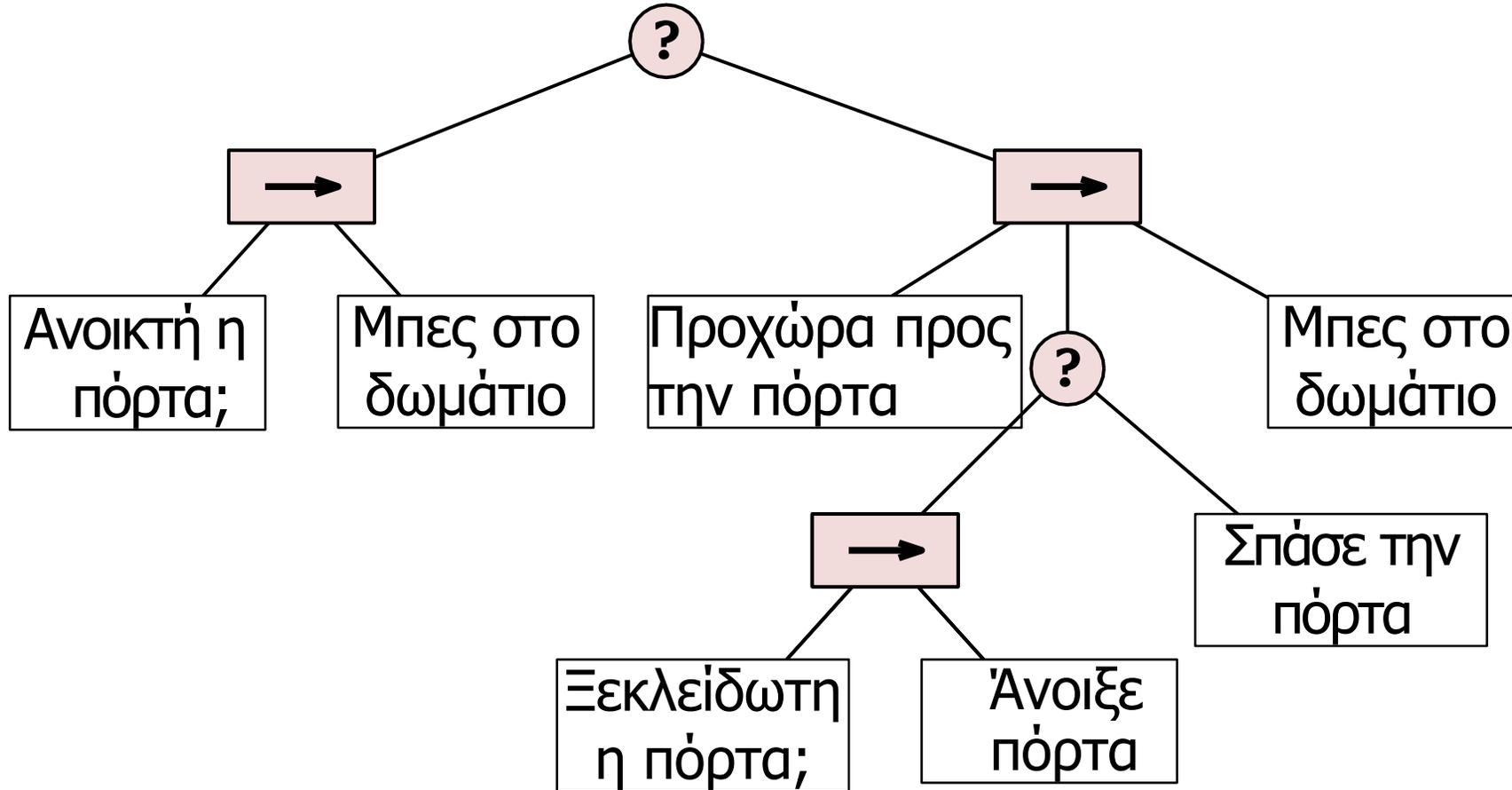
Σειριακός κόμβος AND



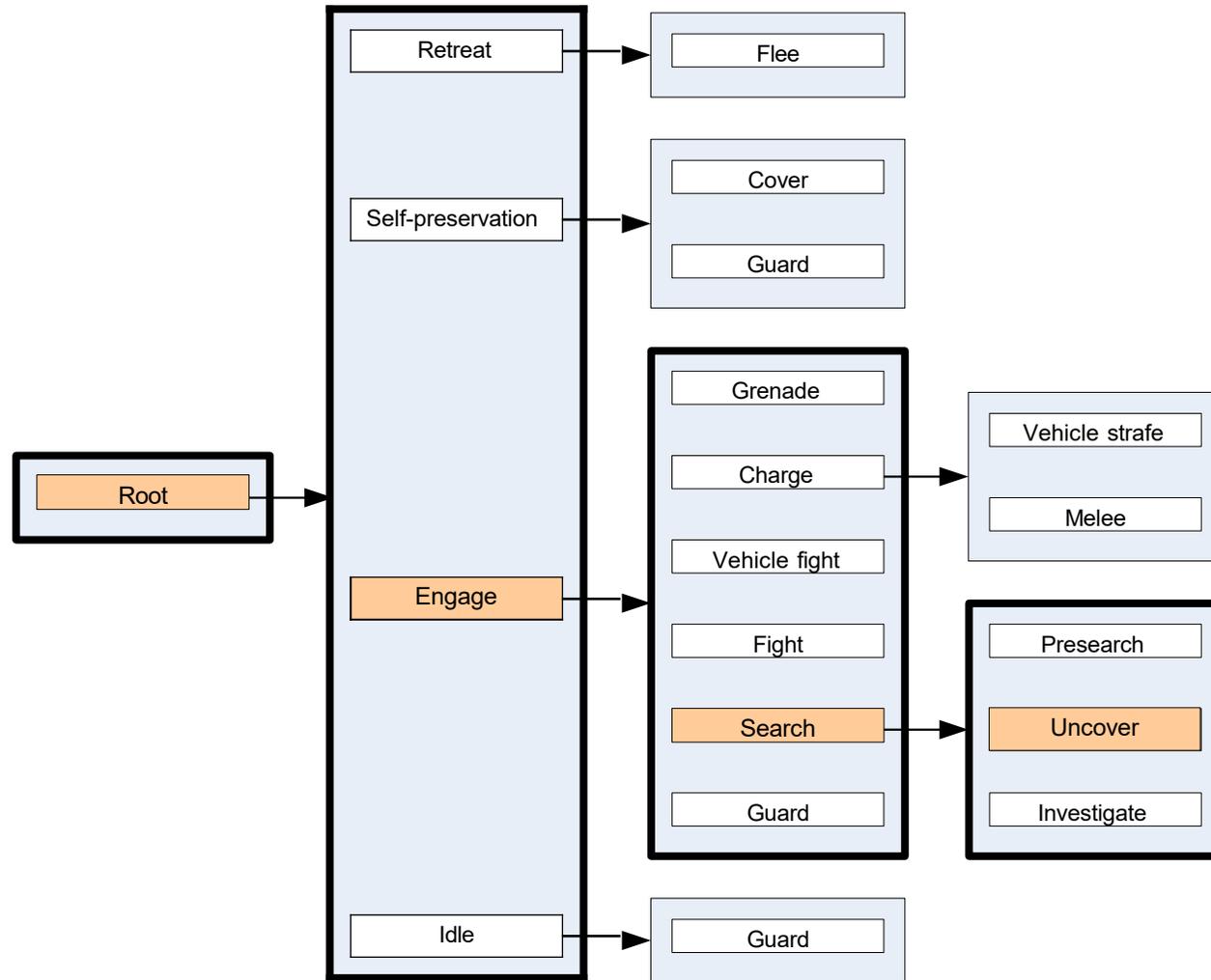
Σειριακός Κόμβος OR

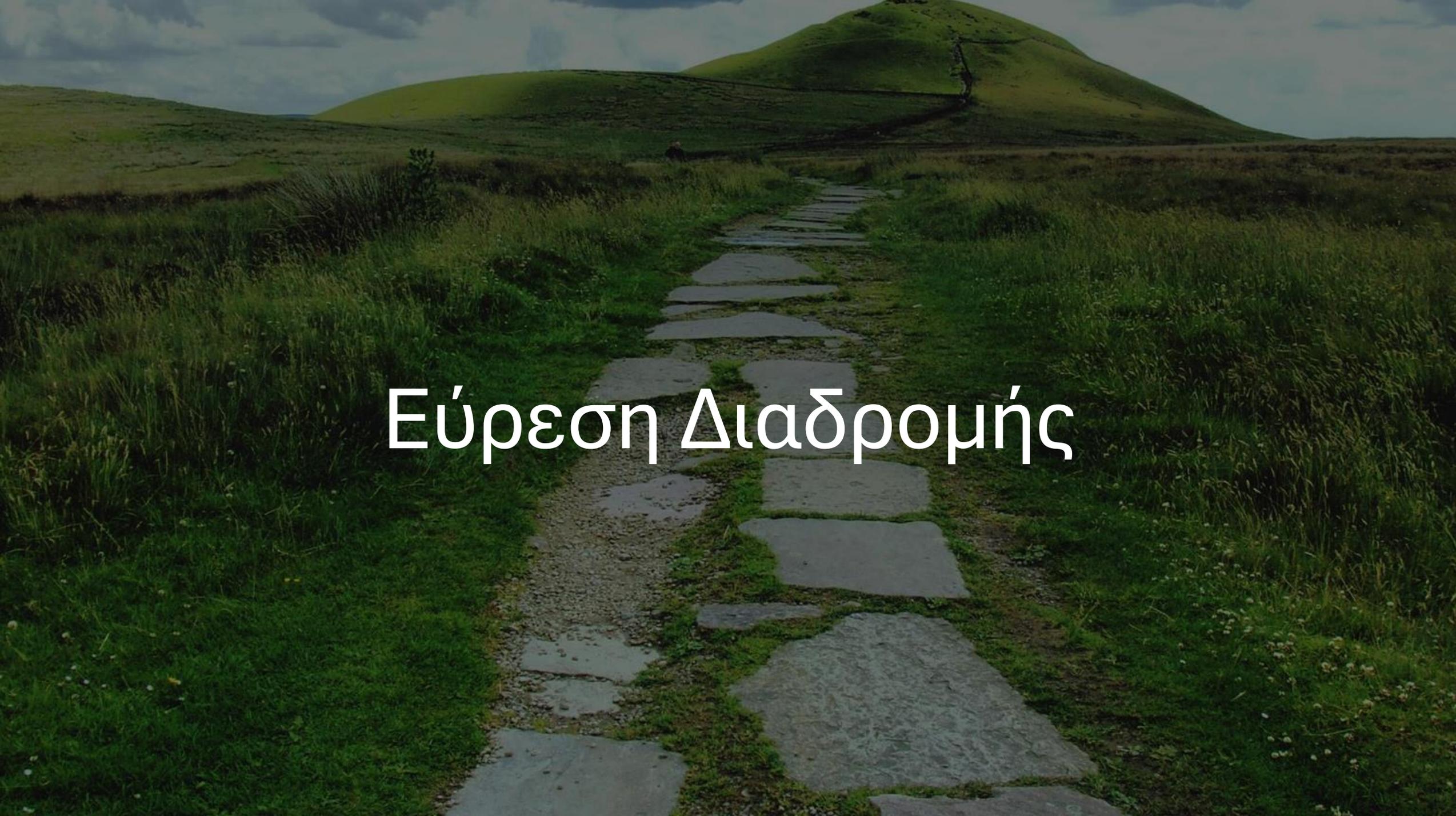


Στόχος: Μετακίνηση στο Δωμάτιο



HALO 2: Λήψη Αποφάσεων

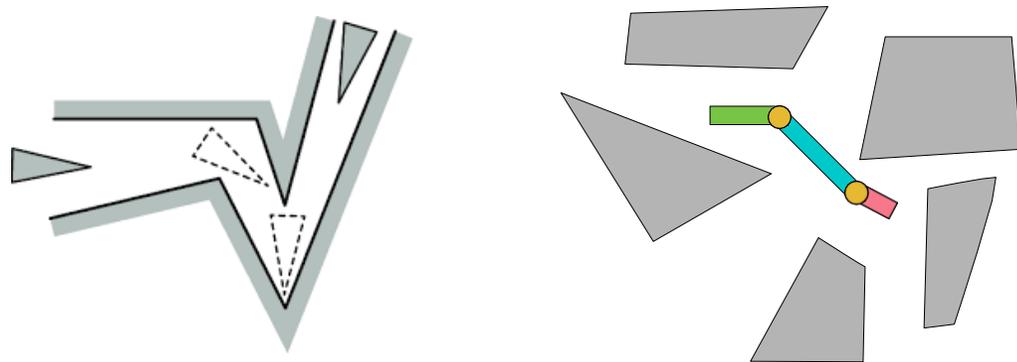


A scenic landscape featuring a stone path that winds through a lush, green field. The path is composed of large, flat, grey stones set in a line. The surrounding vegetation is dense and vibrant green. In the background, a large, rounded green hill rises against a sky filled with soft, grey clouds. The overall atmosphere is serene and natural.

Εύρεση Διαδρομής

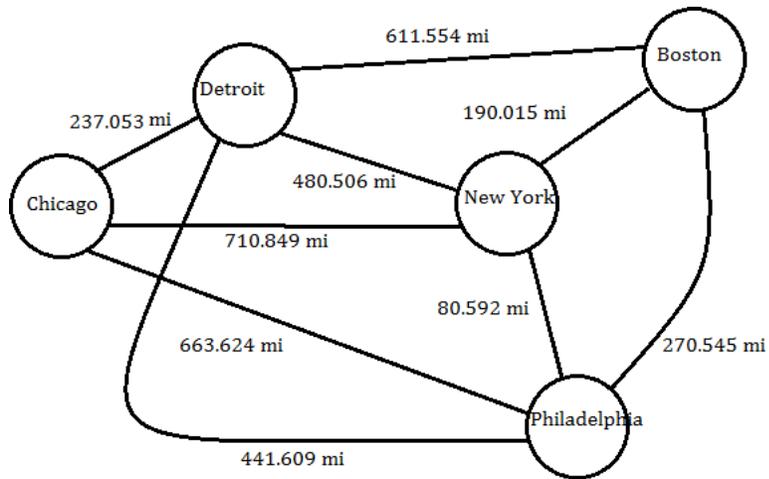
Προβλήματα Πλοήγησης

- Πλοήγηση μεταξύ τοποθεσιών
- Πλοήγηση πυκνού πλήθους
- Συντονισμένη ομαδική κίνηση
- Καταδίωξη
- Κίνηση πολύπλοκου/αρθρωτού σχήματος
 - Πρόβλημα μετακίνησης πιάνου (άκαμπτο)
 - Σκελετός (αθρωτό)



Χρήση Χάρτη

- Δημιουργία γραφήματος
- Χρήση αλγόριθμου Dijkstra για συντομότερη διαδρομή



Υπόθεση: Ο κόσμος είναι άκαμπτος και περιορισμένες λωρίδες κίνησης

Τί είναι Συντομότερη Διαδρομή;

- Απόσταση;
- Ταχύτητα;
- Κόστος ενέργειας;
- Έκθεση στον εχθρό;

- Εξαγωνικός χάρτης
 - 6 κατευθύνσεις
 - Τύποι εδάφους → κόστος στην ταχύτητα



Ανοικτό Έδαφος

- Μίξη εμποδίων και ανοικτών χώρων
- Περισσότερες επιλογές για κατευθύνσεις
- Πώς επιλέγουμε διαδρομή;



Εύρεση Διαδρομής σε Βιντεοπαιχνίδια

Οι χαρακτήρες στο παιχνίδι συνήθως κινούνται μέσα στο επίπεδο

- Φρουροί σε ένα stealth παιχνίδι
- Μονάδες σε RTS



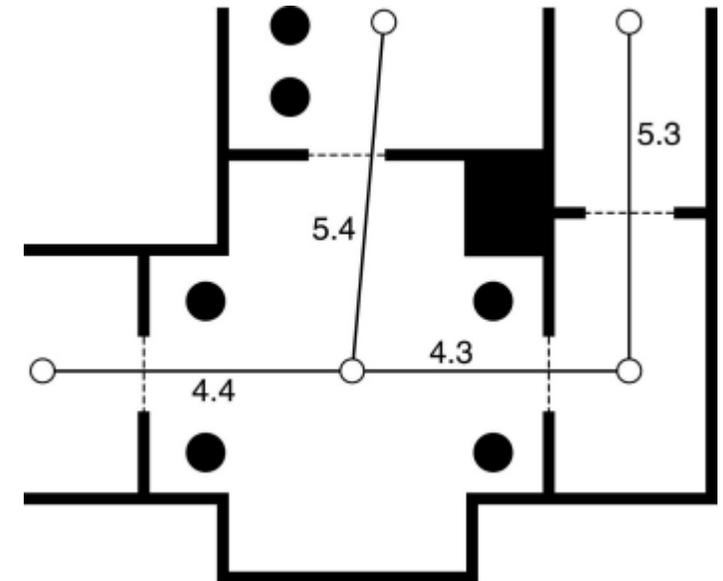
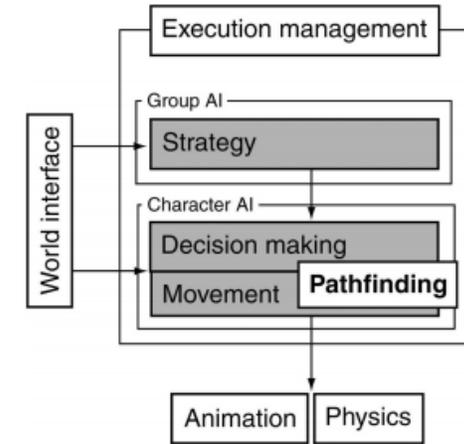
Είναι εύκολο για τον παίκτη να κοροϊδέψει τη προδιαγεγραμμένη κίνηση

Η τυχαία περιπλάνηση συνήθως δεν είναι ρεαλιστική και ταυτόχρονα είναι εύκολη για τον παίκτη



Αλγόριθμοι Εύρεσης Διαδρομής

- Η πλειοψηφία των παιχνιδιών χρησιμοποιεί κάποια παραλλαγή του A^*
- Ο αλγόριθμος του Dijkstra χρησιμοποιείται σε AI που αφορά την τακτική
- Οι αλγόριθμοι αυτοί δεν λειτουργούν απευθείας στη γεωμετρία αλλά σε μία απλοποιημένη αναπαράστασή του βασισμένη σε γραφήματα
- Συνήθως χρησιμοποιούνται κατευθυνόμενα γραφήματα με μη-αρνητικά βάρη



Πλέγματα

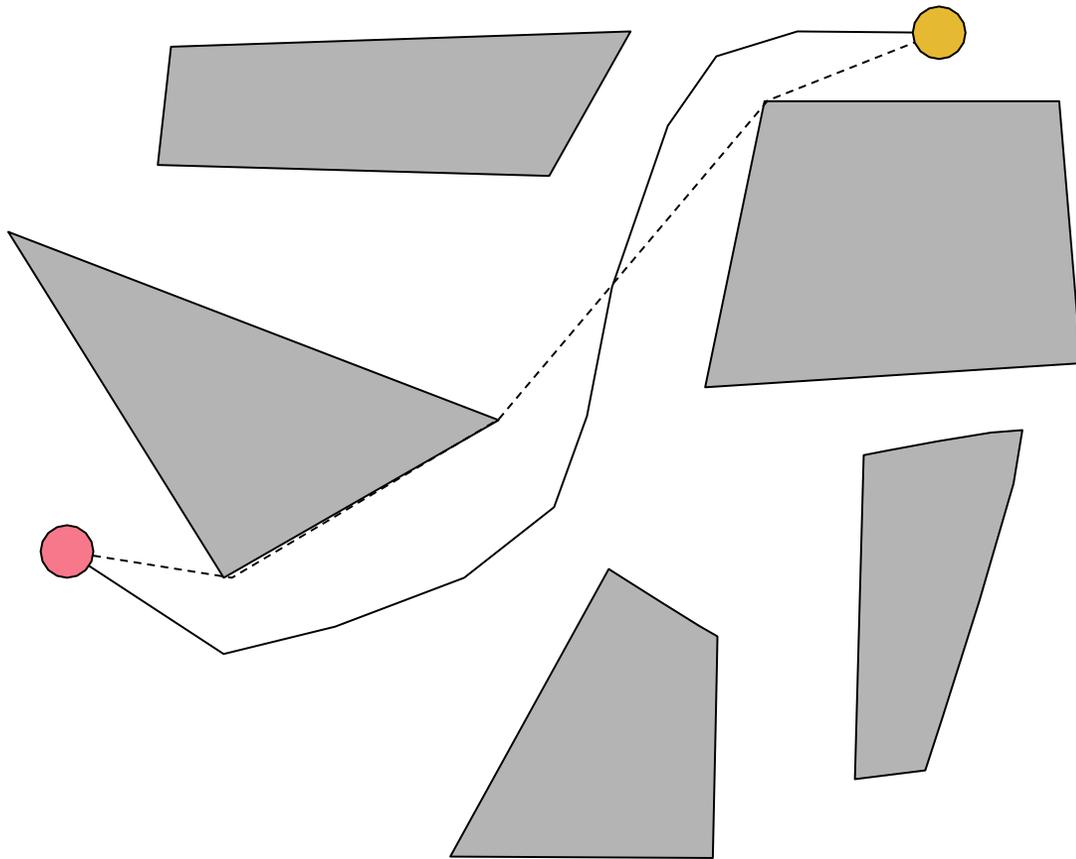
- 2d πλέγματα:
 - Δουλεύει αρκετά καλά για πολλά παιχνίδια όπως το *Warcraft III*
- Κάθε κελί επισημαίνεται ως:
 - Προσπελάσιμο ή απροσπέλαστο
- Κάθε αντικείμενο στον εικονικό κόσμο μπορεί να καταλάβει ένα ή περισσότερα κελιά
- Χαρακτηριστικά:
 - Γρήγορη Αναζήτηση
 - Εύκολη προσπέλαση σε γειτονικά κελιά
 - **Πλήρης** αναπαράσταση του επιπέδου



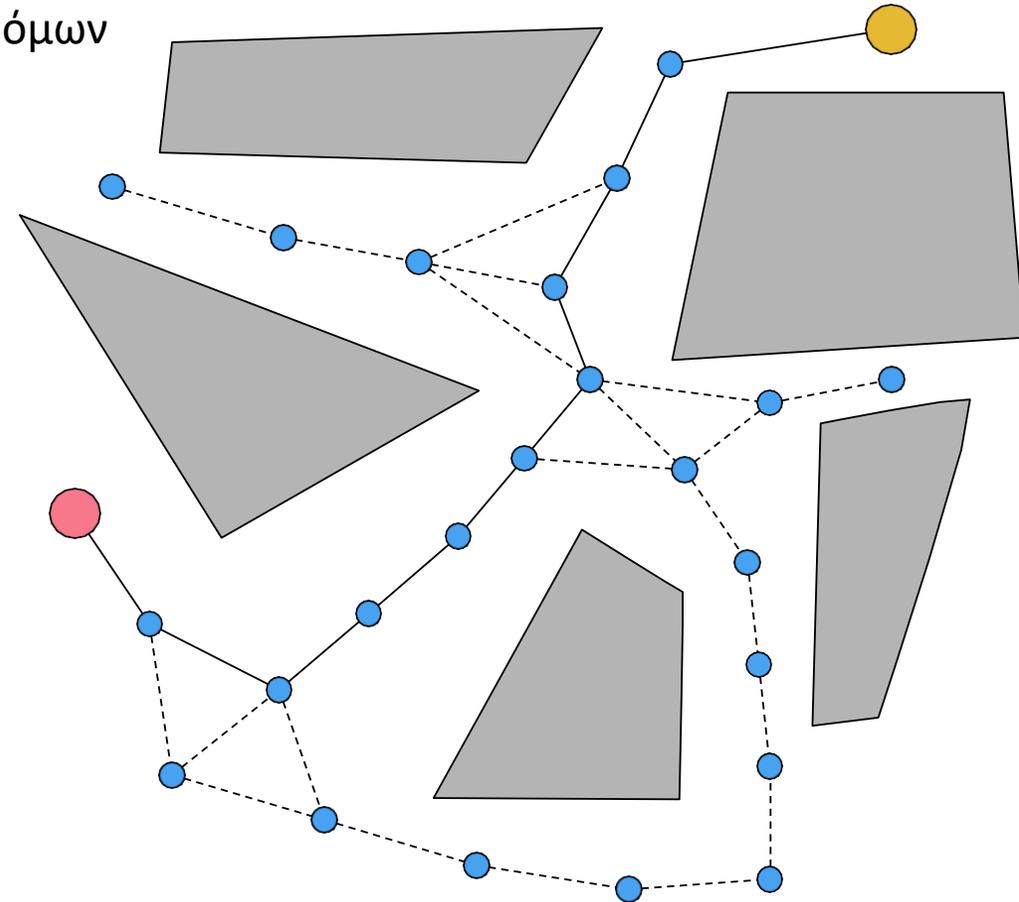
Επιλογή Καλύτερης Διαδρομής

Χρειάζονται πολλά σημεία και δύσκολη η σχεδίαση συντονισμένης ομαδικής κίνησης

Η συντομότερη μπορεί να μην είναι η πιο φυσική διαδρομή (διακεκομμένη γραμμή)

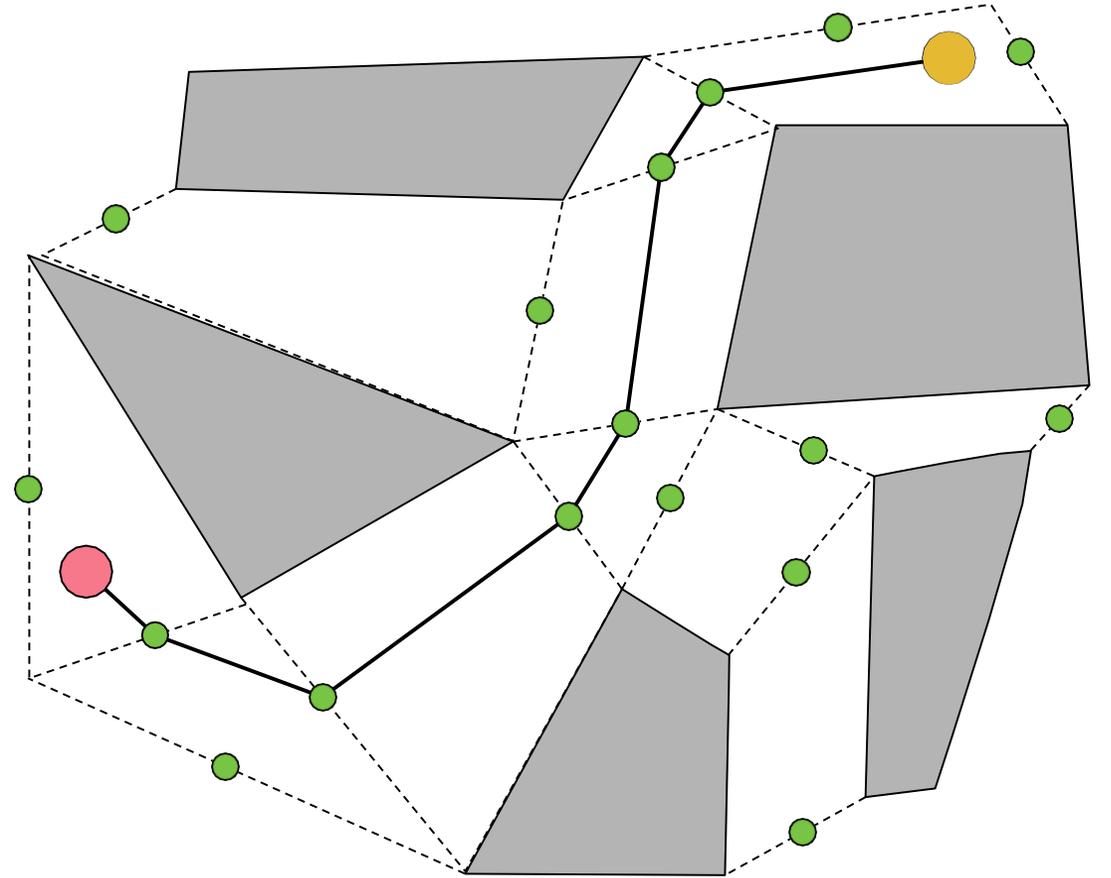


- Προεπεξεργασία χώρου σε γράφημα ενδιάμεσων σημείων
- Τοποθέτηση σημείων στη μέση των φυσικών διαδρόμων



Άλλη λύση: Navmesh

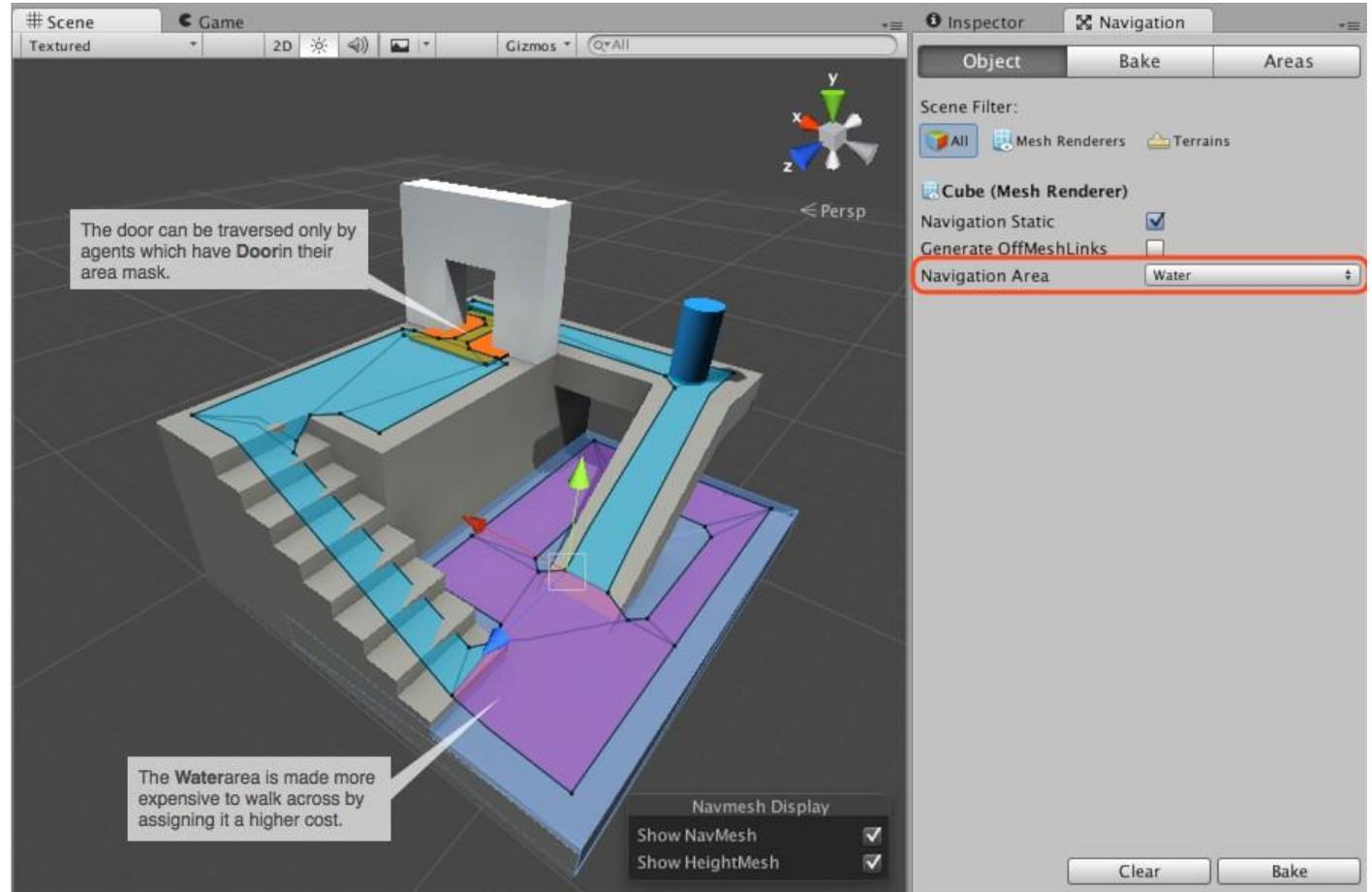
- Προεπεξεργασία χώρου σε ένα πολυγωνικό πλέγμα ελεύθερων χώρων
- Σχεδίαση κίνησης μεταξύ πολυγώνων
 - Μεταξύ ακμών
 - Μεταξύ κέντρων
- Πολλαπλών επιπέδων αναζήτηση
 - Πρώτα μεταξύ περιοχών
 - Έπειτα επιλογή σημείου εισόδου και σημείου εξόδου



Navmesh

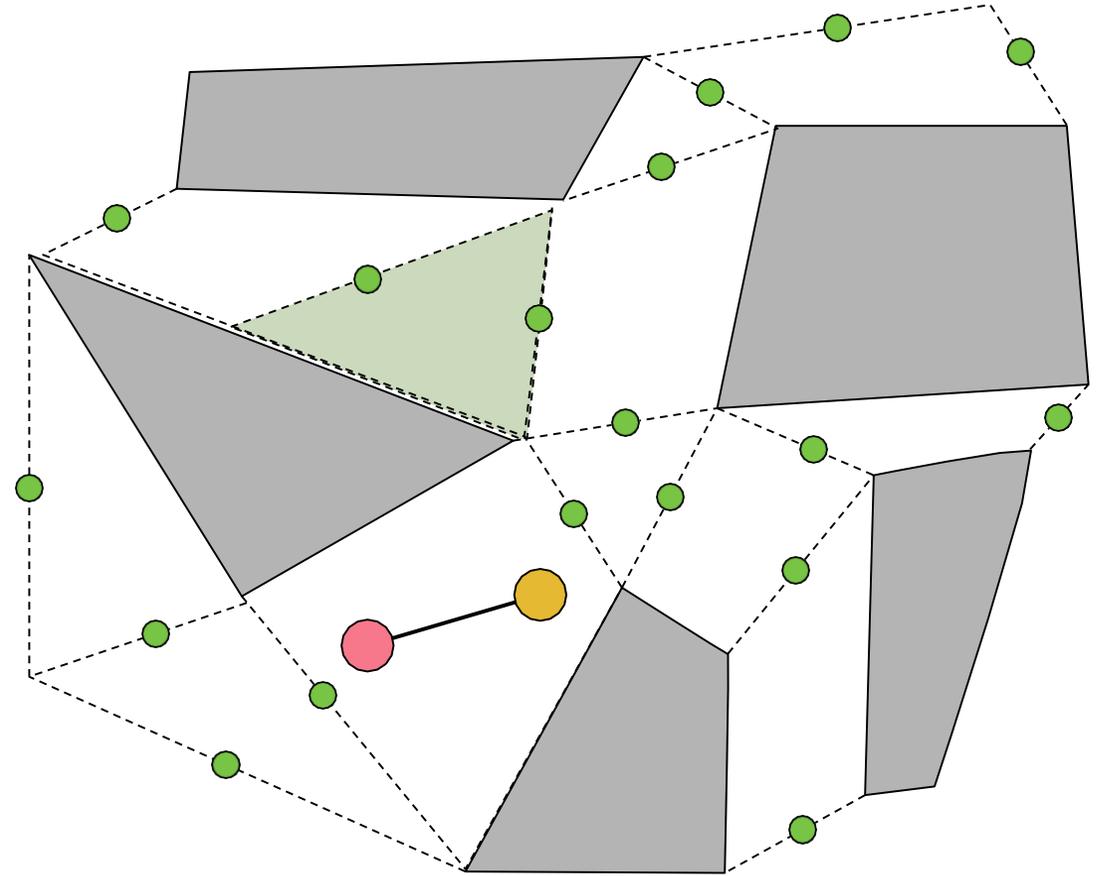
Μπορούν να:

- Έχουν διαφορετικούς τύπους εδάφους
- Καθορίζονται από «πύλες»
- Να είναι 2d τοπολογικά πολύπλοκες επιφάνειες



Χρήση Navmesh

- Μέσα στην περιοχή χρησιμοποιήσε ευθεία διαδρομή
- Οι περιοχές (και υποπεριοχές) μπορούν να έχουν ως χαρακτηριστικό διαφορετικούς τύπους εδάφους με διαφορετικό κόστος.



Δημιουργία Navmesh από το έδαφος

Βήμα 1: Εύρεση προσιτών επιφανειών

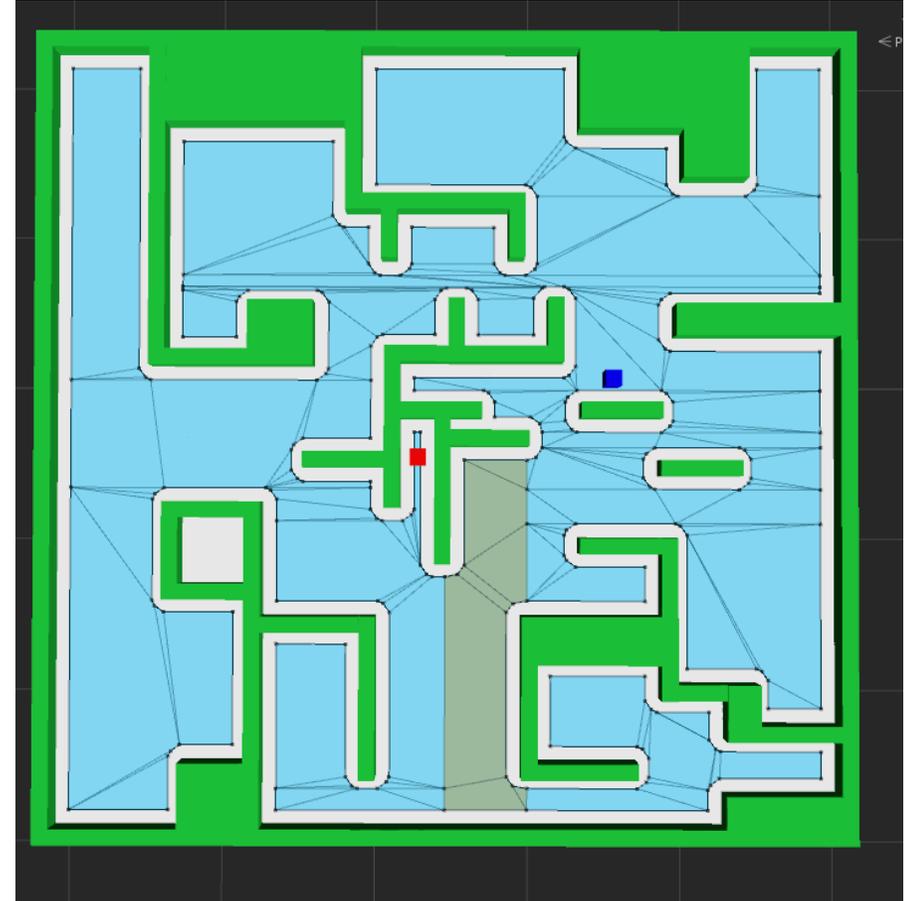
- Αναπαρίσταται ως πολυγωνικός χάρτης

Βήμα 2: Απλοποίηση συνόρων

- Απλοποίηση του πολυγωνικού χάρτη

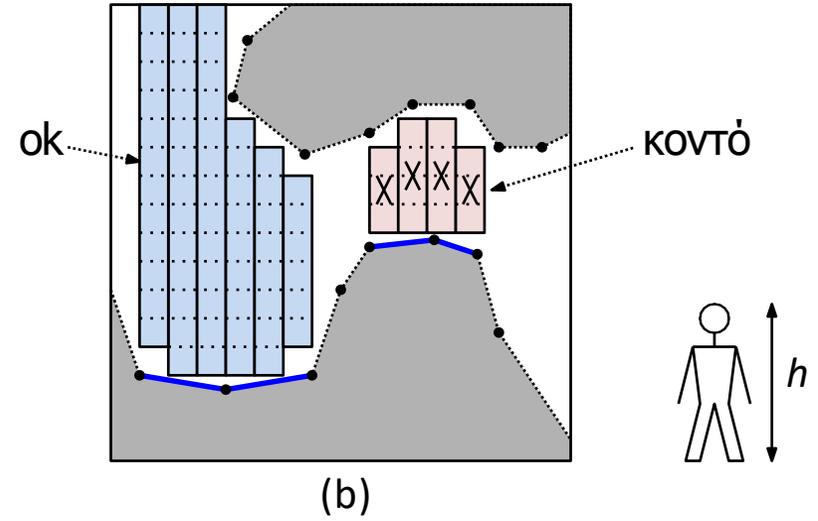
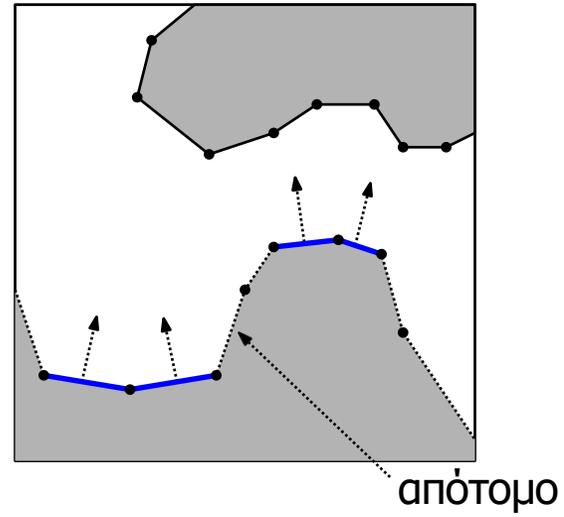
Βήμα 3: Τριγωνοποίηση χάρτη

- Κάλυψη τον χάρτη με τρίγωνα

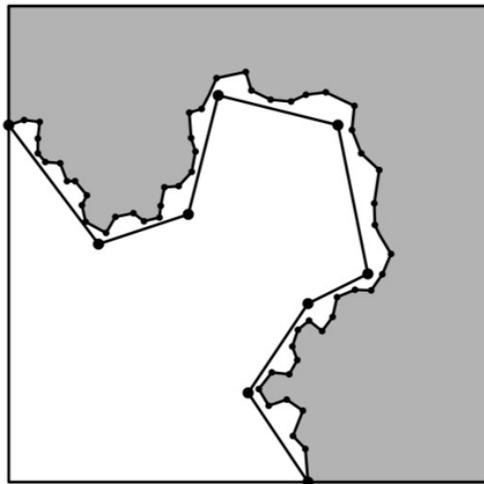


Βήματα

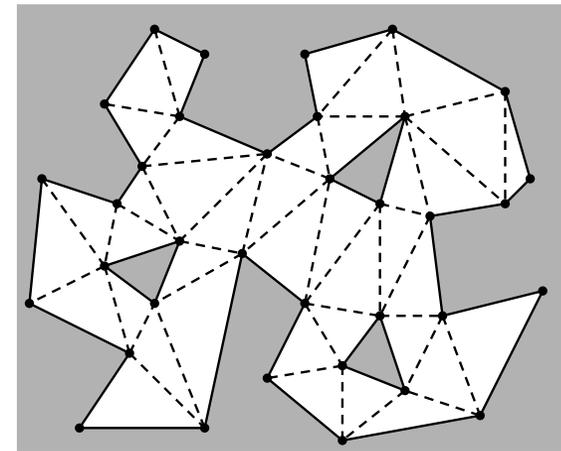
(1)



(2)

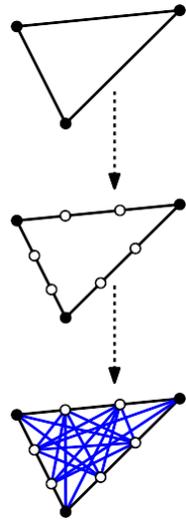


(3)

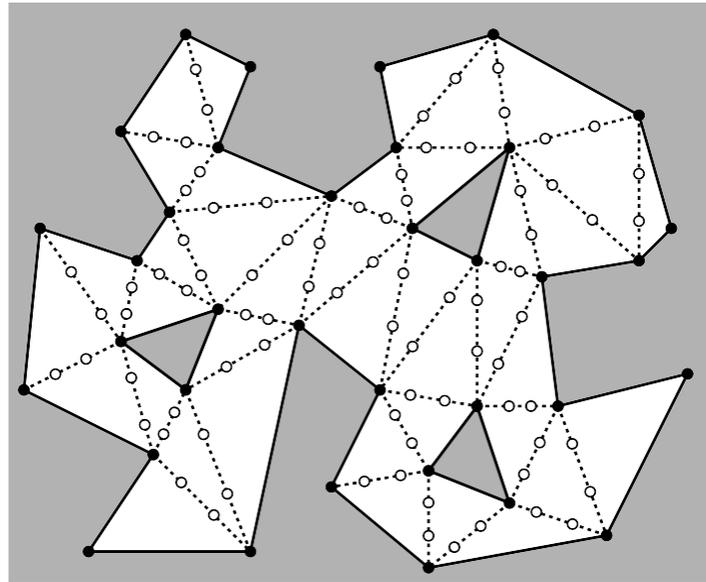


Χρήση Navmesh: Εύρεση Διαδρομής

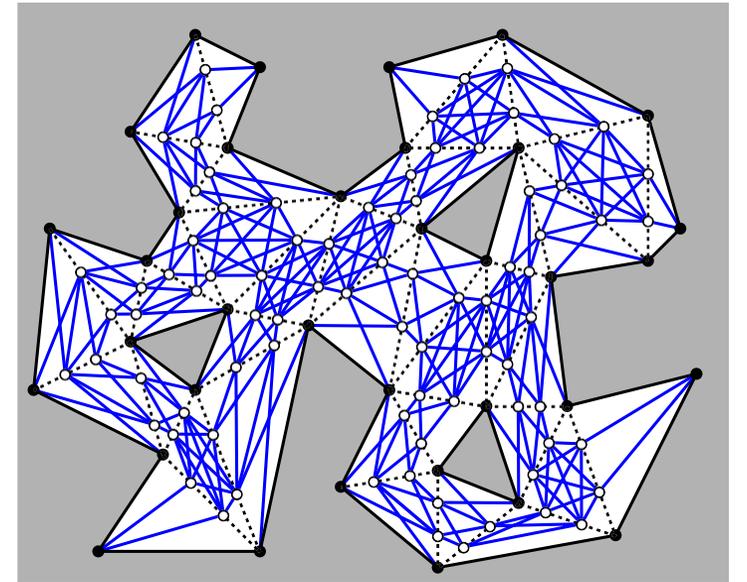
- Διακριτοποίηση με πρόσθεση σημείων
- Αναζήτηση συντομότερης διαδρομής



(a)



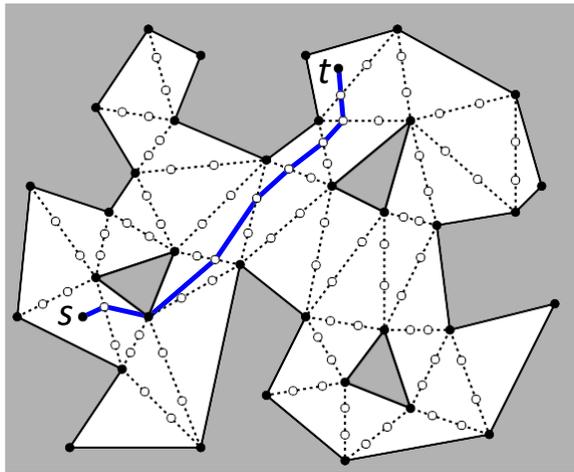
(b)



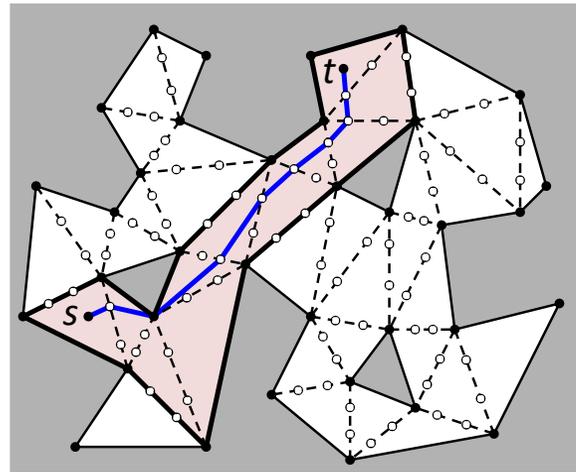
(c)

Χρήση Navmesh: Βελτίωση

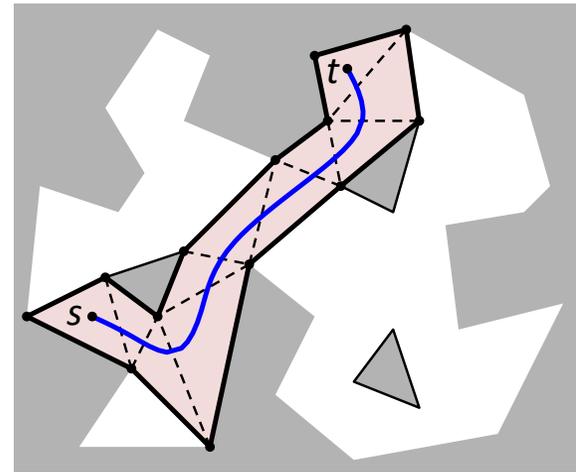
Ομαλοποίηση και καθάρισμα διαδρομής



(a)



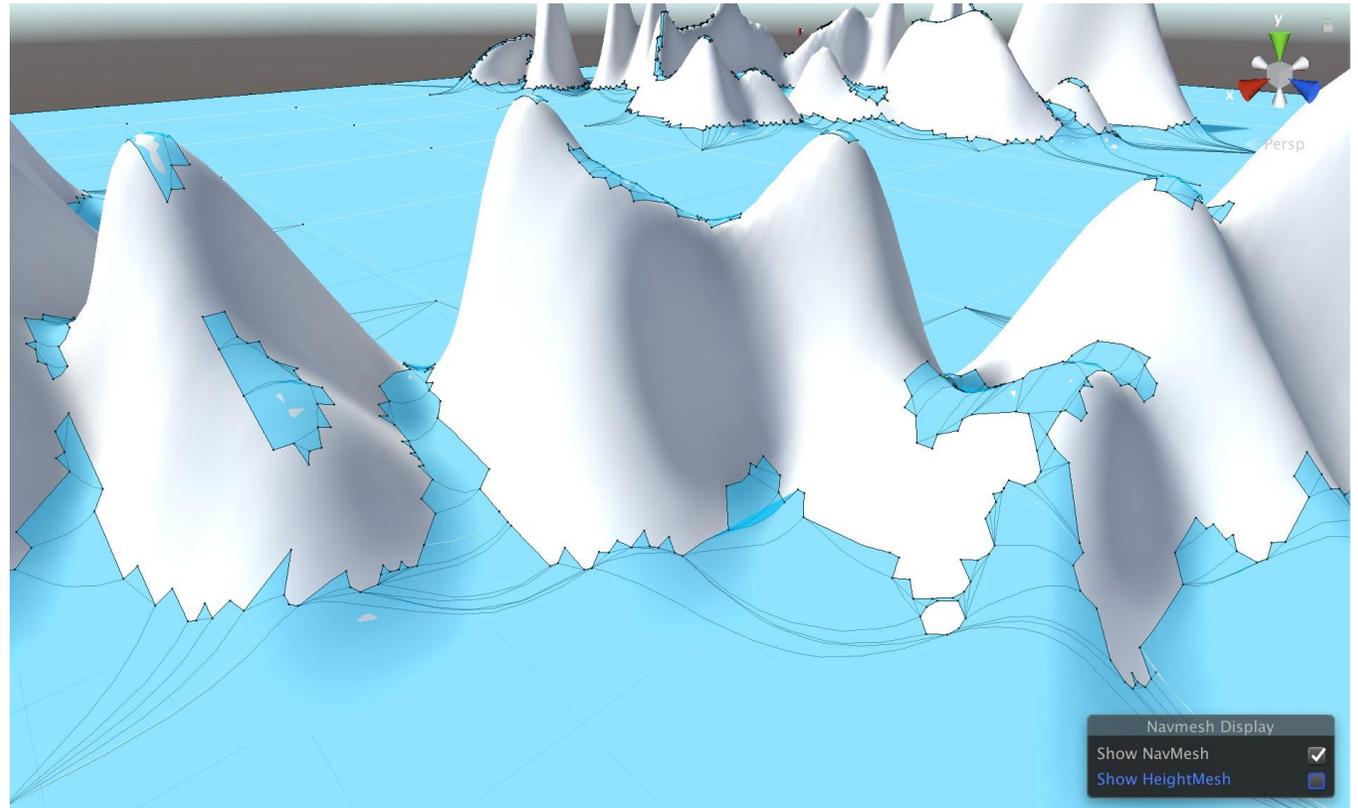
(b)



(c)

Navmesh: Unity

- Δημιουργία εδάφους
 - Terrain editor
- Ιδιότητες πλοήγησης του πράκτορα
 - Ύψος, πλάτος
 - Μέγιστη κλίση
- Κάψιμο (Bake) του Navmesh
 - Εύρεση προσιτών περιοχών
 - Δημιουργία πλέγματος

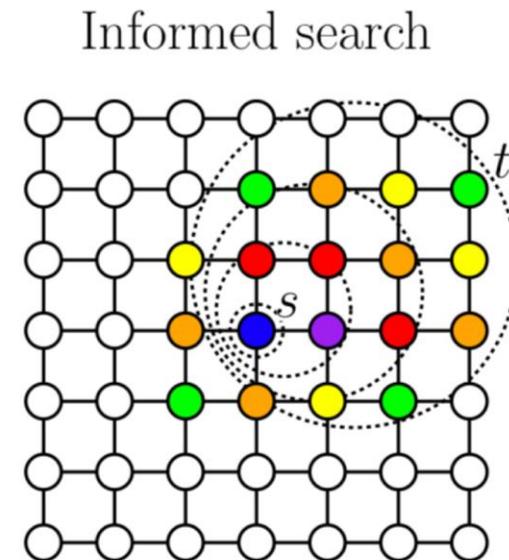
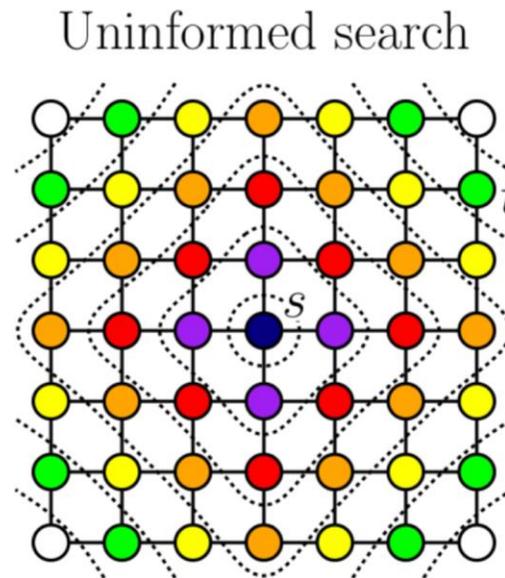


Μη-ενημερωμένη έναντι Ενημερωμένης Αναζήτησης

- Μη ενημερωμένη
 - Επιλογή κόμβου με βάση την απόσταση

- Ενημερωμένη –
πρόσθεση προτίμησης
προς τον προορισμό

- Ευρετική προσέγγιση
 - Επέλεξε τον επόμενο κόμβο με βάση την απόσταση προς τον προορισμό



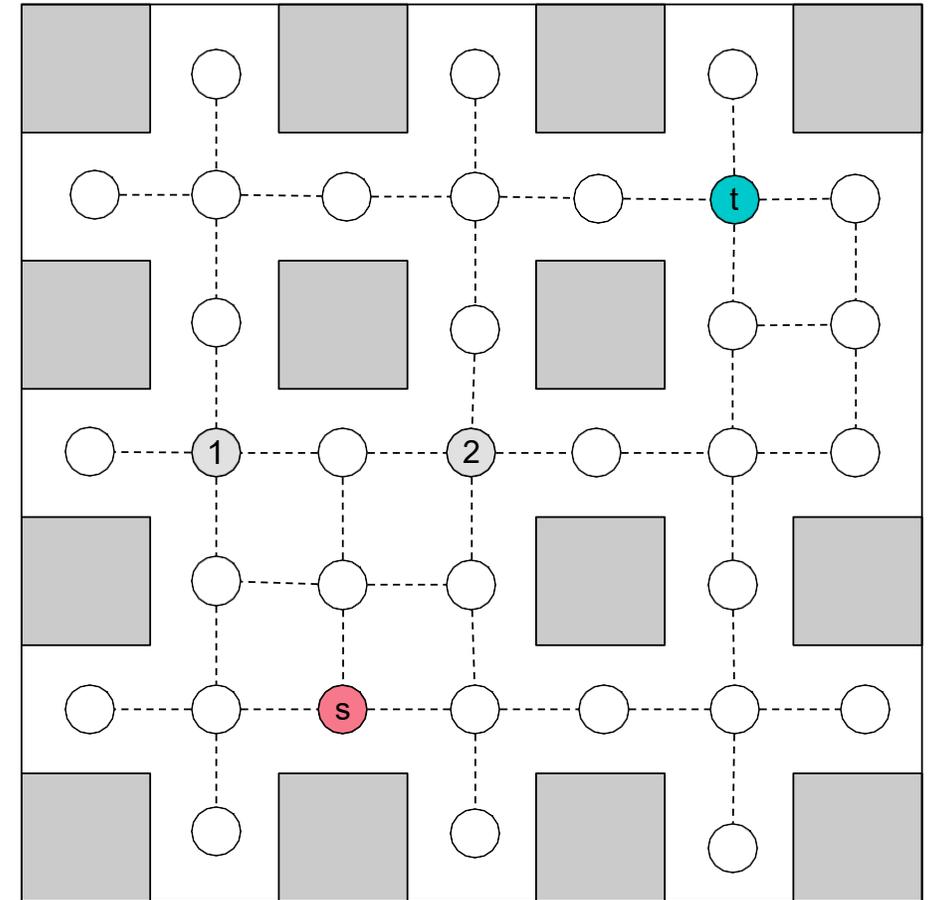
Ενημερωμένη Αναζήτηση Διαδρομής

Συναρτήσεις απόστασης

- $w(u, v)$: απόσταση u σε v
- $d[u]$: απόσταση από αρχή s στον u
- $dist(u, t)$: απόσταση από u στον t

- $w(s, 1) = 3$ $dist(1, t) = 6$
- $w(s, 2) = 3$ $dist(2, t) = 4$

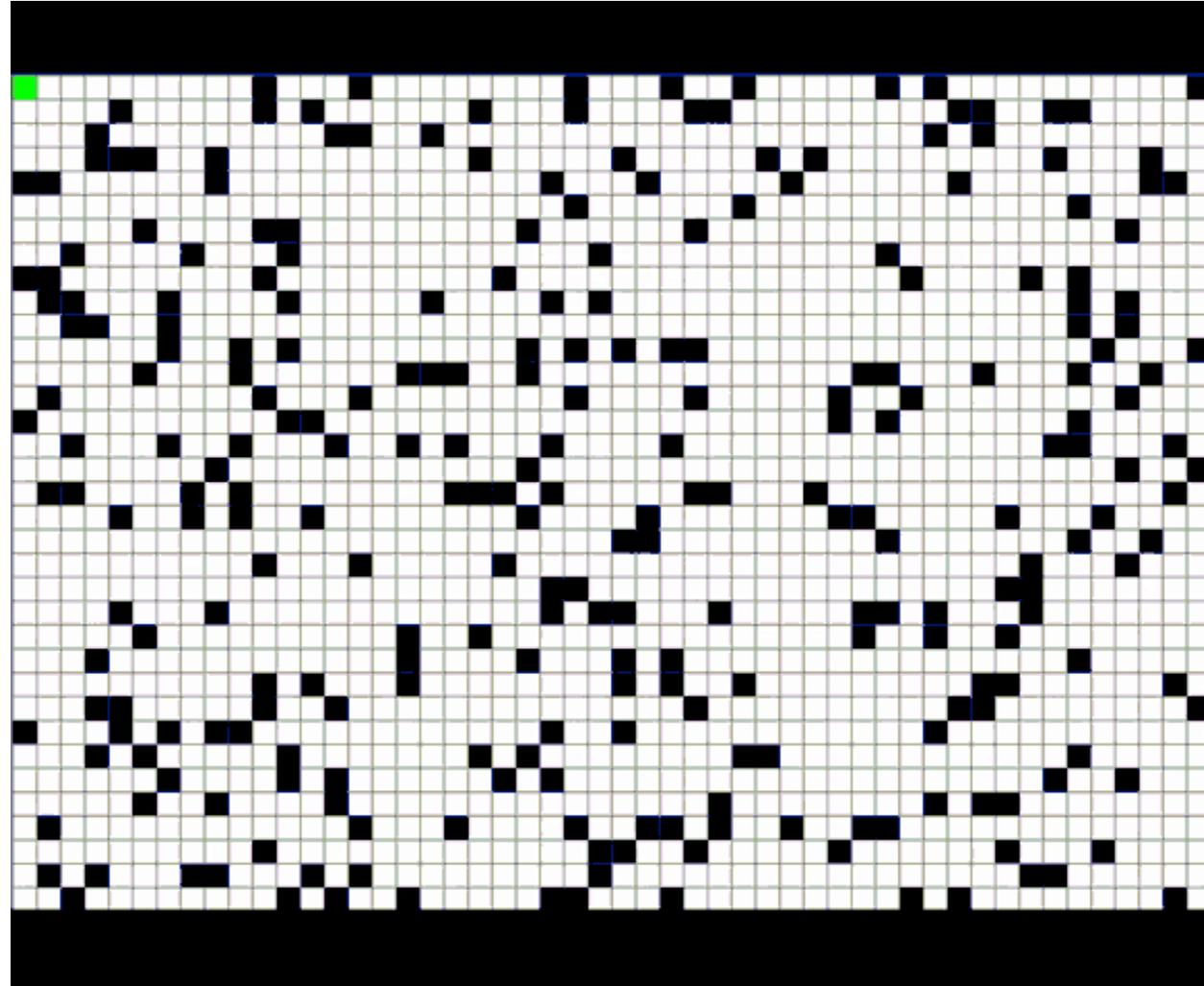
Ευρετική: προσεγγίζουμε με
Ευκλείδια απόσταση ή
απόσταση Manhattan



A^*

Επέλεξε τον επόμενο κόμβο βασισμένο στο άθροισμα της απόστασης από την αρχή και της ευρετικής

$$f(u) = d[u] + h(u) = d[u] + dist(u, t)$$



Λήψη Αποφάσεων

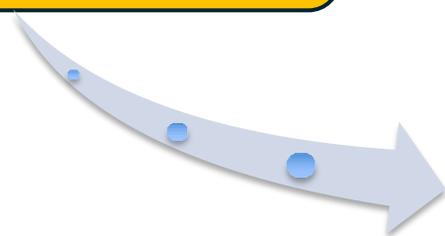
Σχεδιασμός/Διαδρομή

Κίνηση

Κίνηση

Steering behavior

Game Engine



Διεύθυνση (steering)

- Δύο βασικές στρατηγικές
 - Προσέγγιση (seek)
 - Μετακίνηση προς το στόχο
 - Απομάκρυνση (flee)
 - Μετακίνηση μακριά από τον στόχο
- Πολύπλοκη διεύθυνση
 - Συνδυασμός απλών μετακινήσεων



Κινηματική: Προσέγγιση

Κατεύθυνση:

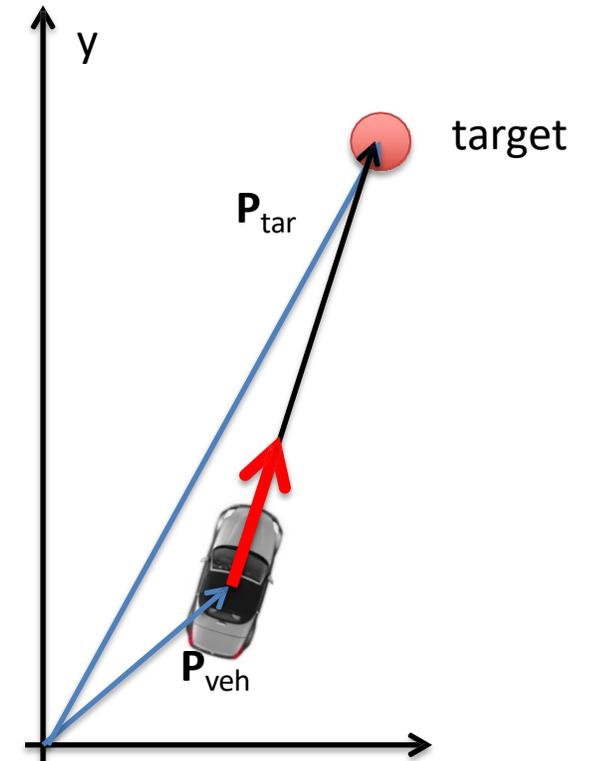
$$D = P_{tar} - P_{veh}$$

Ταχύτητα:

$$V = D.normalize \cdot maxSpeed$$

Θέση:

$$P_{veh} = P_{veh} + V \cdot deltaTime$$



Κινηματική: Απομάκρυνση

Κατεύθυνση:

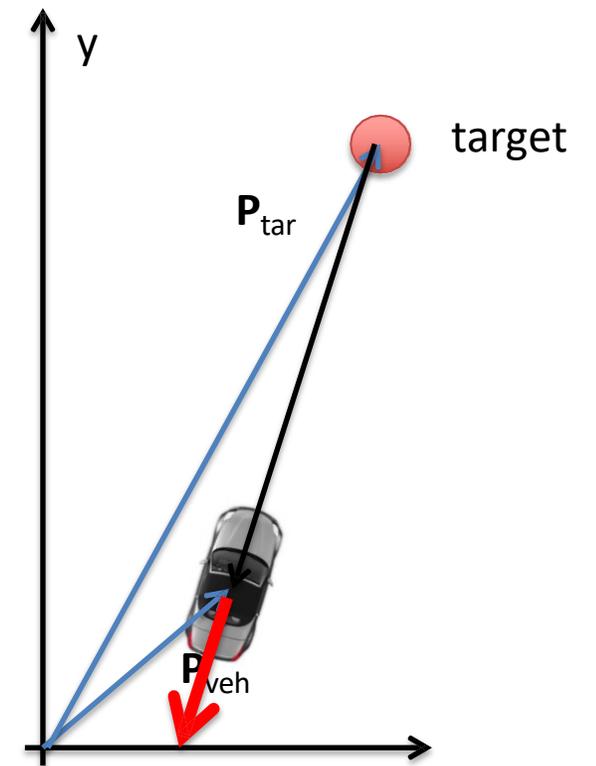
$$D = -(P_{tar} - P_{veh})$$

Ταχύτητα:

$$V = D.normalized \cdot maxSpeed$$

Θέση:

$$P_{veh} = P_{veh} + V \cdot deltaTime$$

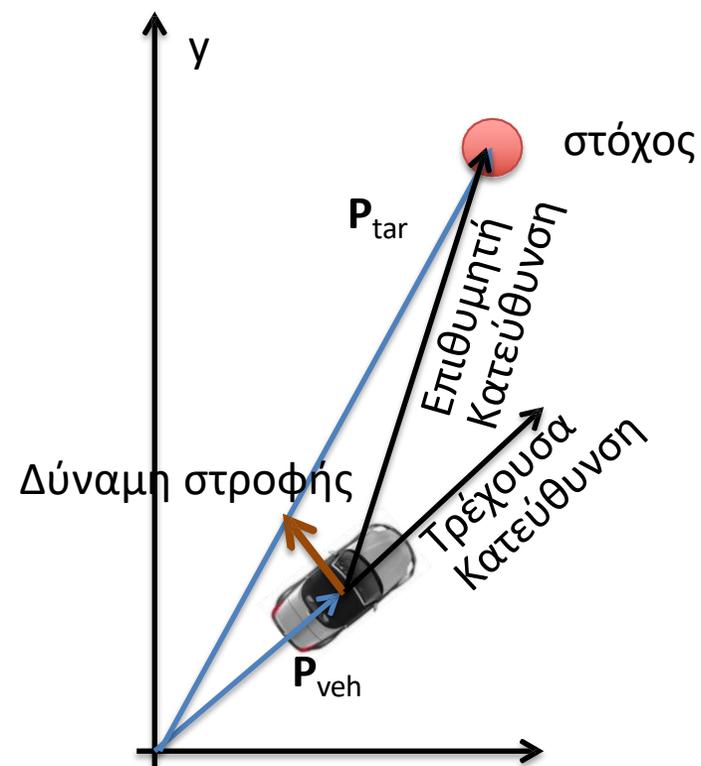


Προσέγγιση (Δυναμική)

- Επιθυμητή κατεύθυνση:

$$D = P_{tar} - P_{veh}$$

- Αν είναι διαφορετική από την τρέχουσα κατεύθυνση εφαρμόσε μία δύναμη στροφής προς τον στόχο

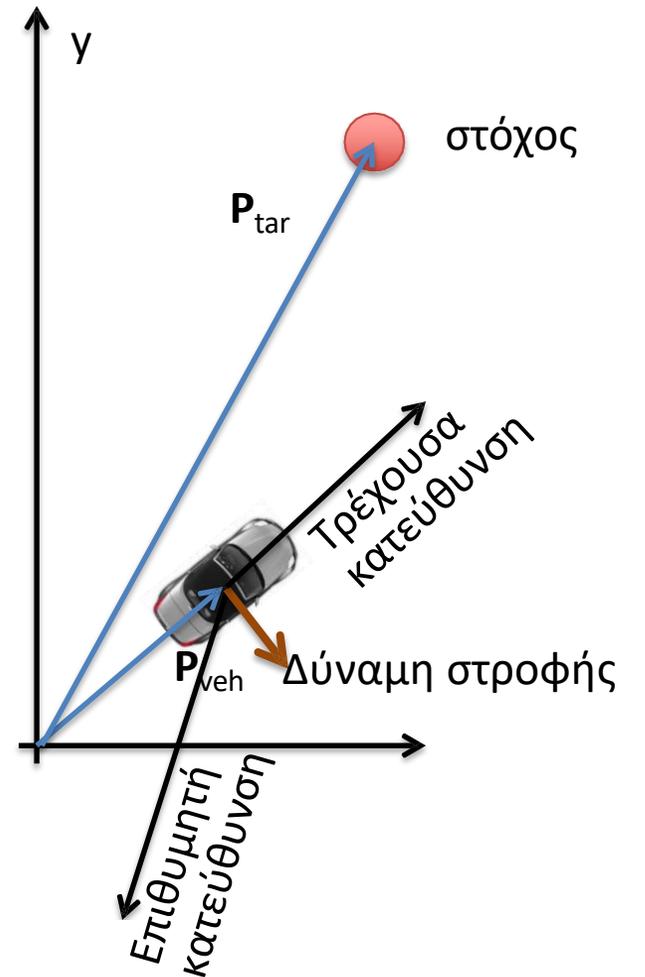


Απομάκρυνση (Δυναμική)

- Επιθυμητή κατεύθυνση:

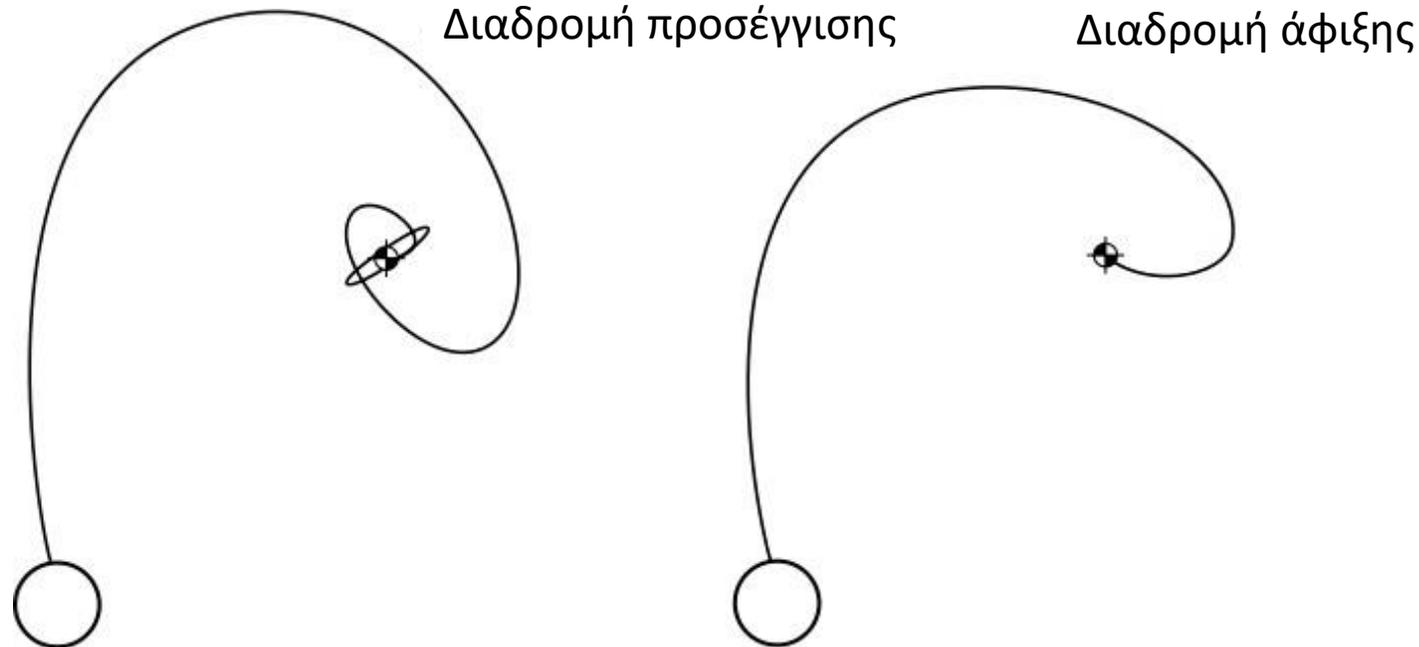
$$D = -(P_{tar} - P_{veh})$$

- Αν είναι διαφορετική από την τρέχουσα κατεύθυνση εφάρμοσε μία δύναμη στροφής μακριά από το στόχο



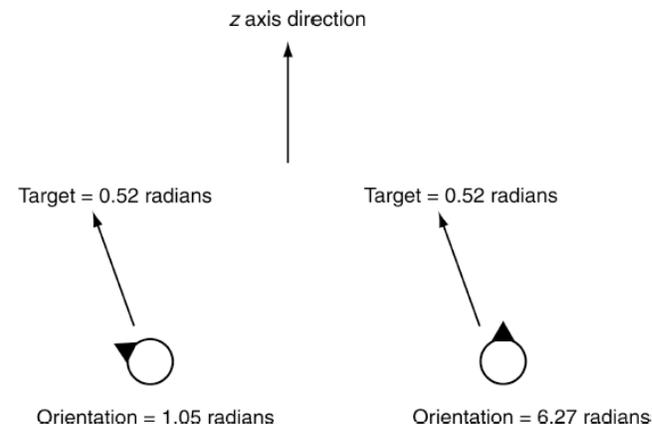
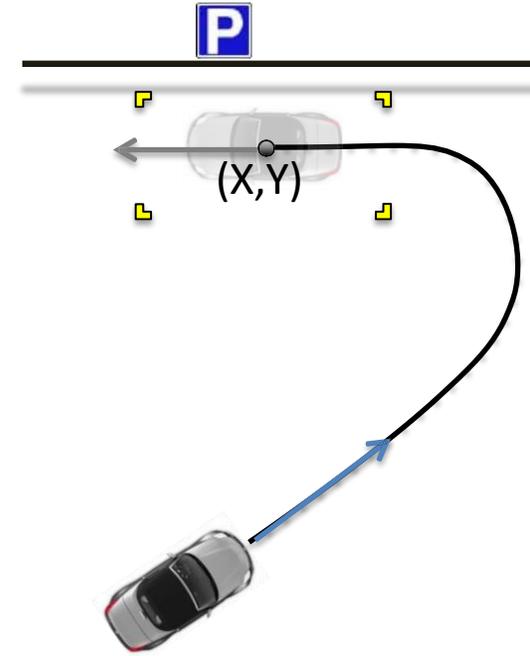
Άφιξη

- Αν η μετακίνηση γίνεται σε υψηλή ταχύτητα μπορεί να ξεφύγουμε
 - Στην κινηματική δεν υπάρχει τέτοιο πρόβλημα
- Όταν είσαι κοντά στον στόχο πάντα φρένο



Ταίριασμα και Ευθυγράμμιση

- Ο έλεγχος κίνησης μπορεί να απαιτείται να συνεργαστεί στενά με τη μηχανή φυσικής
 - Ταίριασμα
 - Ταίριασμα της ταχύτητας του πράκτορα με την ταχύτητα του στόχου (απλοποιημένη **άφιξη**)
 - Ευθυγράμμιση
 - Άφιξη με συγκεκριμένη κατεύθυνση



Πολύπλοκες Συμπεριφορές

- Καταδίωξη/Αποφυγή
- Περιπλάνηση
- Διαχωρισμός
- Παρακολούθηση διαδρομής

Ορίζονται σε σχέση με

- Προσέγγιση/Απομάκρυνση
 - Άφιξη, Ταίριασμα,
Ευθυγράμμιση

Καταδίωξη ή Αναχαίτιση

- Μετακινήσου εκεί που θα είναι ο στόχος
 - Υπόθεση: η ταχύτητα του στόχου δεν αλλάζει
 - Υπολόγισε το χρόνο για να μεταβείς στη τρέχουσα θέση του στόχου

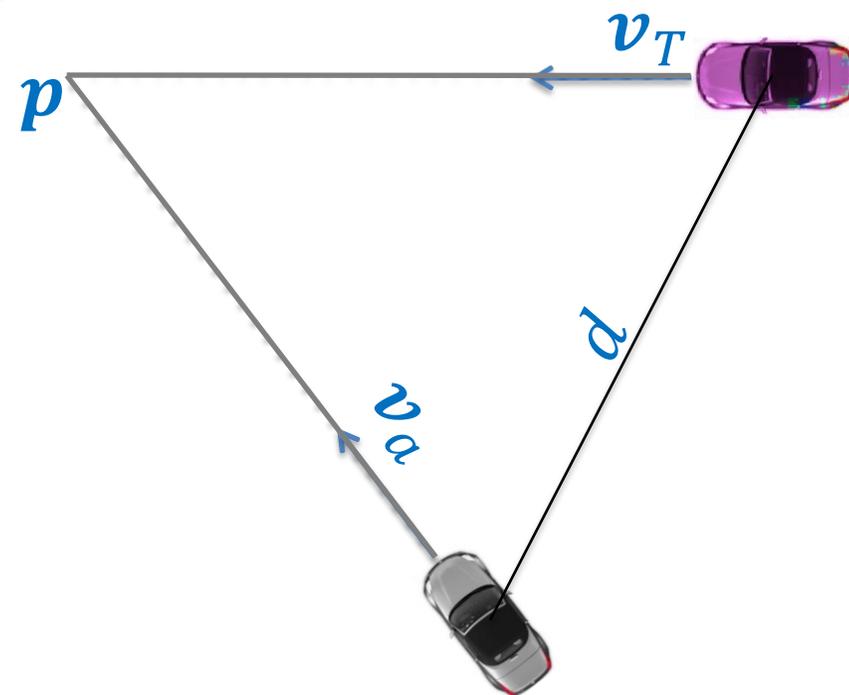
$$t = \frac{d}{v_a}$$

- Υπολόγισε τη θέση του στόχου αφού έχει περάσει χρόνος t

$$p = v_T t$$

- Οδήγησε προς τα εκεί: προσέγγιση στο p

Ανακριβές
αλλά απλό



Αποφυγή

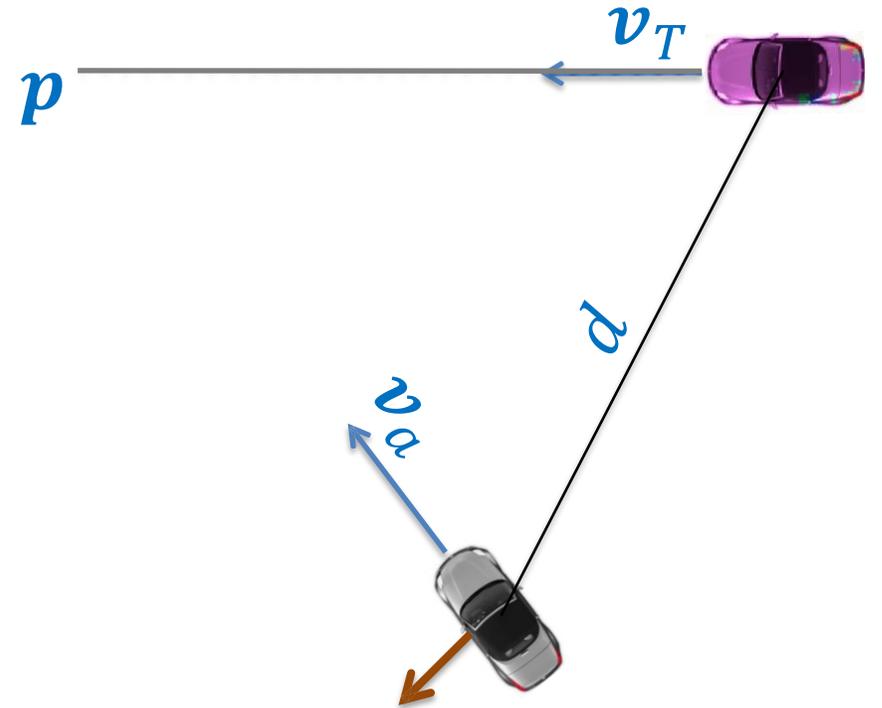
- Μετακινήσου μακριά από εκεί που θα είναι ο στόχος
 - Υπόθεση: η ταχύτητα του στόχου δεν αλλάζει
 - Υπολόγισε το χρόνο για να μεταβείς στη τρέχουσα θέση του στόχου

$$t = \frac{d}{v_a}$$

- Υπολόγισε τη θέση του στόχου αφού έχει περάσει χρόνος t

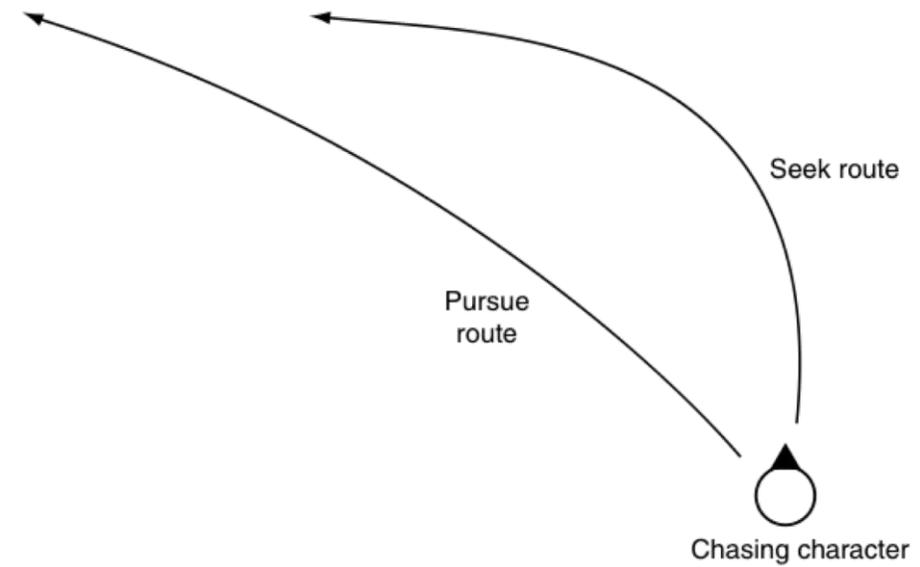
$$p = v_T t$$

- Οδήγησε μακριά από εκεί: **απομάκρυνση** από p



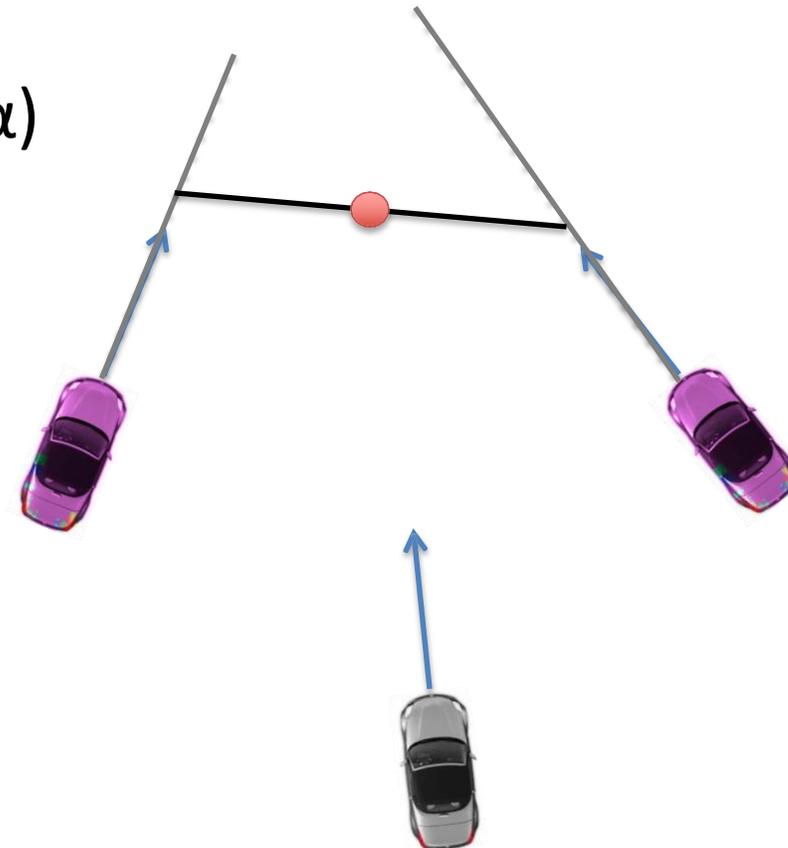
Καταδίωξη Στόχου που κάνει Αποφυγή

- Η ταχύτητα του στόχου δεν είναι σταθερή
 - Δεν μπορεί να γίνει πρόβλεψη
 - Επαναυπολογισμός της θέσης
 - Δεν υπάρχει λόγος για κάτι πιο «έξυπνο»



Παρεμβολή

- Στρίψε προς το μέσο σημείο της γραμμής που συνδέει δύο οντότητες.
 - Τερματοφύλακας (τέρμα – μπάλα)
- Παρόμοιο με καταδίωξη



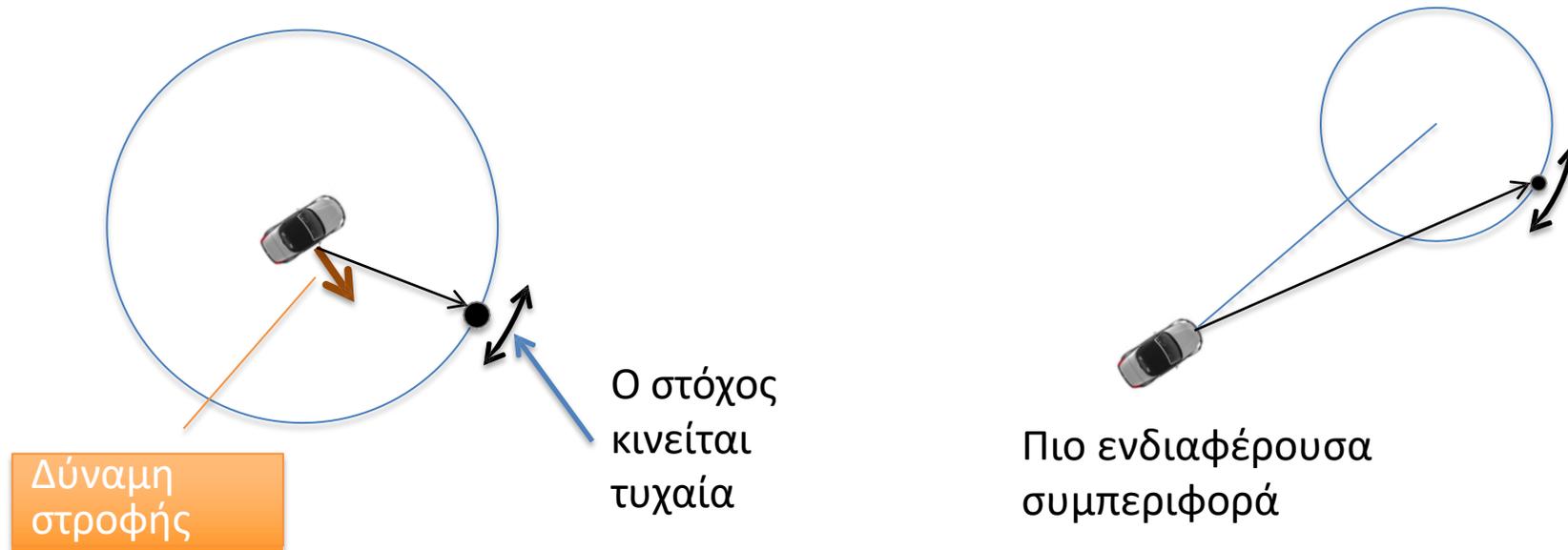
Περιπλάνηση

1. Τυχαίες δυνάμεις στροφής

- “ταλάντωση” πάνω σε μία ευθεία γραμμή



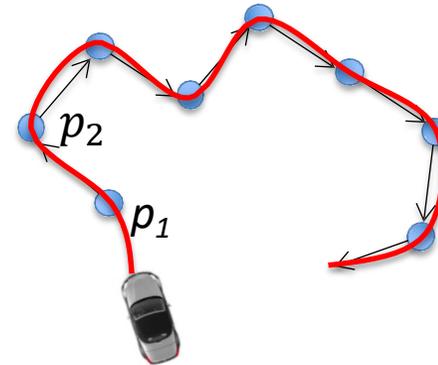
2. Προσέγγιση σε έναν τυχαία κινούμενο στόχο



Ακολουθώντας Διαδρομή

Διαδρομή: μία ακολουθία από ενδιάμεσα σημεία

- Ανοικτή ή κλειστή διαδρομή (κύκλωμα)
- Εντοπισμός του πλησιέστερου σημείου p_1
- Προσέγγιση(p_1)
- Όταν φτάσει κοντά στο p_1
- Προσέγγιση(p_2)
- ...



Ακολουθώντας μία
πίστα αγώνων

Συνδυασμός Συμπεριφορών Κίνησης

- Περιπολικό:
 - Καταδίωξη
 - Αποφυγή εμποδίων
- Ζώο
 - Περιπλάνηση
 - Αποφυγή εμποδίων
 - Αποφυγή θηρευτών

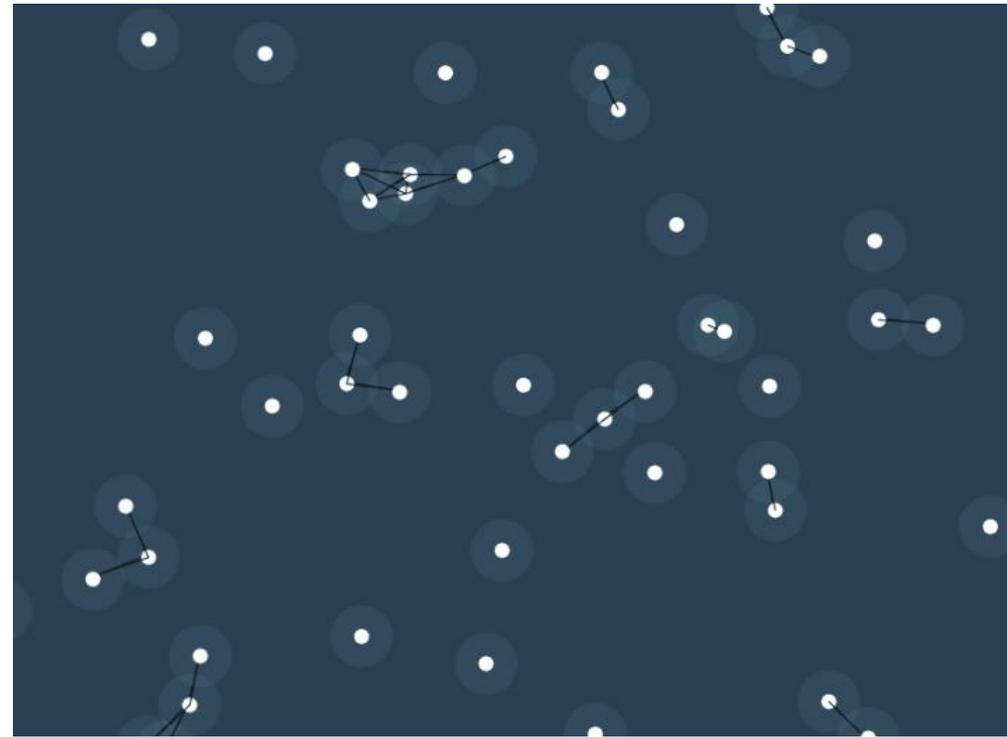
Τεχνικές

- Ανάμειξη
 - Συλλογή όλων των δυνάμεων στροφής
$$F = w_1 F_1 + w_2 F_2 + \dots$$
- Προτεραιότητες
 - Ταξινόμηση των μεθόδων κίνησης κατά προτεραιότητα
 - Εκτέλεσε αυτή με την υψηλότερη προτεραιότητα μέχρι να τελειώσει ή να βρεθεί άλλη με υψηλότερη προτεραιότητα
- Άλλα κόλπα; (Hacks)

Παράδειγμα Ανάμειξης: Flocking

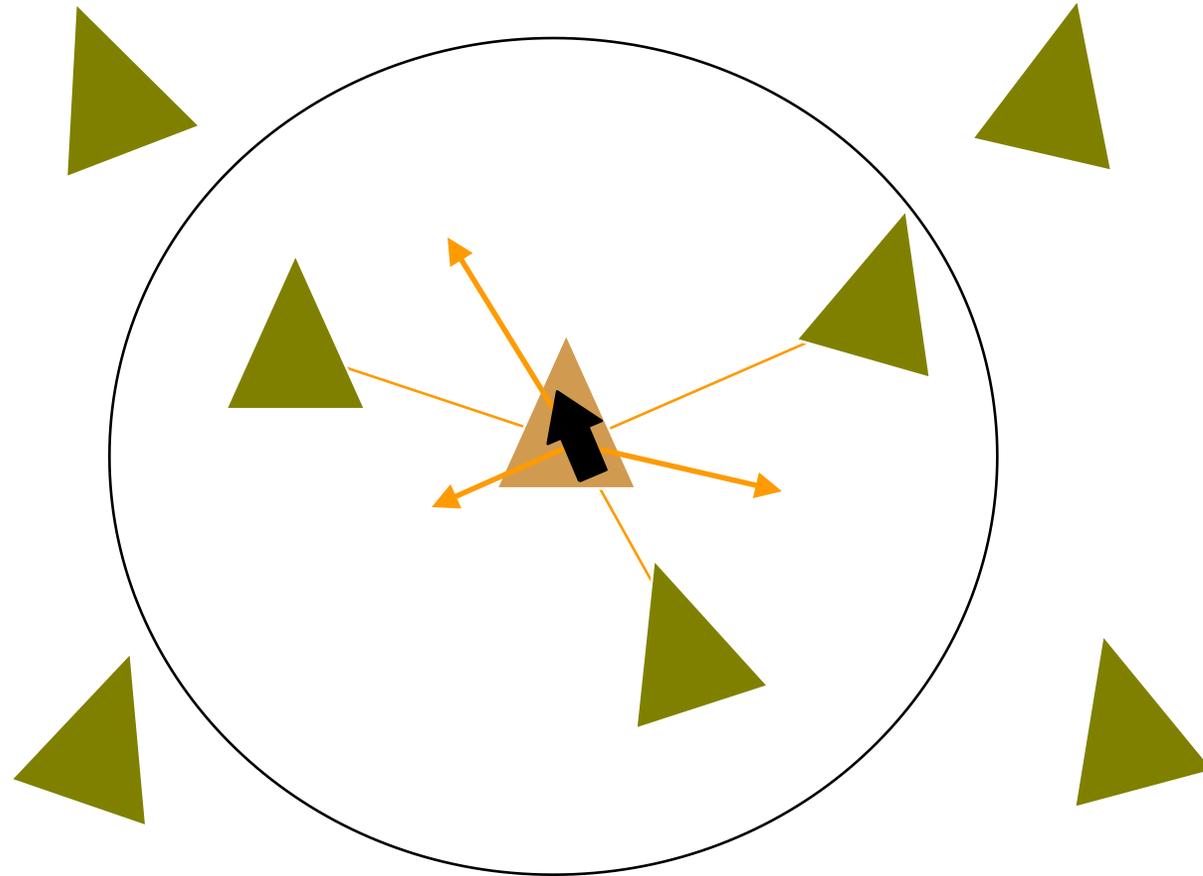


- Συνδυασμός από:
 - Διαχωρισμός
 - Ευθυγράμμιση
 - Συσώρευση
- Παράγει ρεαλιστικά αποτελέσματα



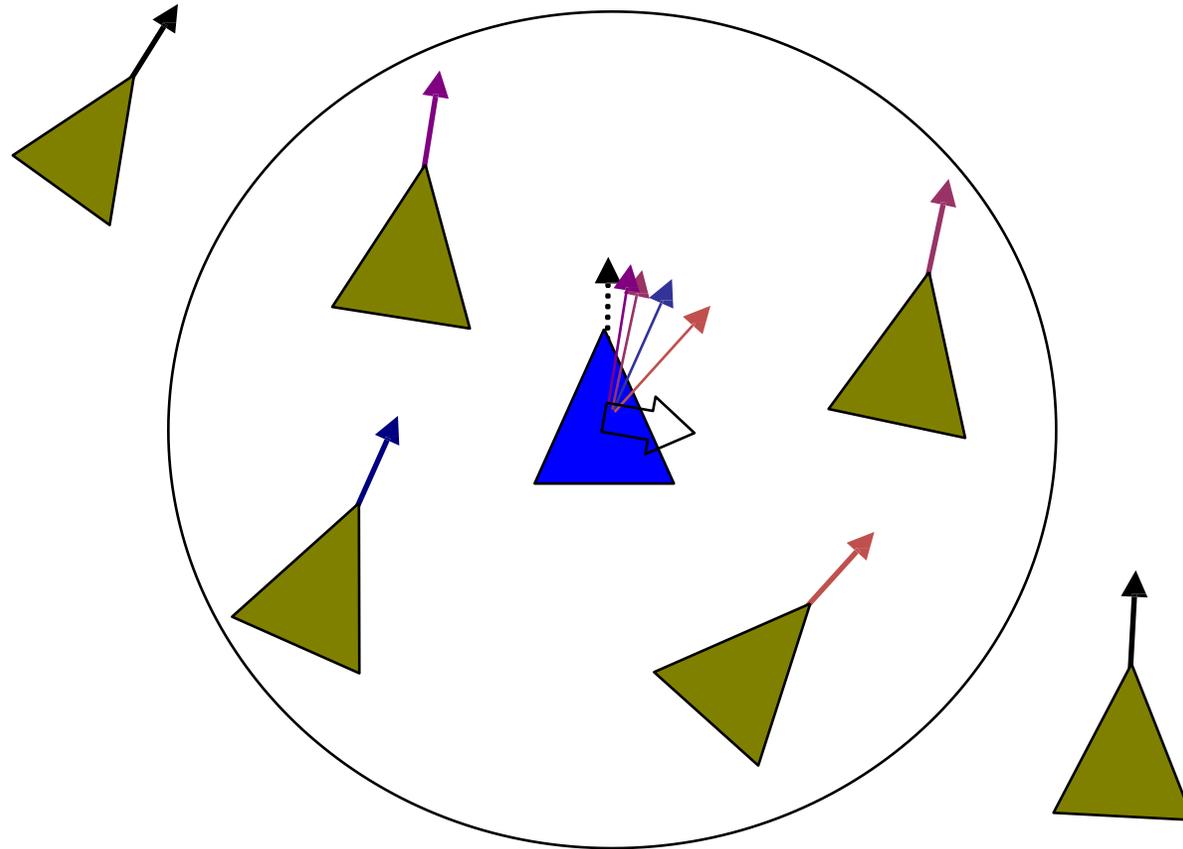
Διαχωρισμός: Αποφυγή Void

- Απομακρύνσου από τα boids που είναι κοντά σου



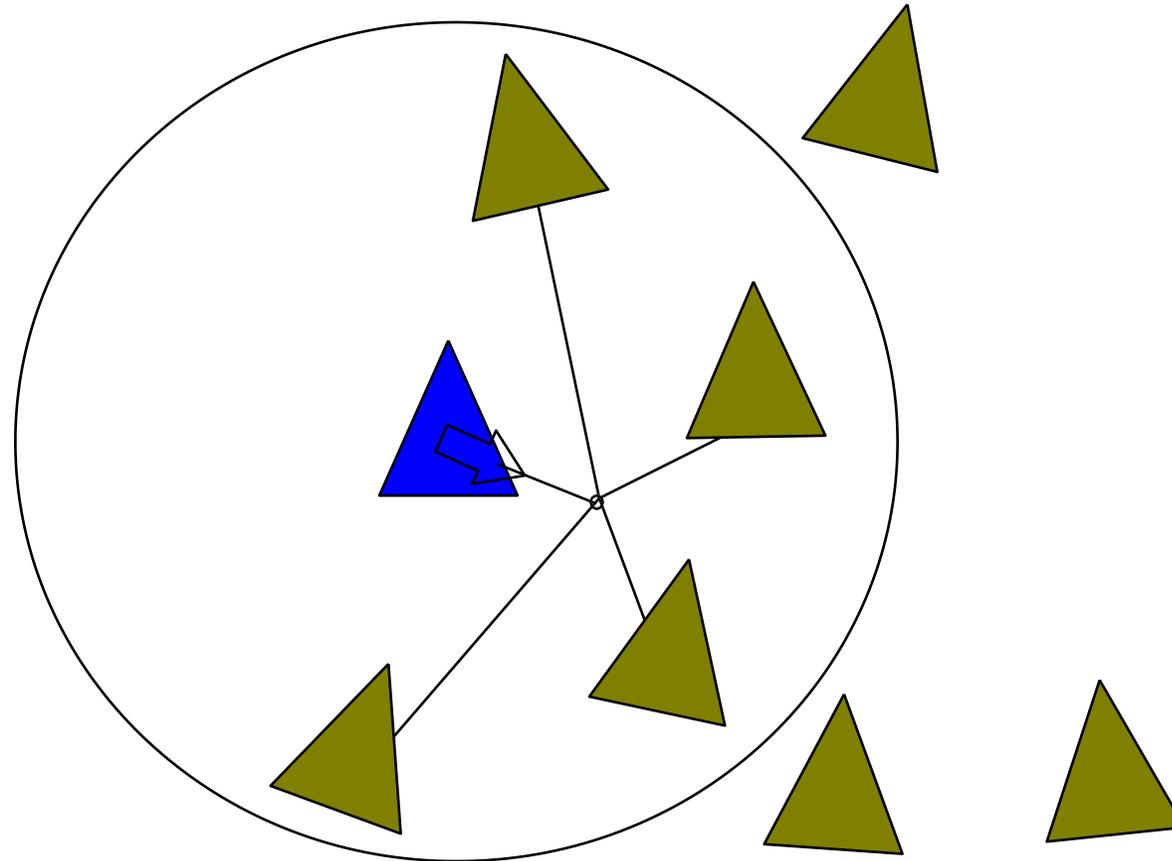
Ευθυγράμμιση

- Μετακίνηση προς την ίδια κατεύθυνση και ταχύτητα με το σμήνος



Συσσώρευση

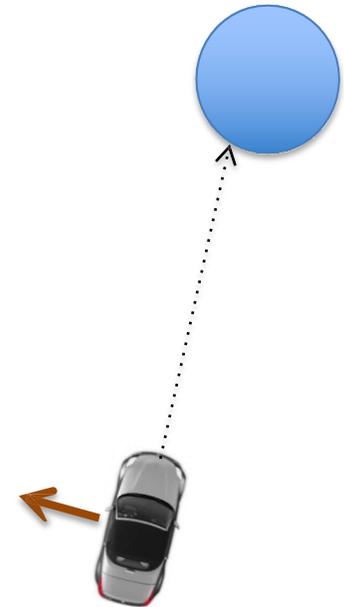
- Μετακίνηση προς το κέντρο μάζας του σμήνους



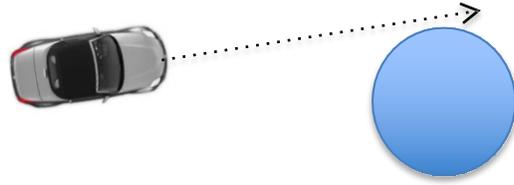
Ακόμα πιο Πολύπλοκες Συμπεριφορές Στροφής

Αποφυγή Σύγκρουσης

- *Στρίψε γύρω από τα εμπόδια*
 - Ρίψη ακτίνας προς τη κατεύθυνση της κίνησης
 - Αν σύγκρουση τότε
 - Εφαρμογή δύναμης στροφής
 - *Απομάκρυνση* μέχρι αποφυγή σύγκρουσης
 - Αποφυγή κοντινότερου εμποδίου
 - Δεν δουλεύει σε πολύπλοκα περιβάλλοντα



Ρίψη Ακτίνας



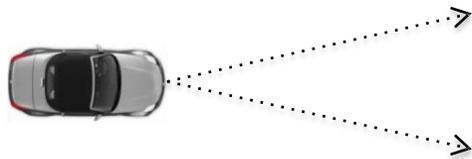
Μία απλή ακτίνα μπορεί να μην εντοπίσει το εμπόδιο

Παραλλαγές

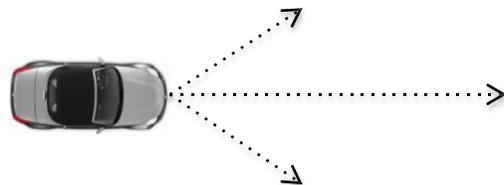
– Parallel side rays



– Whiskers

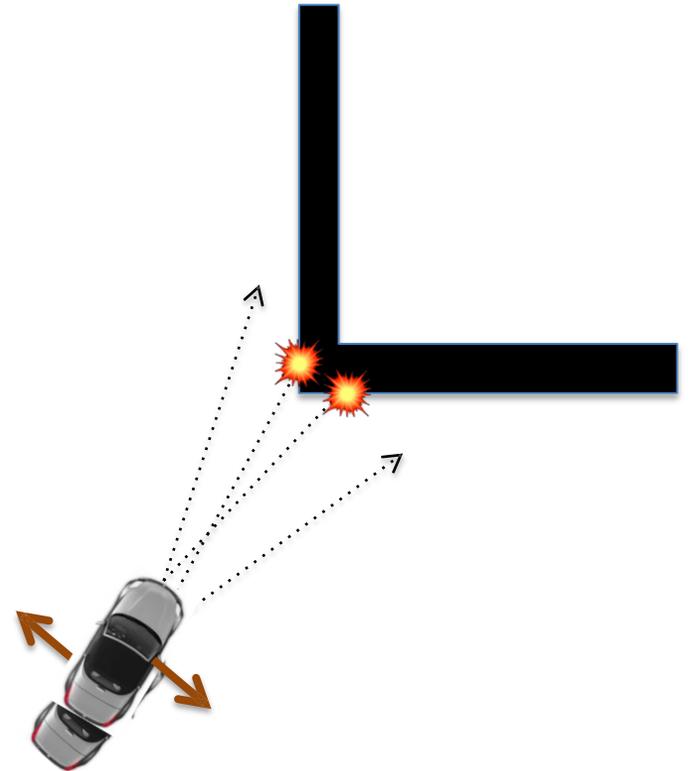


– Central ray + whiskers

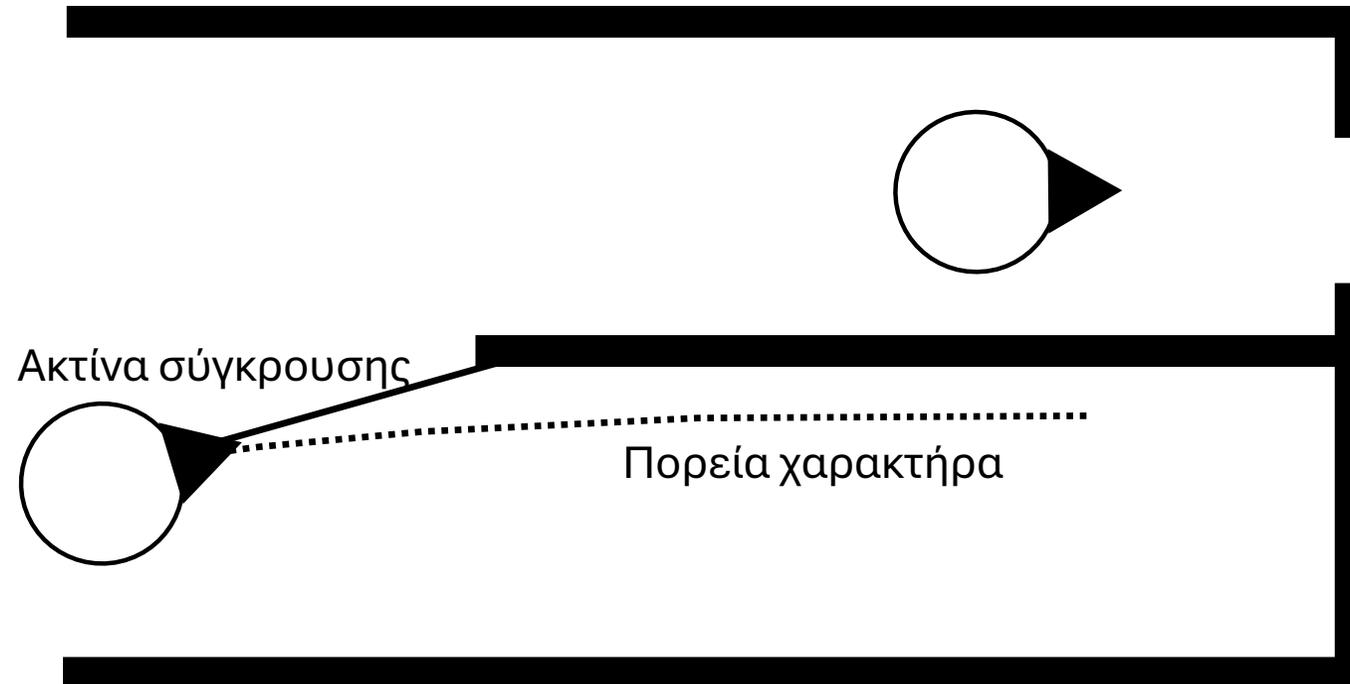


Πρόβλημα: Παγίδα γωνίας

- Μπορεί να γίνει με οποιονδήποτε αριθμό ακτινών
 - Ιδιαίτερη μεταχείριση γωνιών



Αποτυχία Κίνησης



Μελέτη
Περίπτωσης:
1) Engage
2) Chase

10/10
BEST GAME OF 2016*
PCGAMESN

10/10
ANTASTIC*
IGN OF THE FANBOY

9.3/10
AMAZING*
IGN

4.5/5
SUPERB*
GAMESRADAR

10/10
CREDIBLE LEVEL
WORLD DESIGN*
GOGAMER.COM

9/10
BENT AND DARING*
METRO

4.5/5
MASTERPIECE*
GAMESREVIEWS

A promotional image for the video game Dishonored 2. The image is split vertically by a glowing blue line. On the left side, there is a close-up of a character's face wearing a dark, textured, and somewhat damaged mask. On the right side, there is a close-up of a woman's face, likely Emily Kaldwin, wearing a dark blue and gold patterned veil. The background is dark and atmospheric. At the bottom, the title "DISHONORED 2" is written in a large, stylized, metallic font.

DISHONORED 2

Το Πρόβλημα της Καταδίωξης

- *Engage*: NPCs με άμεση οπτική επαφή με τον παίκτη επιτίθενται
- *Chase*: NPCs που έχασαν τον στόχο τους τρέχουν πίσω τους να τους βρουν.
- Ένας παίκτης που καταδιώκει:
 - Τρέχει προς ένα σημείο που πιστεύει ότι είναι ο παίκτης
 - Αν τον βρει, επιτίθεται.
 - Αν δεν τον βρει, μπαίνει σε κατάσταση αργής αναζήτησης τριγύρω στο χώρο
- Πρέπει να ληφθούν υπόψη:
 - Κάθετες περιοχές
 - Μαγικές δυνάμεις (π.χ., Far Reach)

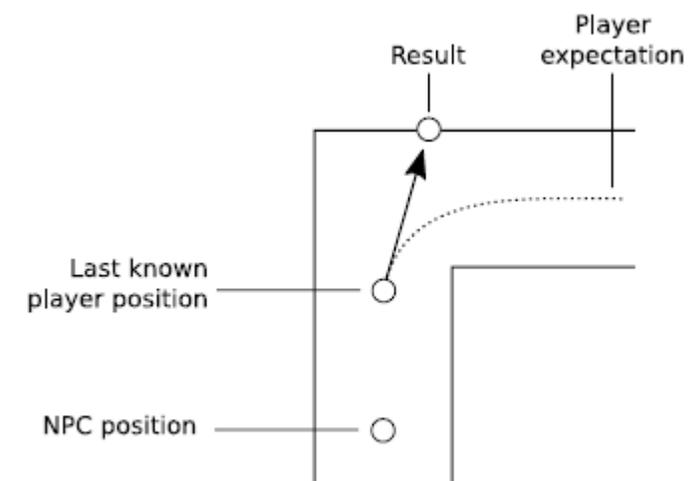
Τεχνικές

Dishonored 1:

- Οι καταδιωκόμενοι αφήνουν στοιχεία πίσω τους (ψίχουλα – breadcrumbs). Οι NPCs τα βρίσκουν και ακολουθούν
 - Σχετικά καλά αποτελέσματα αλλά άδικο για τους παίκτες

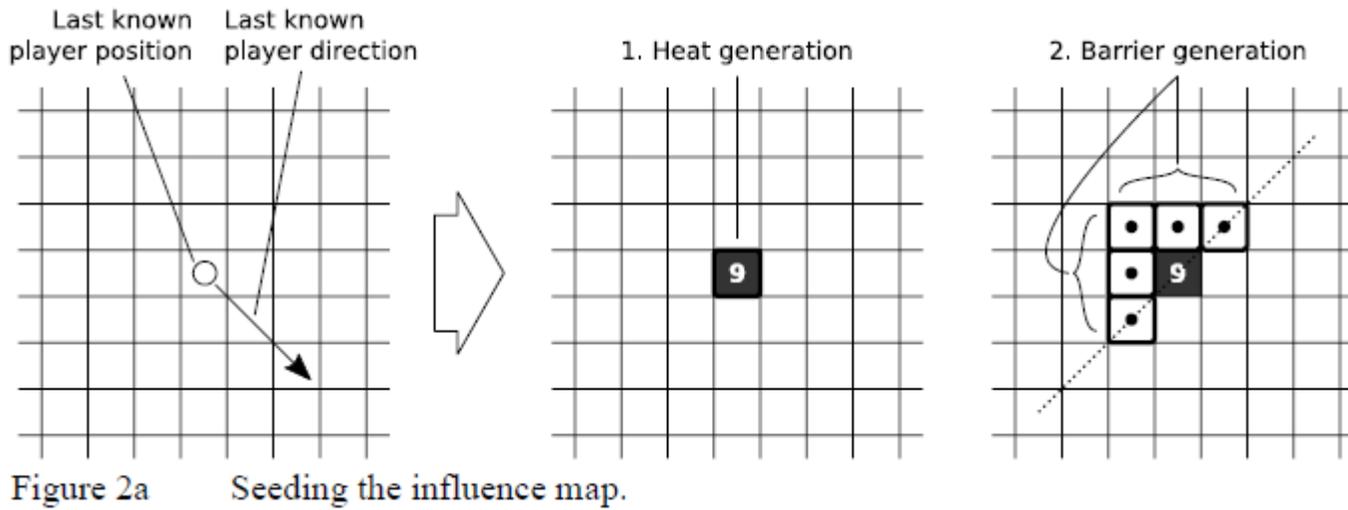
- Αρχικά για το **Dishonored 2**: τεχνική **dead reckoning** (πλοήγηση εξ αναμετρήσεως)

- Γραμμική προέκταση (extrapolation) με χρήση της τελευταίας γνωστής θέσης και κατεύθυνσης του στόχου
- Αν και δούλεψε καλά στην αρχή οι play-testers ανέφεραν ότι η καταδίωξη σταματούσε γρήγορα και ήταν απαγοητευτική



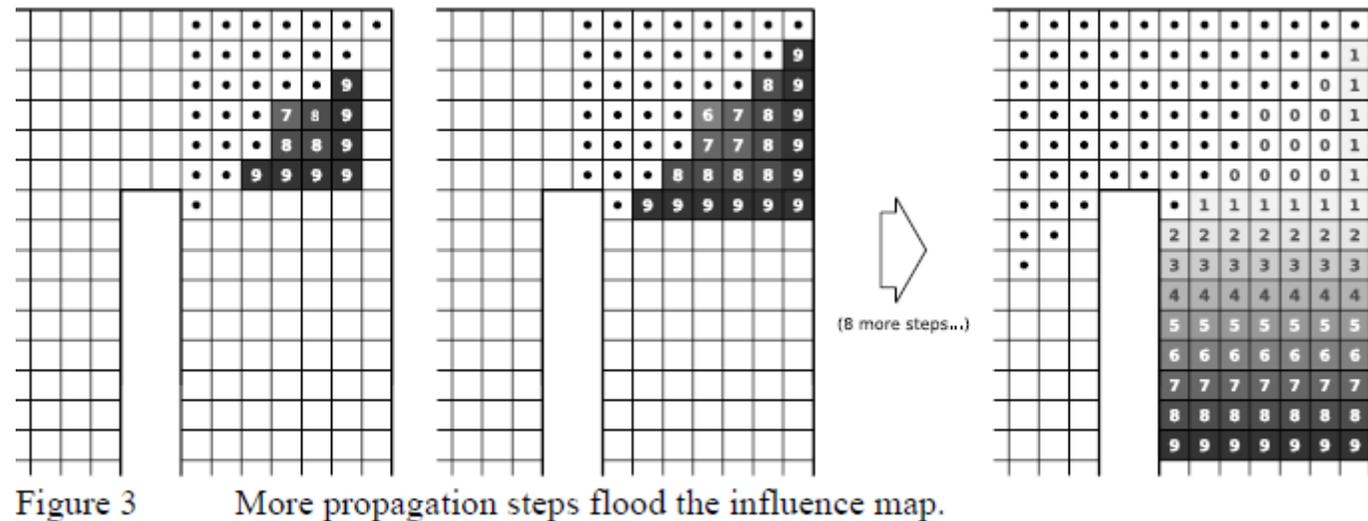
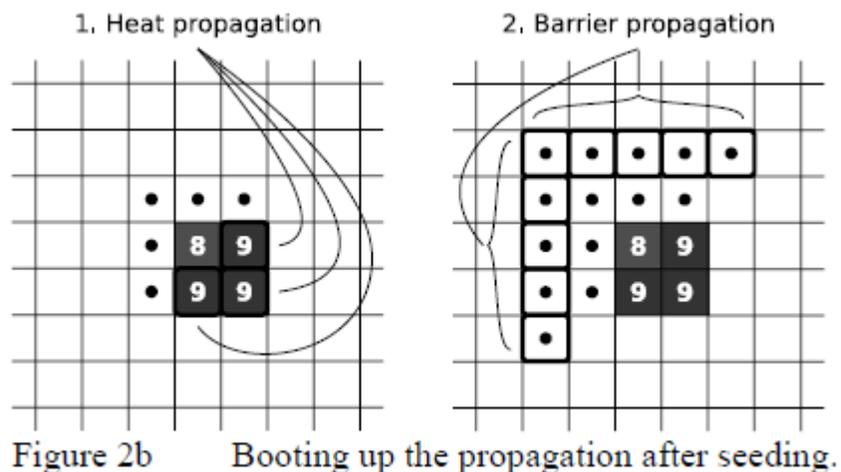
Χάρτες Επιρροής – Αρχικοποίηση

- Θα το δούμε μόνο για 2d – 3d είναι ανάλογο
- Παραγωγή θερμότητας
- Παραγωγή εμποδίων



Χάρτες Επιρροής – Διάδοση

- Διάδοση θερμότητας
- Διάδοση εμποδίων

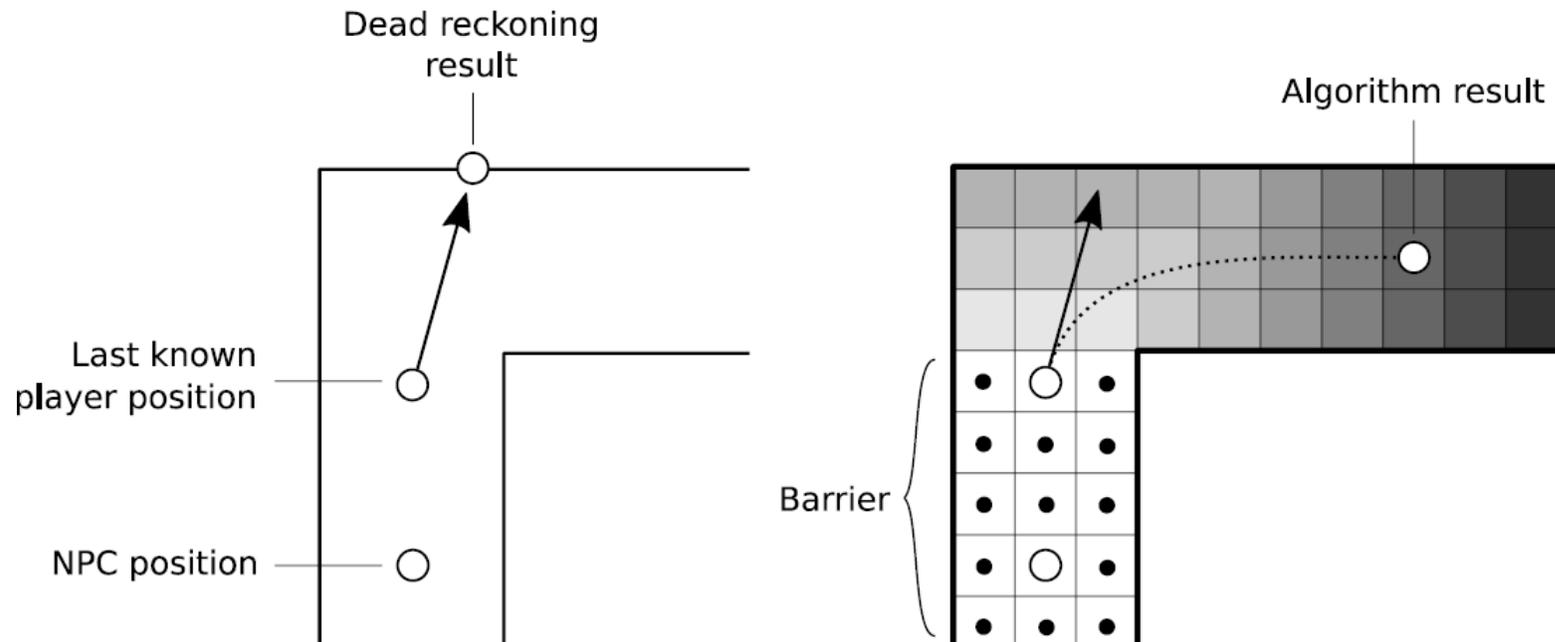


Συνθήκες τερματισμού:

- Δεν υπάρχουν άλλα κελιά για διάδοση θερμότητας
- Πολλά κελιά θερμαίνονται στο ίδιο βήμα ($\geq M_H$)
- Έχουν γίνει πολλές επαναλήψεις (M_S)

Σύγκριση με Dead Reckoning

Ο προορισμός καταδίωξης προσδιορίζεται με τον υπολογισμό της μέσης θέσης των θερμών κελιών στον χάρτη (το κέντρο τους), σταθμισμένα με τις τιμές τους.



Κάποιες Ιδιότητες

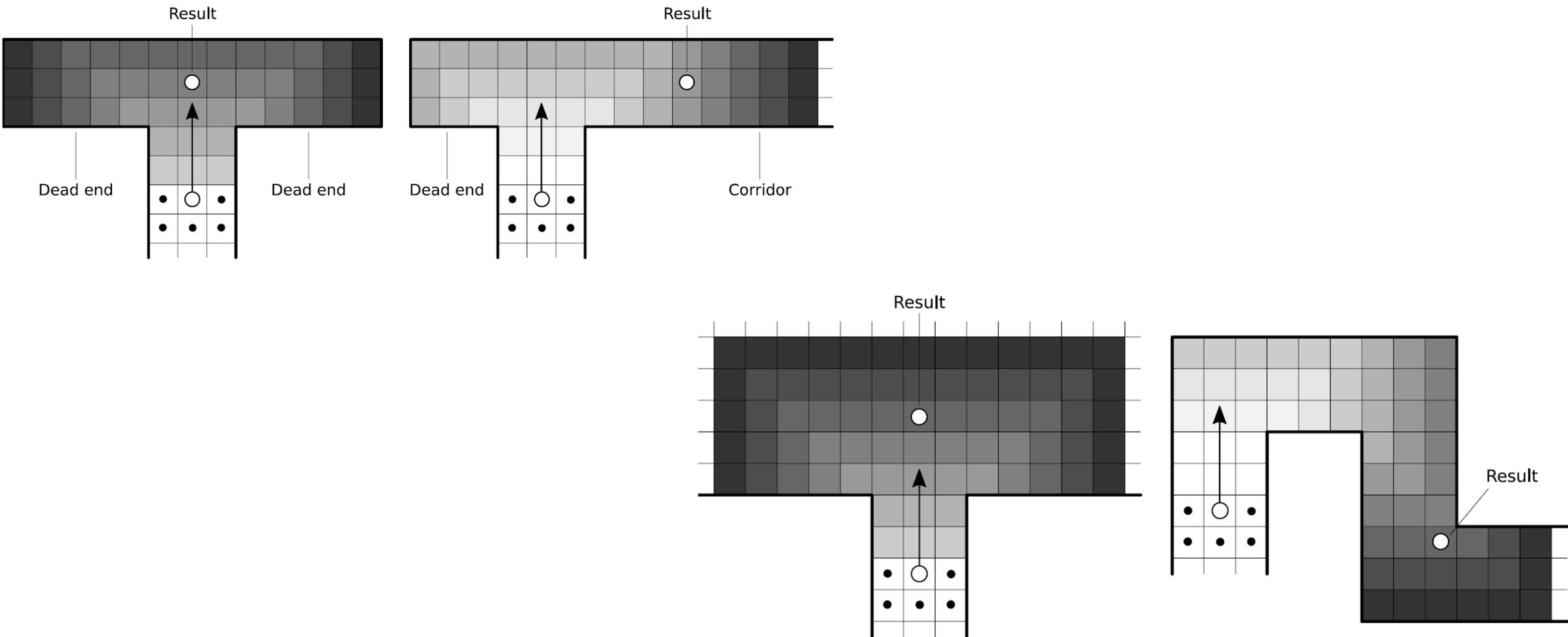


Figure 6 The chase stops early in wide areas but keeps going in long corridors.

Ρύθμιση Δυσκολίας

Table 1 Tweakable constants

| Constant | Effect | Value in <i>Dishonored 2</i> |
|--|---|---|
| Hot value (H) | Sets how prone NPCs are to explore long corridors. The lower the value, the faster the cells in a dead end cool down to zero, so the more the result is skewed towards corridors. | 20 |
| Maximum heated cells per step (M_H) | Drives how easily the propagation stops in wide spaces and therefore how aggressively the NPCs will venture into wider spaces. | 25 |
| Maximum steps (M_s) | Drives how far the propagation can run. This value is the one that strictly limits how far NPCs will run when they chase a player, so it was used as a difficulty level variable in <i>Dishonored 2</i> . | Level 1: 20 Level 2: 30 Level 3: 40 |

Θέματα

- Δεν προσαρμόζεται σε νέα πληροφορία (πυροβολισμός από άλλη κατεύθυνση)
 - Στο Dishonored 2 απλά όταν υπάρχει τέτοιο γεγονός, ο αλγόριθμος ξανατρέχει από την αρχή
- Μπορεί να χαθούν περιοχές για καταδίωξη
 - Δεν λύθηκε μιας και το κόστος ήταν αρκετά μεγάλο
 - Εμφανίζεται σπάνια αν όχι καθόλου

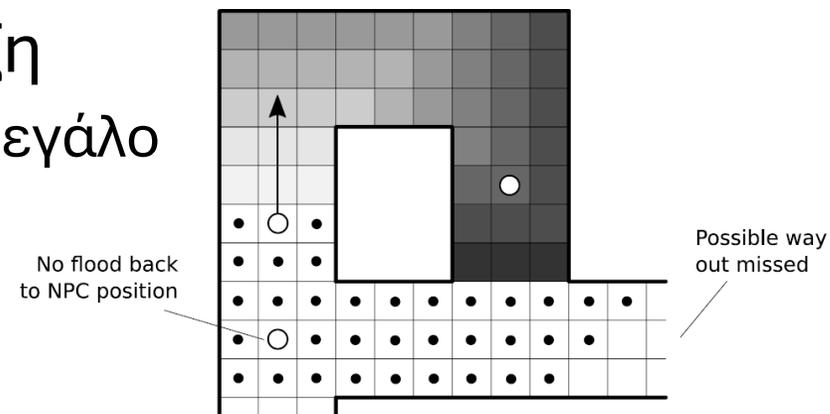


Figure 7 The barrier sometimes prevents NPCs from exploring an area.