

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**  
**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ**  
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**



**Μάθημα Προπτυχιακών Σπουδών:**  
**Ευφυείς Πράκτορες**  
**Ακαδημαϊκό έτος: 2023 - 2024**  
**Εξάμηνο: 8ο**  
**Εγχειρίδιο Παρουσίασης Απαλλακτικής Εργασίας**

<b><u>Ομάδα Εργασίας:</u></b> Θεόδωρος Κοξάνογλου Π20094 Απόστολος Σιαμπάνης Π20173 Δημήτρης Στυλιανού Π20004 Αντώνιος Ρούσσος Π20167	<b><u>Υπεύθυνος Καθηγητής:</u></b> Θέμης Παναγιωτόπουλος
---	---

## Περιεχόμενα

### Περιεχόμενα

Περιγραφή προβλήματος .....	3
<b>Ανάλυση</b> .....	3
<b>Οι Πράκτορες</b> .....	3
<b>Κίνηση και Αντίληψη</b> .....	4
<b>Προσομοίωση και Στόχοι</b> .....	4
<b>Αναζήτηση και Συλλογή</b> .....	4
<b>Ανταλλαγή Γνώσεων και Γονιμοποίηση</b> .....	5
<b>Τερματισμός προσομοίωσης</b> .....	5
 Υλοποίηση.....	6
 Περιγραφή θεωρητικής βάσης.....	7
 Περιγραφή σημαντικών σχεδιαστικών αποφάσεων και στοιχείων υλοποίησης.....	13
 Παράδειγμα εκτέλεσης .....	14
<b>Αρχική Οθόνη παιχνιδιού</b> .....	14
<b>Επεξεργασία μεταβλητών</b> .....	15
<b>Οθόνη Προσομοίωσης</b> .....	16
<b>Εμφάνιση του χάρτη που έχει ανακαλύψει κάποιος agent</b> .....	17
<b>Παρακολούθηση ενός συγκεκριμένου πράκτορα</b> .....	17
<b>Τερματισμός παιχνιδιού</b> .....	18
 Πηγές .....	19

# Περιγραφή προβλήματος

Δύο αντίπαλες ομάδες ευφυών εικονικών πρακτόρων με διαφορετικές ικανότητες προσπαθούν να συγκεντρώσουν συγκεκριμένα αγαθά από το κοινό περιβάλλον τους. Η ομάδα που θα τα συγκεντρώσει πρώτη νικάει το παιχνίδι.

## Ανάλυση

Ο εικονικός κόσμος είναι επίπεδος και αποτελείται από  $N \times M$  θέσεις (από  $25 \times 25$  έως  $100 \times 100$ ). Οι διαστάσεις  $N$  και  $M$  ορίζονται ως παράμετροι στην αρχή της εφαρμογής. Ο χώρος αυτός είναι μια πεδιάδα όπου υπάρχουν διάφοροι πόροι (ξύλεια, πέτρα, χρυσός) καθώς και τα χωριά στα οποία κατοικούν οι δύο ομάδες.

Κάθε θέση μπορεί να περιέχει:

1. ένα μέρος συγκεκριμένου πόρου
2. ένα μέρος συγκεκριμένου χωριού
3. γρασίδι (κενός χώρος)
4. εμπόδιο

## Οι Πράκτορες

Οι πράκτορες έχουν την δυνατότητα να κινηθούν ελεύθερα στην πεδιάδα, ανακαλύπτοντας και αποφεύγοντας τα εμπόδια. Κάθε ομάδα πρακτόρων προσπαθεί να συγκεντρώσει στο χωριό της συγκεκριμένο αριθμό πόρων. Οι πόροι που πρέπει να συγκεντρωθούν καθορίζονται από τον χρήστη κατά την εκκίνηση της εφαρμογής. Κάθε πράκτορας μπορεί να μεταφέρει μερικές μονάδες από κάθε πόρο, ανάλογα με τα χαρακτηριστικά του. Η κάθε ομάδα αποτελείται από  $K$  πράκτορες - το  $K$  ορίζεται ως παράμετρος στην αρχή της εφαρμογής.

## Χαρακτηριστικά Πρακτόρων

Κάθε πράκτορας μιας ομάδας διαθέτει τα εξής χαρακτηριστικά:

1. **Χωριό στο οποίο ανήκει ο πράκτορας (1 bit):** 0 (village 1) ή 1 (village 2)
2. **Ενέργεια που καταναλώνει ο πράκτορας ανά clock tick (1 bit):** 0 (-1 energy) ή 1 (-2 energy)
3. **Ταχύτητα του πράκτορα (1 bit):** 0 (250 speed) ή 1 (300 speed)
4. **Χωρητικότητα κουβαλήματος ξυλείας (2 bits):** 00 (10 wood) ή 01 (20 wood) ή 10 (30 wood) ή 11 (40 wood)

5. **Χωρητικότητα κουβαλήματος πέτρας (2 bits):** 00 (5 stone) ή 01 (10 stone) ή 10 (15 stone) ή 11 (20 stone)
6. **Χωρητικότητα κουβαλήματος χρυσού (1 bits):** 0 (1 gold) ή 1 (3 gold)

## Κίνηση και Αντίληψη

Οι πράκτορες κινούνται σε σχήμα σταυρού, δηλαδή κατά τον οριζόντιο και κάθετο άξονα. Κάθε φορά που φτάνουν σε ένα πλακίδιο (tile), αποφασίζουν για την επόμενη κίνησή τους. Επιπλέον, μπορούν να αντιληφθούν το περιβάλλον τους, συγκεκριμένα αν οι γειτονικές τους θέσεις είναι κενές, περιέχουν τμήμα συγκεκριμένου πόρου ή κάποιο εμπόδιο.

## Προσομοίωση και Στόχοι

Κατόπιν εισαγωγής των αρχικών δεδομένων από τον χρήστη, πραγματοποιείται μια τυχαία δημιουργία του περιβάλλοντος. Στη συνέχεια, οι πράκτορες τοποθετούνται στα χωριά τους και η προσομοίωση ξεκινά. Οι πράκτορες ξεκινούν με 100 μονάδες ενέργειας, οι οποίες μειώνονται ανά δευτερόλεπτο (κατά μία ή δύο μονάδες ανάλογα το bit στα χαρακτηριστικά του). Σε κάθε πράκτορα ανατίθεται κάποιος στόχος κάποιου πόρου που πρέπει να βρει. Η επιλογή του στόχου γίνεται από τον GameManager, ο οποίος χρησιμοποιεί τα χαρακτηριστικά του εκάστοτε πράκτορα για να υπολογίσει ποιος πόρος είναι καλύτερο να μαζέψει εκείνη την χρονική στιγμή βάσει της κατάστασης της ομάδας και των χαρακτηριστικών του.

## Αναζήτηση και Συλλογή

Αφού ο πράκτορας λάβει κάποιο στόχο, ξεκινά την “τυφλή” αναζήτηση στον χάρτη. Κατά την ανακάλυψη του χάρτη, αποθηκεύει τα χρήσιμα πλακίδια σε μια βάση γνώσης, ώστε να γνωρίζει την θέση τους αν χρειαστεί να τα επισκεφτεί στο μέλλον. Μόλις βρει τον στόχο του, τον συλλέγει και τον επιστρέφει στο χωριό του, όπου του ανατίθεται νέος στόχος βάσει των νέων δεδομένων. Η αναζήτηση μπορεί να πραγματοποιηθεί είτε με “τυφλή” αναζήτηση είτε με τη μέθοδο A\* (Astar). Ο A\* είναι χρήσιμος στην περίπτωση που ο πράκτορας γνωρίζει τη θέση ενός πλακιδίου και τον τρόπο που μπορεί να μετακινηθεί προς αυτό.

## **Ανταλλαγή Γνώσεων και Γονιμοποίηση**

Κάθε φορά που συναντιούνται δύο πράκτορες μπορούν να ανταλλάξουν γνώσεις, αν τουλάχιστον ένας από τους δύο το επιθυμεί. Επίσης, έχουν τη δυνατότητα να γονιμοποιηθούν εφόσον το επιθυμούν και οι δύο.

## **Τερματισμός προσομοίωσης**

Η προσομοίωση τερματίζεται εφόσον ικανοποιηθεί ένα από τα παρακάτω:

1. Ολοκλήρωση στόχου από κάποια ομάδα
2. Εξάλειψη όλων των πρακτόρων μιας ομάδας λόγω έλλειψης ενέργειας.

# Υλοποίηση

Αρχικά, πριν ξεκινήσουμε την υλοποίηση της εφαρμογής, έπρεπε να επιλέξουμε μεταξύ της Unity Engine και της Godot Engine. Η Unity και η Godot είναι δύο δημοφιλείς game engines, καθεμία με τα δικά της χαρακτηριστικά και πλεονεκτήματα. Η Unity είναι γνωστή για την ευρεία χρήση της στη βιομηχανία των παιχνιδιών, προσφέροντας μεγάλη ευελιξία και υποστήριξη για 2D και 3D γραφικά. Ωστόσο, επιλέξαμε την **Godot engine** (έκδοση 4.2.2) για την ανάπτυξη του λογισμικού μας για διάφορους λόγους. Ο κυριότερος λόγος ήταν επειδή θέλαμε να πειραματιστούμε και με άλλες πλατφόρμες εκτός της Unity. Επίσης, η Godot είναι μια καλή open-source λύση πιο ήπια υπολογιστικά σε σχέση με τη Unity.

Για το γραφικό περιβάλλον χρησιμοποιήθηκαν μερικά δωρεάν **assets** από το [Kenney](#). Τα assets που χρησιμοποιήθηκαν αναφέρονται στις [πηγές](#) του documentation.

# Περιγραφή θεωρητικής βάσης

## Διαστάσεις χάρτη

Για καλύτερη κατανόηση της θεωρητικής βάσης της εφαρμογής, θα ξεκινήσουμε από την αρχικοποίηση του δισδιάστατου κόσμου. Όταν ο χρήστη ξεκινάει για πρώτη φορά την εφαρμογή, του ζητείται να συμπληρώσει τις διαστάσεις που θα ήθελε να είχε ο χάρτης σε (Σειρές x Στήλες). Η ελάχιστη και η προεπιλεγμένη τιμή που μπορούν να έχουν οι Σειρές και οι Στήλες είναι **25**, ενώ η μέγιστη τιμή είναι **100**. Έτσι έχουμε:

$$25 \times 25 \leq \text{Map Size}(\text{rows} \times \text{cols}) \leq 100 \times 100$$

$$\text{rows}, \text{cols} \in [25, 100]$$

## Μορφοποίηση εδάφους

Στην συνέχεια, σύμφωνα με τις διαστάσεις του χάρτη μπορούμε να θεωρήσουμε πως το 3% των συνολικών πλακιδίων - tiles - θα περιέχουν εμπόδια τα οποία οι πράκτορες θα είναι σε θέση να αποφύγουν.

$$\text{obstacles} = \lceil \text{rows} \cdot \text{cols} \cdot 0.03 \rceil$$

## Κατανομή πηγών πόρων

Παρομοίως, σύμφωνα με τις διαστάσεις του χάρτη θεωρούμε πως το 1% των συνολικών πλακιδίων - tiles - θα περιέχουν πηγές στις οποίες οι πράκτορες μπορούν να βρουν τους διαθέσιμους πόρους.

$$\text{total sources} = \lceil \text{rows} \cdot \text{cols} \cdot 0.01 \rceil$$

Έχοντας στην διάθεση μας το συνολικό πλήθος πηγών (total sources), μπορούμε με ευκολία να υπολογίσουμε το πλήθος των πηγών στις οποίες οι πράκτορες μπορούν να βρουν τους εκάστοτε πόρους. Επίσης, κάθε πόρος έχει διαφορετικό πλήθος εμφάνισης, δηλαδή βλέπουμε μια διαφορετική συχνότητα εμφάνισης των πόρων στην έκταση του κόσμου. Έτσι, από το σύνολο των διαθέσιμων πηγών, το 30% με 40% αποτελεί πηγές προμήθειας πέτρας, το 40% με 60% αποτελεί πηγές προμήθειας ξυλείας και τέλος το 10% με 20% αποτελεί πηγές προμήθειας χρυσού.

$$wood\ sources \in \left[ \max\{1, \lfloor total\ sources \cdot 0.4 \rfloor\}, \max\{1, \lfloor total\ sources \cdot 0.6 \rfloor\} \right]$$

$$stone\ sources \in \left[ \max\{1, \lfloor total\ sources \cdot 0.3 \rfloor\}, \max\{1, \lfloor total\ sources \cdot 0.4 \rfloor\} \right]$$

$$gold\ sources \in \left[ \max\{1, \lfloor total\ sources \cdot 0.1 \rfloor\}, \max\{1, \lfloor total\ sources \cdot 0.2 \rfloor\} \right]$$

$$resource\ sources \in \{wood\ sources, stone\ sources, gold\ sources\}$$

### Επιλογή πλήθους πρακτόρων ανά χωριό

Ακόμα, το 10% με 20% από το συνολικό πλήθος πηγών (total sources), ορίζει το πλήθος πρακτόρων που θα δημιουργηθούν ανά χωριό.

$$agents\ per\ village \in \left[ \max\{1, \lfloor total\ sources \cdot 0.1 \rfloor\}, \max\{1, \lfloor total\ sources \cdot 0.2 \rfloor\} \right]$$

Επιπλέον, μπορούμε να αναφέρουμε τη μέγιστη ποσότητα που μπορεί να μεταφέρει ο κάθε πράκτορας για κάθε είδος πόρου ξεχωριστά.

$$resource\ type \in \{wood, stone, gold\}$$

$$resource\ maximum\ carrying\ capacity(resource\ type) = \begin{cases} 40 & \text{if } resource\ type = wood \\ 20 & \text{if } resource\ type = stone \\ 3 & \text{if } resource\ type = gold \end{cases}$$

Στο σημείο αυτό, έχοντας στην διάθεση μας το πλήθος πρακτόρων ανά χωριό (agents per village) και την μέγιστη ποσότητα ανά πόρο που μπορούν να μεταφέρουν οι πράκτορες (resource maximum carrying capacity(resource type)), μπορούμε να υπολογίσουμε τον τελικό-νικητήριο στόχο.

το πολλαπλασιάζουμε με 2 διότι υπάρχουν δύο χωριά

$$goal\ resource = agents\ per\ village \cdot resource\ maximum\ carrying\ capacity(resource\ type) \cdot 2$$

Επιπρόσθετα, γνωρίζοντας τον στόχο μπορούμε να υπολογίσουμε το συνολικό πλήθος των πόρων που θα βρίσκονται στο συγκεκριμένο χώρο.



το πολλαπλασιάζουμε με 2 διότι υπάρχουν δύο χωριά

$$total\ map\ resource\ quantity = goal\ resource \cdot 2 \cdot 2$$

επειδή δεν ελέγχουμε κατά τη δημιουργία του χάρτη να μην υπάρχουν κύκλοι (δηλαδή ένας πόρος να μην είναι προσβάσιμος από έναν πράκτορα), έτσι πολλαπλασιάζουμε τους πόρους για να μην υπάρξει περίπτωση να τελειώσουν οι πόροι πριν κάποιο χωριό φτάσει τον νικητήριο στόχο.

Τέλος, θα υπολογίσουμε το πλήθος των πόρων που μπορούν οι πράκτορες να βρουν σε κάθε πηγή.

$$\left\lceil \frac{total\ map\ resource\ quantity}{resource\ sources} \right\rceil$$

Στο τέλος της παραπάνω διαδικασίας έχουμε καταφέρει να δημιουργήσουμε το δισδιάστατο κόσμο όπου θα βρίσκονται οι ευφυείς πράκτορες. Παράλληλα με αυτούς, σύμφωνα με τις παραπάνω διαδικασίες έχουν αρχικοποιηθεί και τα 2 χωριά όπου ανήκουν οι 2 ομάδες των πρακτόρων, όπως επίσης και ο διαχειριστής παιχνιδιού - game manager -.

### Περιγραφή Οντότητας Χωριό

Κάθε οντότητα χωριό - Village - είναι υπεύθυνο για τη διατήρηση των τιμών του στόχου της ομάδας, την προσθήκη ενέργειας σε κάποιον πράκτορα αν αυτό κρίνεται απαραίτητο από τον ίδιο, αλλά και για την διαχείριση της χρησιμότητας του κάθε πράκτορα την δεδομένη στιγμή βασισμένο στα γονίδια του κάθε φορά.

Αναλυτικότερα, κάθε φορά που επιστρέφει ένας πράκτορας στο χωριό μετά τον έλεγχο για την συμπλήρωση ή μη της απαραίτητης ενέργειας, μετράει την ικανότητα του συγκεκριμένου πράκτορα να βοηθήσει καλύτερα την ομάδα του σύμφωνα με τις τιμές των γονιδίων του και την χρονική στιγμή στο παιχνίδι. Για αυτό λοιπόν, υπολογίζουμε ένα δείκτη καταλληλότητας του πράκτορα για την ανάθεση του κατάλληλου πόρου προς συλλογή.

### Υπολογισμός Δείκτη Καταλληλότητας

Αρχικά, πρέπει να υπολογιστεί η σημαντικότητα συλλογής του συγκεκριμένου πόρου από το σύνολο της ομάδας την συγκεκριμένη χρονική στιγμή. Για τον υπολογισμό του δείκτη καταλληλότητας, διαιρούμε την υπολειπόμενη ποσότητα της ομάδας για τον πόρο αυτόν με την συνολική δυνατότητα όλων των πρακτόρων της ομάδας για την μεταφορά του συγκεκριμένου πόρου.

***remaining resource quantity:*** the remaining quantity until completing the resource's goal

***agents' total resource carrying capacity:*** the total resource carrying capacity that all agents together can carry

$$resource\ significance = \frac{remaining\ resource\ quantity}{agents\ total\ resource\ carrying\ capacity}$$

Ύστερα, κοιτάζουμε την βάση γνώσης του συγκεκριμένου πράκτορα ώστε να ελέγξουμε αν υπάρχει κάποια πηγή συλλογής του συγκεκριμένου πόρου από τον πράκτορα.

$$agent\ knowledge = \begin{cases} 0.5 & \text{if agent doesn't have knowledge how to collect the resource} \\ 1 & \text{else} \end{cases}$$

Στη συνέχεια, υπολογίζουμε το ποσοστό των πρακτόρων της ομάδας που τους ανατέθηκε η συλλογή του συγκεκριμένου πόρου, με τον παρακάτω τύπο:

***number of agents gathering this resource:*** number of agents that are gathering this resource

$$proportion\ of\ agents\ assigned\ to\ this\ resource\ goal = 1 - \left( \frac{number\ of\ agents\ gathering\ this\ resource}{agents\ per\ village} \right)$$

Τέλος, ελέγχουμε τα γονίδια του πράκτορα που αφορούν τον συγκεκριμένο πόρο και με την εκτίμηση αυτών βαθμονομούμε τον πράκτορα κατάλληλα.

$$wood\ genes, stone\ genes \in \{00, 01, 10, 11\}$$

$$gold\ genes \in \{0, 1\}$$

$$resource\ genes = \begin{cases} 0.25 & \text{if wood or stone genes} = 00 \\ 0.5 & \text{if wood or stone genes} = 01 \text{ or gold genes} = 0 \\ 0.75 & \text{if wood or stone genes} = 10 \\ 1 & \text{if wood or stone genes} = 11 \text{ or gold genes} = 1 \end{cases}$$

Τέλος, υπολογίζεται ο δείκτης καταλληλότητας για τον συγκεκριμένο πόρο πολλαπλασιάζοντας τις παραπάνω τιμές:

$$\begin{aligned} resource\ capability \\ &= resource\ significance \cdot agent\ knowledge \\ &\cdot proportion\ of\ agents\ assigned\ to\ this\ resource\ goal \cdot resource\ genes \end{aligned}$$

Η παραπάνω διαδικασία θα πρέπει να επαναληφθεί τόσες φορές όσες και οι διαφορετικές πηγές - 3 διαφορετικές πηγές -, ώστε να μπορεί να επιλέξει το χωριό σε ποιο πόρο θα σταλεί ο πράκτορας αυτή τη φορά.

$$\max \{wood\ capability, stone\ capability, gold\ capability\}$$

### Περιγραφή Οντότητας Διαχειριστή Παιχνιδιού

Η οντότητα του διαχειριστή παιχνιδιού - game manager - είναι υπεύθυνη για την συλλογή των πόρων που μεταφέρουν οι πράκτορες στα χωριά. Επίσης, αναλαμβάνει την ενημέρωση του χωριού όταν ένας πράκτορας πεθάνει καθώς και την ανταλλαγή γνώσης μεταξύ δύο πρακτόρων και την γονιμοποίηση δύο πρακτόρων.

### Περιγραφή Οντότητας Ευφυούς Πράκτορα

Η οντότητα ευφυής πράκτορας - Agent - κατά την αρχικοποίηση του, αποκτά κάποια χρωμοσώματα τα οποία του προσδίδουν τα χαρακτηριστικά του. Ένας πράκτορας έχει ως σκοπό να συνεισφέρει στην ομάδα του, ώστε να νικήσουν τους αντιπάλους. Στόχος του είναι να μεταφέρει πόρους πηγές που υπάρχουν σε τυχαίες και άγνωστες, για αυτόν, θέσεις. Έτσι, όταν του ανατίθενται να συλλέξει ένα είδος πόρου για το οποίο δεν έχει γνώση, χρησιμοποιεί έναν τυφλό αλγόριθμο ο οποίος βασίζεται στον τυφλό αλγόριθμο DFS (Depth-first search) ώστε να ανακαλύψει τα πλακίδια - tiles - του χάρτη όσο προχωράει. Στην περίπτωση που αναλάβει την συλλογή ενός πόρου για τον οποίο έχει γνώση για κάποια πηγή, χρησιμοποιεί τον αλγόριθμο Astar. Ο πράκτορας επαναπροσδιορίζει την απόφασή του

σε κάθε πλακίδιο (tile) για να καθορίσει αν θα συνεχίσει την αναζήτηση ή αν, στην περίπτωση που η ενέργειά του (energy) είναι κάτω από το 33%, θα επιστρέψει στο χωριό (village) για να αναπληρώσει την ενέργειά του στο 100% και να λάβει νέο στόχο. Αν ο πράκτορας δεν καταφέρει να επιστρέψει έγκαιρα στο χωριό, τότε πεθαίνει. Εάν κατά την αναζήτηση συναντήσει κάποιον άλλον πράκτορα, μπορούν, αν το επιθυμεί έστω ένας από τους δύο, να ανταλλάξουν τις βάσεις γνώσης τους. Επίσης, αν το επιθυμούν και οι δύο, μπορούν να γονιμοποιηθούν, βελτιώνοντας μερικά από τα χαρακτηριστικά τους.

## Περιγραφή σημαντικών σχεδιαστικών αποφάσεων και στοιχείων υλοποίησης

Η λειτουργικότητα της εφαρμογής καθώς την σχεδιάζουμε για έναν δισδιάστατο κόσμο, γίνεται με την χρήση της μεθόδου “`physics_process`”. Αυτή έχει την δυνατότητα να καλείται με σταθερό ρυθμό, δηλαδή 60 φορές το δευτερόλεπτο. Αυτή η σταθερότητα στο βήμα μας εξασφαλίζει μια ντετερμινιστική συμπεριφορά η οποία κρίνεται απαραίτητη και χρήσιμη μια εφαρμογή αυτού του είδους. Τέλος, για τον περιορισμό του πλήθους των υπολογισμών ανά καρέ οι πράκτορες αποφασίζουν για τις κινήσεις τους από πλακίδιο σε μπλοκ και μόνο τότε.

Προσθήκες που θα μπορούσαν να υλοποιηθούν στο μέλλον:

- Ελέγχουμε τον τρόπο με τον οποίο τοποθετούνται τα εμπόδια (obstacles) στον χάρτη, ώστε να μην υπάρξει περίπτωση δημιουργίας κύκλου όπου μέσα σε αυτόν να υπάρχει κάποιος πόρος ή χωριό. Σκοπός είναι να εξασφαλίσουμε ότι όλοι οι πόροι και τα χωριά είναι προσβάσιμα. Επίσης, είναι σημαντικό τα εμπόδια να μην διαχωρίζουν τον χάρτη σε επιμέρους τμήματα.
- Η τυχαιότητα και ο τρόπος τοποθέτησης των χωριών, πόρων και εμποδίων να είναι πιο έξυπνη. Δηλαδή να μην μπορούν τα χωριά και οι πόροι να βρίσκονται το ένα δίπλα στο άλλο, οι πόροι να έχουν αρκετή απόσταση μεταξύ τους ώστε οι ευφυείς πράκτορες να πρέπει να πλοηγηθούν περισσότερο στο χάρτη για να βρουν όλες τις πηγές.
- Στην περίπτωση όπου συναντιούνται δύο πράκτορες διαφορετικού χωριού, το ιδανικό θα ήταν να γίνει ανταλλαγή γνώσης μόνο όταν το επιθυμούν και οι δύο (ακριβώς όπως γίνεται και στην γονιμοποίηση).

# Παράδειγμα εκτέλεσης

## Αρχική Οθόνη παιχνιδιού



Παρουσιάζεται στον χρήστη ο τίτλος, μία σύντομη περιγραφή και οι παράμετροι που μπορεί να επεξεργαστεί πριν τη δημιουργία της προσομοίωσης. Η μορφολογία του χάρτη που δημιουργείται κάθε φορά είναι μοναδική.

## Επεξεργασία μεταβλητών



Ο χρήστης μπορεί να αλλάξει από την αρχική οθόνη τις εξής παραμέτρους:

Μέγεθος χάρτη:

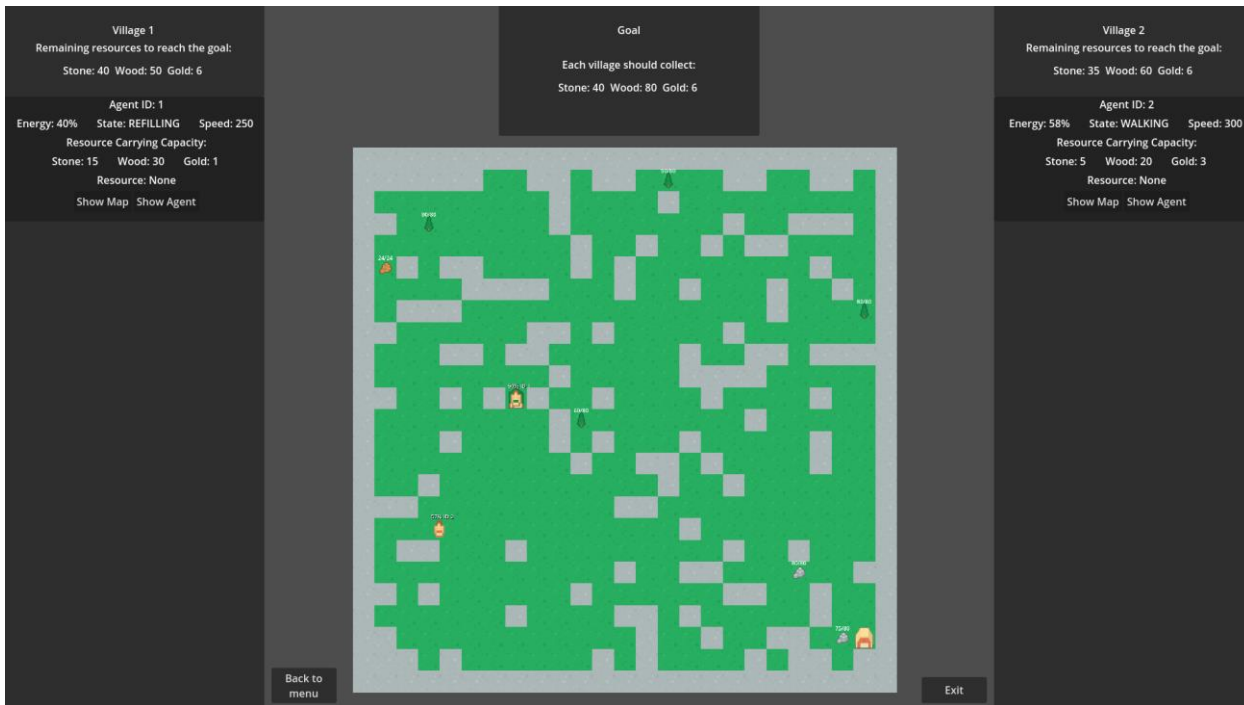
- Σειρές
- Στήλες

Με βάση το μέγεθος του χάρτη, εμφανίζεται ένα συγκεκριμένο πλήθος πόρων. Ο χρήστης μπορεί να επεξεργαστεί τους πόρους αυτούς πριν τη δημιουργία. Οι πόροι είναι:

- Πέτρα
- Ξύλο
- Χρυσό

Τέλος ο χρήστης μπορεί να επιλέξει το πλήθος των πρακτόρων που θα έχει το κάθε χωριό, όπου το όριο του πλήθους υπολογίζεται ξανά σε συνάρτηση με το μέγεθος του χάρτη.

## Οθόνη Προσομοίωσης



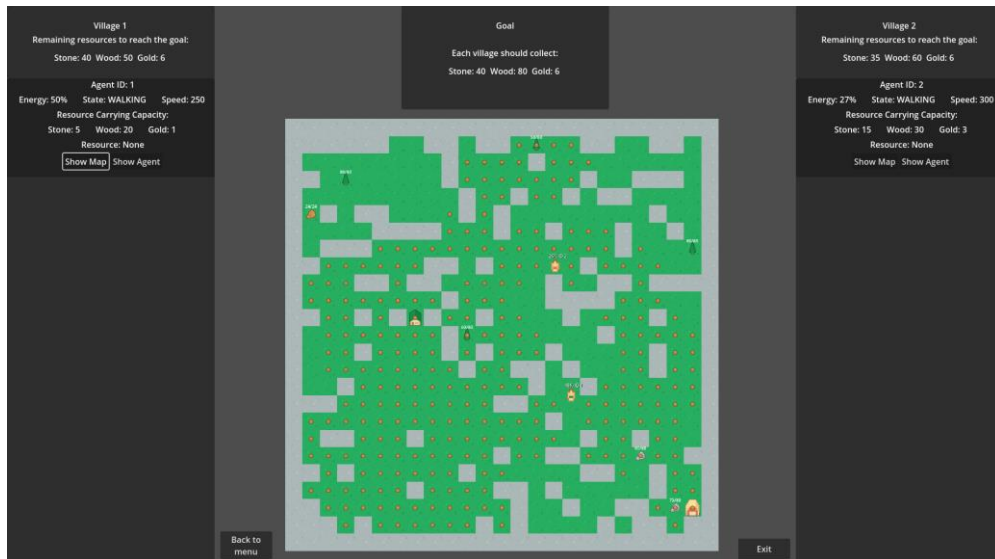
Ο χρήστης με το που ξεκινήσει την προσομοίωση μπορεί να δει δύο στήλες που καταλαμβάνουν την αριστερή και την δεξιά μεριά της οθόνης. Σε κάθε στήλη υπάρχουν οι εξής πληροφορίες:

- Ονομασία χωριού
  - ο Υλικά που απομένουν μέχρι να πετύχει τον στόχο του το χωριό
- Μία καρτέλα με τα βασικά στοιχεία ενός πράκτορα ανά πράκτορα
  - ο Ποσοστό ενέργειας
  - ο Κατάσταση
  - ο Ταχύτητα
  - ο Ικανότητα φορητότητας ανά υλικό
  - ο Κατάσταση κουβαλήματος πόρων
  - ο Δυνατότητα προβολής χάρτη που γνωρίζει
  - ο Δυνατότητα προβολής της θέσης του πάνω στον χάρτη

Υπάρχει και μία τρίτη στήλη που βρίσκεται στο πάνω μέρος της κεντρικής οθόνης όπου αναγράφεται ο στόχος των δύο χωριών, ο οποίος είναι η συγκέντρωση συγκεκριμένου πλήθους πόρων.

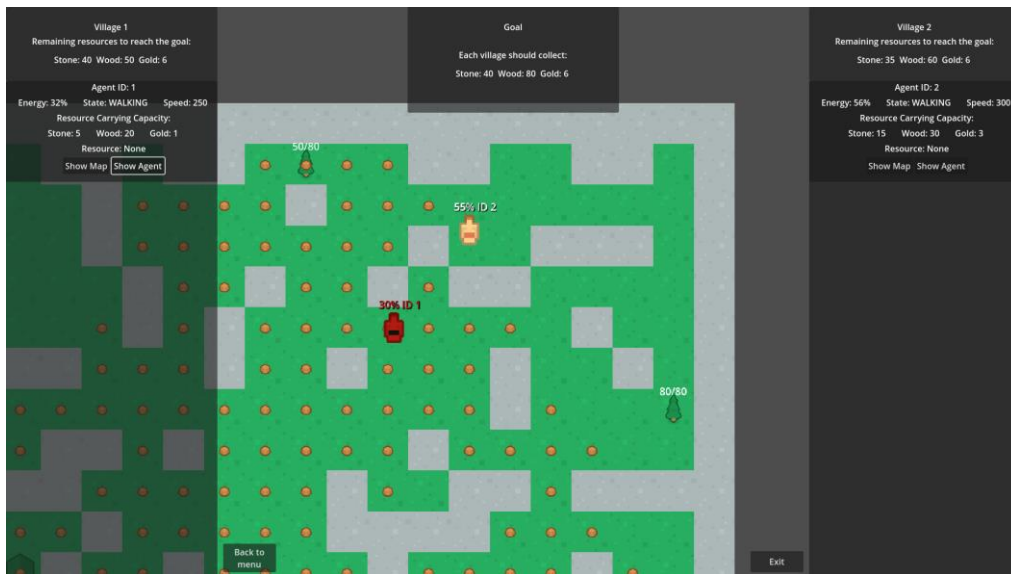


## Εμφάνιση του χάρτη που έχει ανακαλύψει κάποιος agent



Ο χάρτης που έχει ανακαλύψει ο πράκτορες σημειώνεται με πορτοκαλί κουκίδες.

## Παρακολούθηση ενός συγκεκριμένου πράκτορα



Η παρακολούθηση του πράκτορα γίνεται αλλάζοντας το χρώμα του σε σχέση με τους άλλους πράκτορες και εστιάζοντας την κάμερα πάνω του καθώς αυτός κινείται πάνω στον χάρτη.

## Τερματισμός παιχνιδιού

Village 1  
Remaining resources to reach the goal:  
Stone: 40 Wood: 50 Gold: 5

Agent ID: 1  
Energy: 86% State: WALKING Speed: 250  
Resource Carrying Capacity:  
Stone: 5 Wood: 20 Gold: 1  
Carrying: gold (1)  
Show Map Show Agent

Goal  
Each village should collect:  
Stone: 40 Wood: 80 Gold: 6  
Village 2 WON!

Village 2  
Remaining resources to reach the goal:  
Stone: 0 Wood: 0 Gold: 0

Agent ID: 2  
Energy: 91% State: WALKING Speed: 300  
Resource Carrying Capacity:  
Stone: 15 Wood: 30 Gold: 3  
Resource: None  
Show Map Show Agent



Back to menuExit

Το χωριό που κέρδισε σημειώνεται στο κεντρικό μέρος της πάνω οθόνης με την κατάλληλη πρόταση. Στη προκειμένη περίπτωση κέρδισε το Village 2.

# Πηγές

Godot Engine, <https://godotengine.org/>, retrieved 20/05/2024

Kenney, Medieval RTS, <https://kenney.nl/assets/medieval-rts>, retrieved 25/05/2024

How to make a 2D Camera Pan & Zoom in Godot 4,  
[https://www.youtube.com/watch?v=TWCFcv\\_Poc0](https://www.youtube.com/watch?v=TWCFcv_Poc0), retrieved 27/05/2024