
ΕΡΓΑΣΙΑ 3

Μηχανισμοί παιχνιδιού

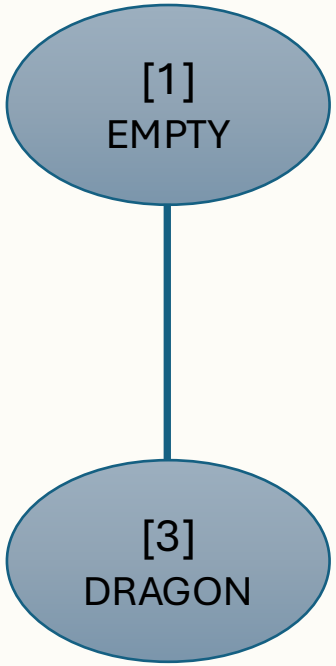
- Ο κόσμος
  - Σύστημα σπηλαίων: Γράφος όπου οι κόμβοι είναι θάλαμοι και οι ακμές σήραγγες
    - Οι σήραγγες είναι αμφίδρομες
- Ο παίκτης
  - Μετακινείται από θάλαμο σε θάλαμο
  - Μπορεί να εκτοξεύει βέλος σε γειτονικό θάλαμο
- Στόχος:
  - Νίκη: Ο παίκτης εκτόξευσε βέλος στο θάλαμο που περιέχει τον δράκο.
  - Αν ο παίκτης μεταβεί στο θάλαμο του δράκου, κερδίζει ο δράκος…

Μηχανισμοί παιχνιδιού

- Περιεχόμενα θαλάμων
  - Νάρκες
    - Καταστρέφουν το θάλαμο (καταστροφή κόμβων/ακμών)
  - Μαγικές πύλες
    - Μετακινούν τον παίκτη σε νεοδημιούργητο θάλαμο (δημιουργία κόμβων/ακμών)
  - Βέλη
    - Οπλίζουν τον παίκτη
- Επιπλέον λειτουργίες
  - Load/Save παιχνιδιού από/σε αρχείο.

Τεχνικά χαρακτηριστικά

- Κόμβος (θάλαμος)
  - Μοναδικό αναγνωριστικό
  - Περιεχόμενα (EMPTY, PORTAL, MINE, ARROW, DRAGON)
  - Δείκτης προς κεφαλή λίστας ακμών προς γειτονικούς κόμβους
- Ακμή (σήραγγα)
  - Δείκτης προς τον γειτονικό κόμβο (του τρέχοντος)
  - Δείκτης προς την επόμενη ακμή της λίστας (άλλοι γειτονικοί κόμβοι του τρέχοντος)
- Γράφος
  - Δυναμικός πίνακας από κόμβους
  - Πλήθος κόμβων
  - Μέγιστο αναγνωριστικό κόμβου που έχει δοθεί



Λεπτομερής εικόνα της δομής του γράφου

graph_t

max_id (int)	3
num_vertices (int)	2
vertices (vertex_t**)	●

vertex_t

id (unsigned int)	1
contents (contents_t)	EMPTY
edges (edge_t*)	●

edge_t

to (vertex_t *)	●
next (edge_t*)	●

NULL

vertex_t

id (unsigned int)	3
contents (contents_t)	DRAGON
edges (edge_t*)	●

edge_t

to (vertex_t *)	●
next (edge_t*)	●

NULL

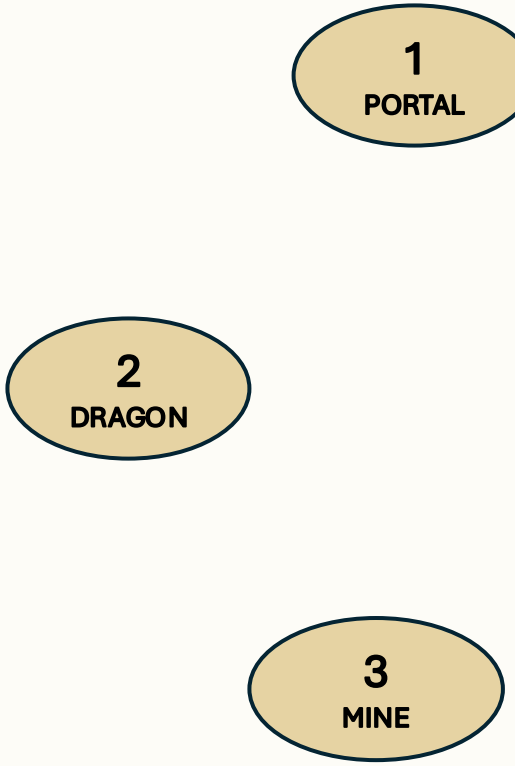
Ο γράφος της εργασίας

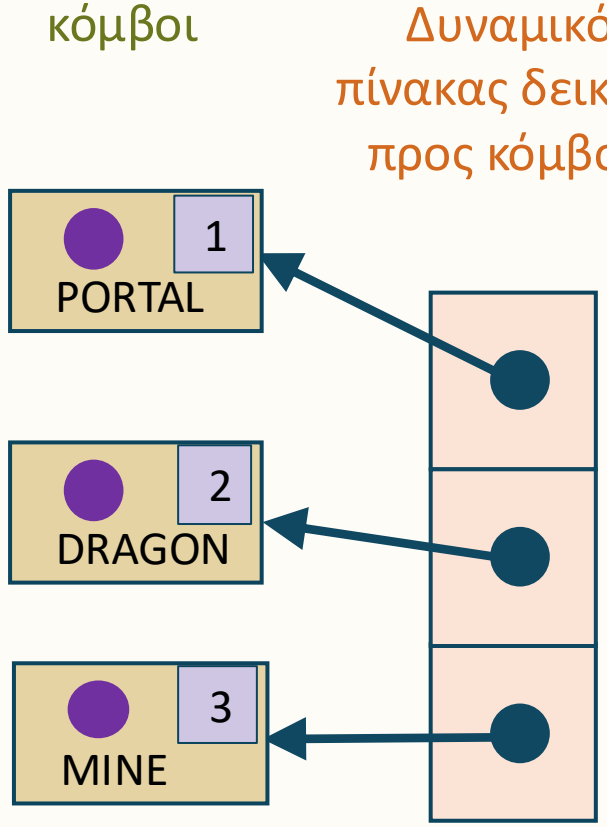
- Ο αρχικός γράφος κατασκευάζεται από έτοιμη συνάρτηση
- Βασικές λειτουργίες που θα χρειαστεί να υλοποιήσετε:
  - Έλεγχος αν δύο κόμβοι είναι γειτονικοί
  - Προσθήκη νέου κόμβου
  - Προσθήκη ακμής
  - Αφαίρεση ακμής
  - Αφαίρεση κόμβου
- Θα χρειαστούν κι άλλες λειτουργίες, π.χ. εύρεση κόμβου με βάση το αναγνωριστικό του, κ.α.

Ο γράφος της εργασίας

- Για τον γράφο που κατασκευάζεται αρχικά από την έτοιμη συνάρτηση, ισχύουν:
  - Υπάρχει πάντα διαδρομή μεταξύ δύο οποιωνδήποτε κόμβων
  - Ο πρώτος κόμβος (θέση [0] του πίνακα κόμβων) είναι πάντα EMPTY
  - Ακριβώς ένας κόμβος περιέχει DRAGON.
- Ακολουθεί παράδειγμα που επιδεικνύει τις βασικές λειτουργίες στο γράφο. Παρότι εσείς ξεκινάτε με έτοιμο γράφο, το παράδειγμα ξεκινά με άδειο γράφο και μετά προσθέτει στοιχεία.
- Για να είναι πιο απλό το σχήμα στο παράδειγμα που ακολουθεί, ο γράφος δεν θα ικανοποιεί τη δεύτερη συνθήκη.

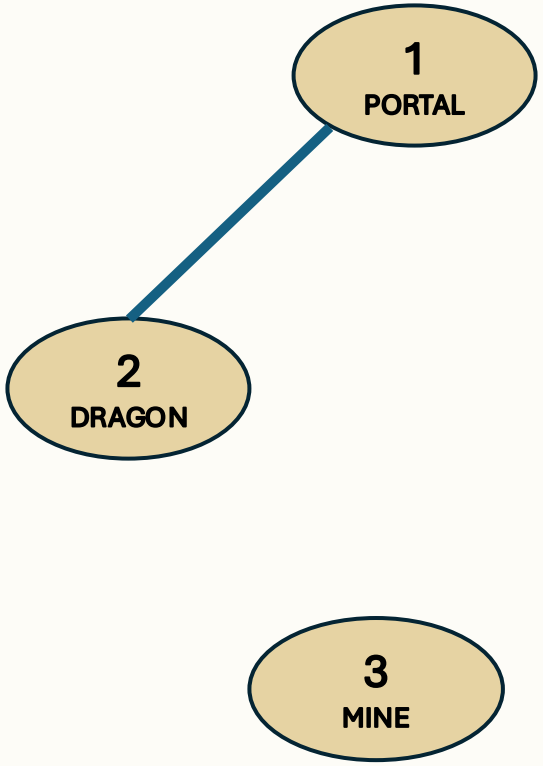
Εισαγωγή κόμβων

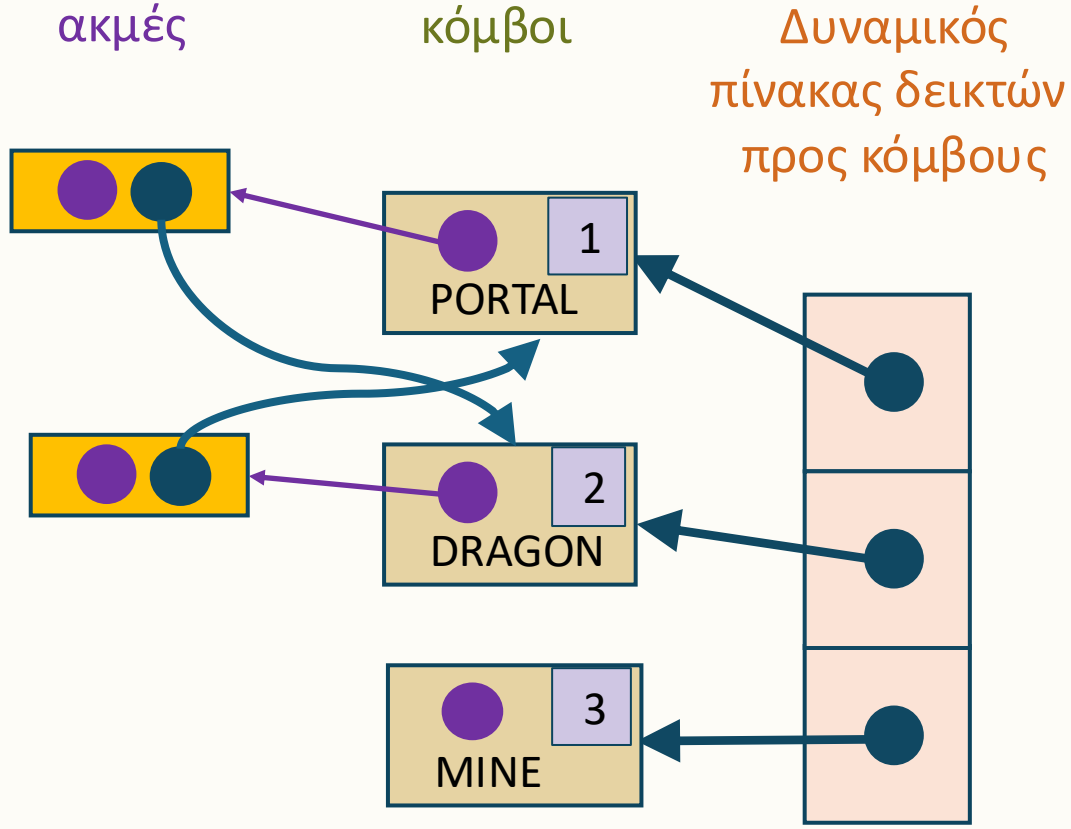




Άδειος γράφος στον οποίο προστέθηκαν 3 κόμβοι.
Προς το παρόν δεν υπάρχουν ακμές.

Εισαγωγή ακμής (ανάμεσα στους κόμβους 1-2)

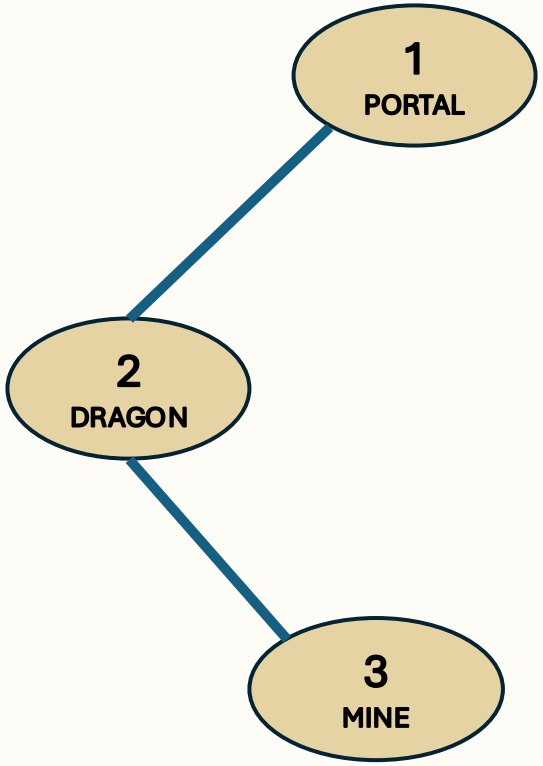


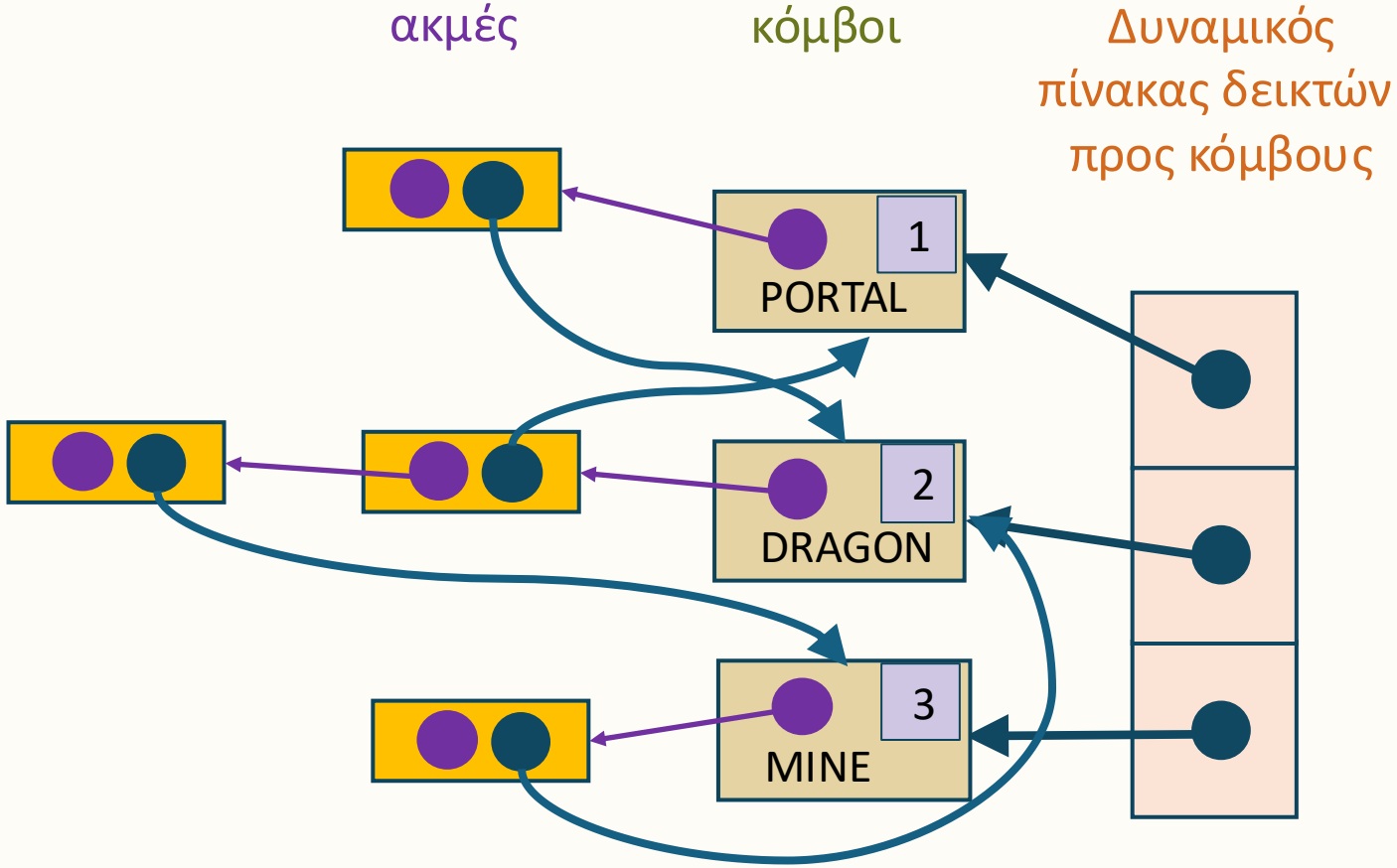


Ο κόμβος 1 γειτνιάζει με τον 2, επομένως η λίστα ακμών του έχει ένα μόνο στοιχείο, με δείκτη προς τον κόμβο 2.

Ο κόμβος 2 γειτνιάζει με τον 1, επομένως η λίστα ακμών του έχει ένα μόνο στοιχείο, με δείκτη προς τον κόμβο 1.

Εισαγωγή ακμής (ανάμεσα στους κόμβους 2-3)

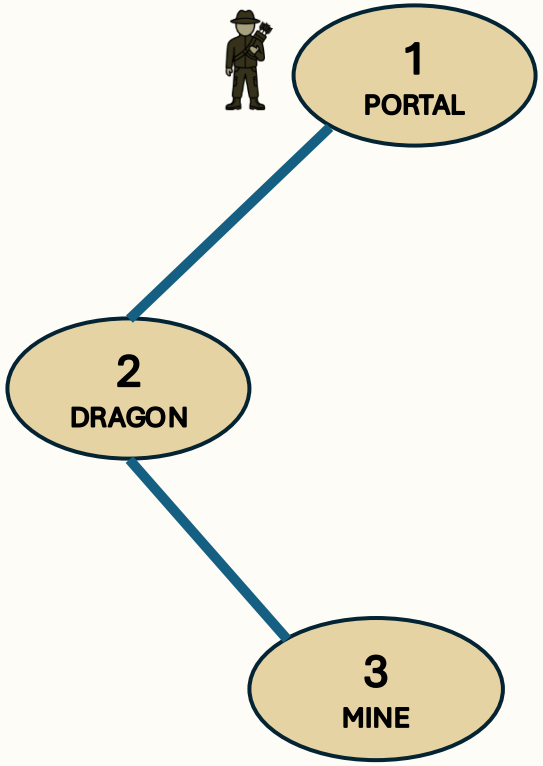


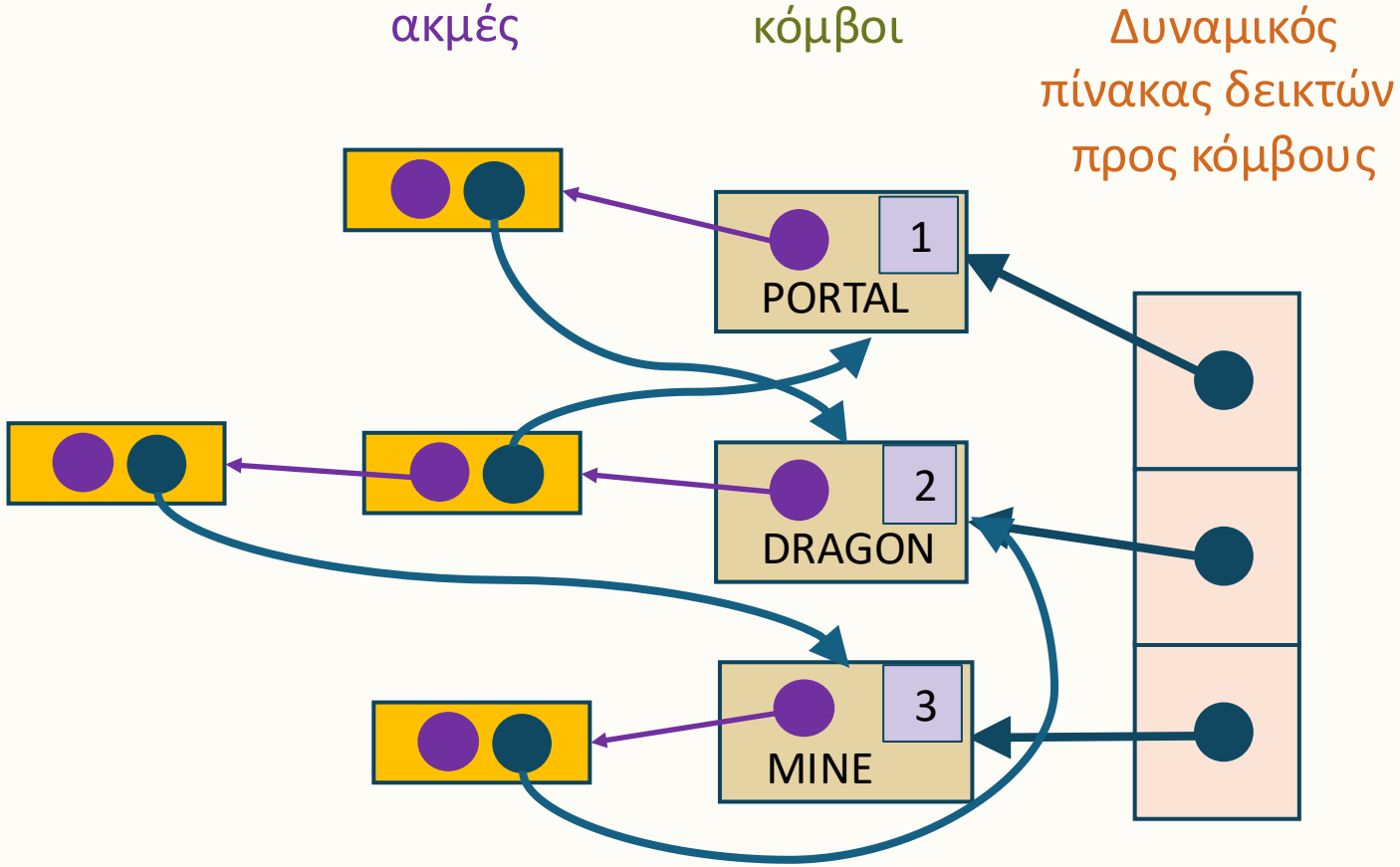


Ο κόμβος 2 έχει ως νέο γείτονα τον 3, επομένως στη λίστα ακμών του προστίθεται νέο στοιχείο με δείκτη προς τον κόμβο 3.

Ο κόμβος 3 γειτνιάζει με τον 2, επομένως στη λίστα ακμών του προστίθεται ένα στοιχείο με δείκτη προς τον κόμβο 2.

Διαχείριση PORTAL

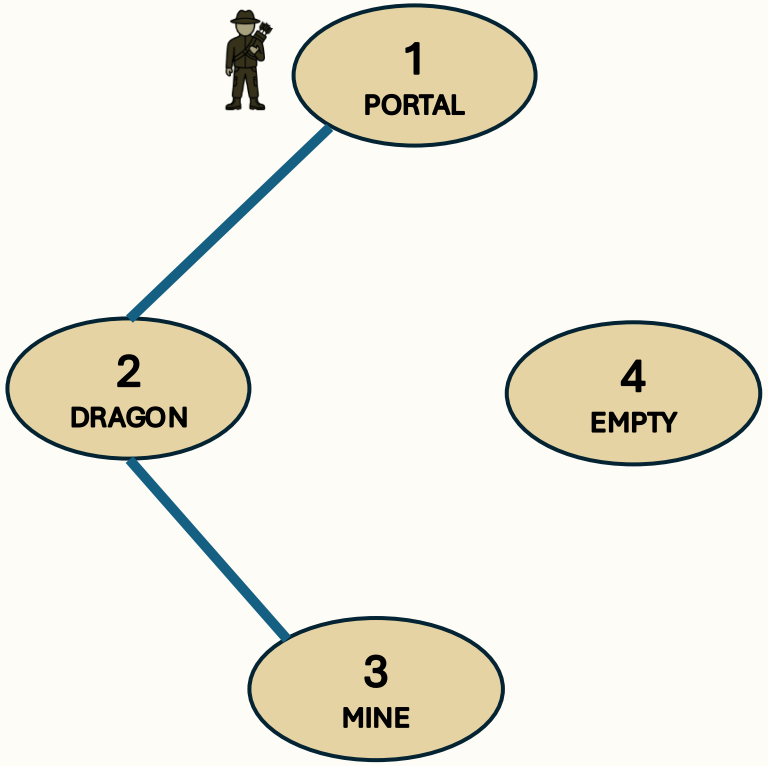


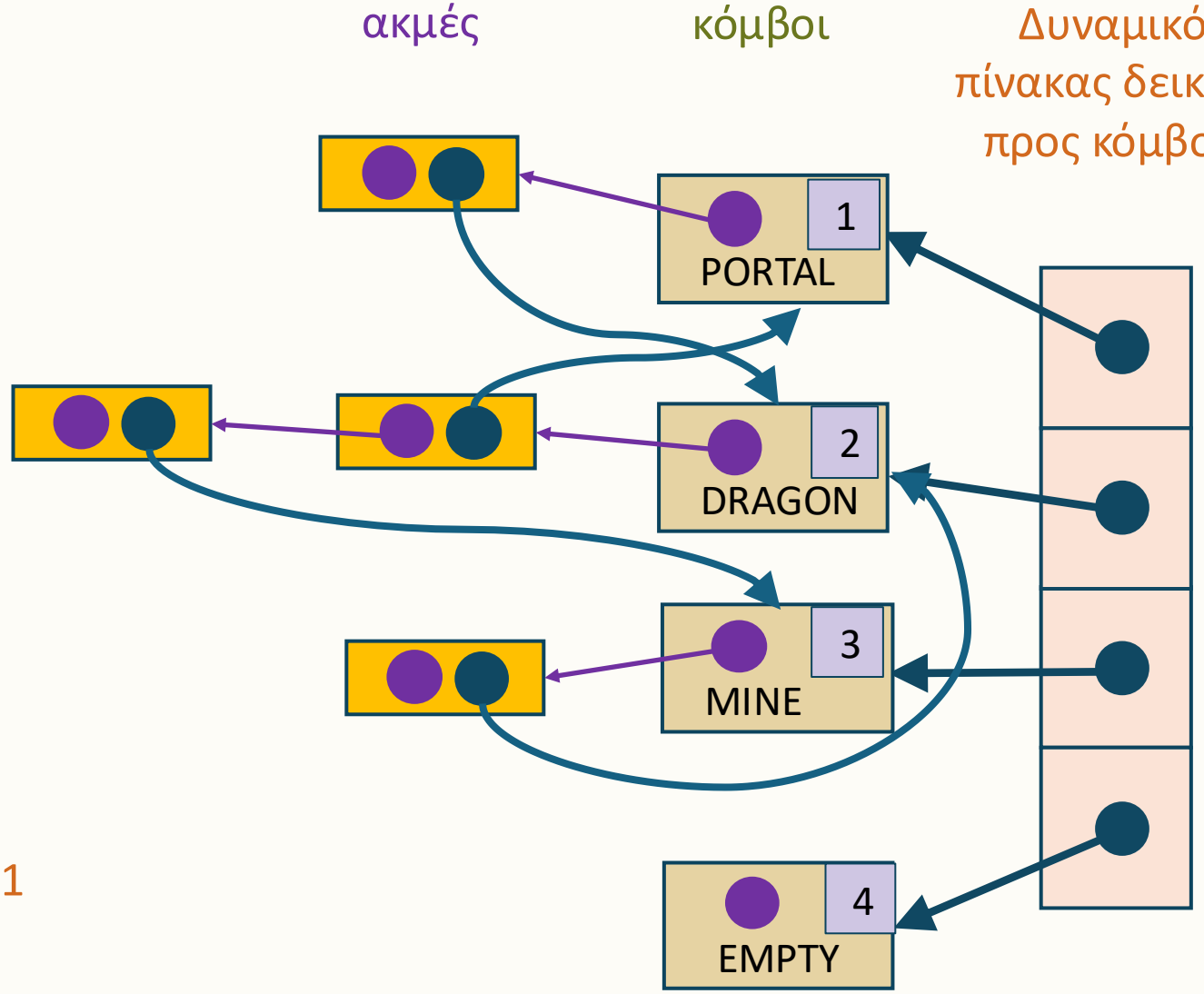


Ο παίκτης μπαίνει στο δωμάτιο 1 (PORTAL). Πρέπει:

1. Να δημιουργηθεί νέος κόμβος.
2. Να συνδεθεί ο κόμβος με δύο άλλους τυχαία επιλεγμένους.
3. Ο παίκτης να μεταβεί στο δωμάτιο που αντιστοιχεί στο νέο κόμβο.

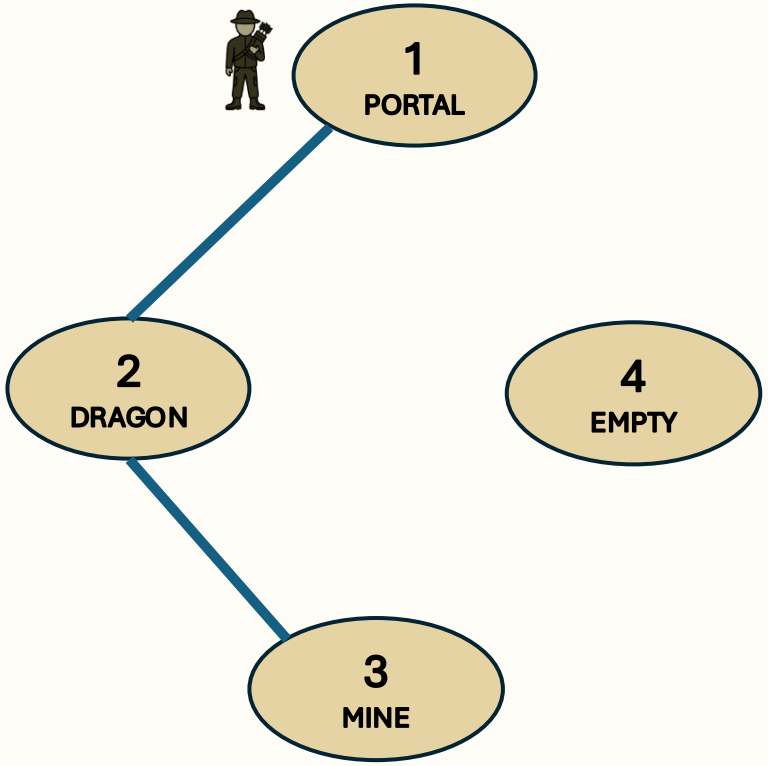
Διαχείριση PORTAL: Δημιουργία νέου κόμβου

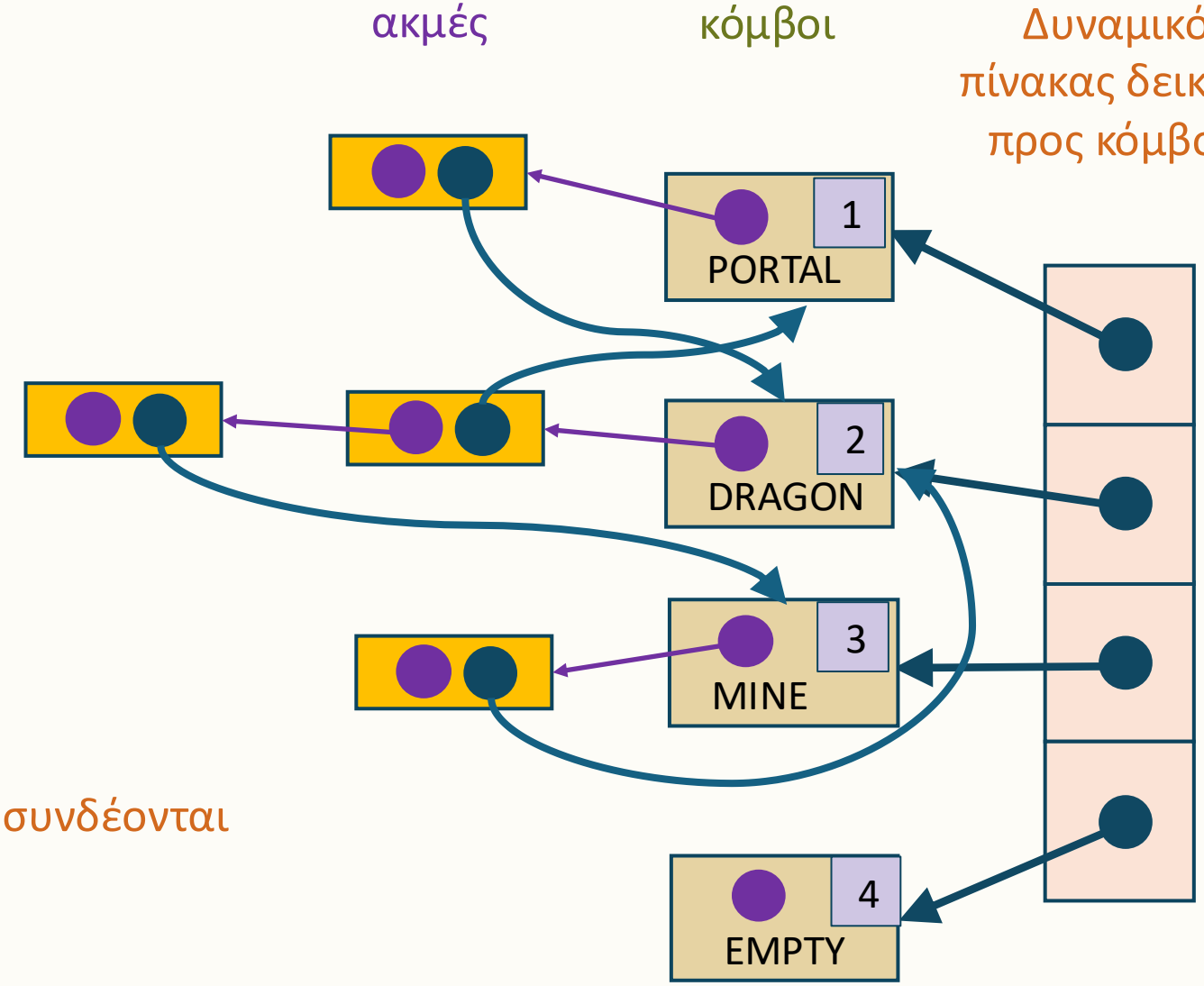




Ο νέος κόμβος είναι άδειος.
Το αναγνωριστικό του είναι όσο το μέγιστο + 1
(Άρα το μέγιστο αυξάνεται!)

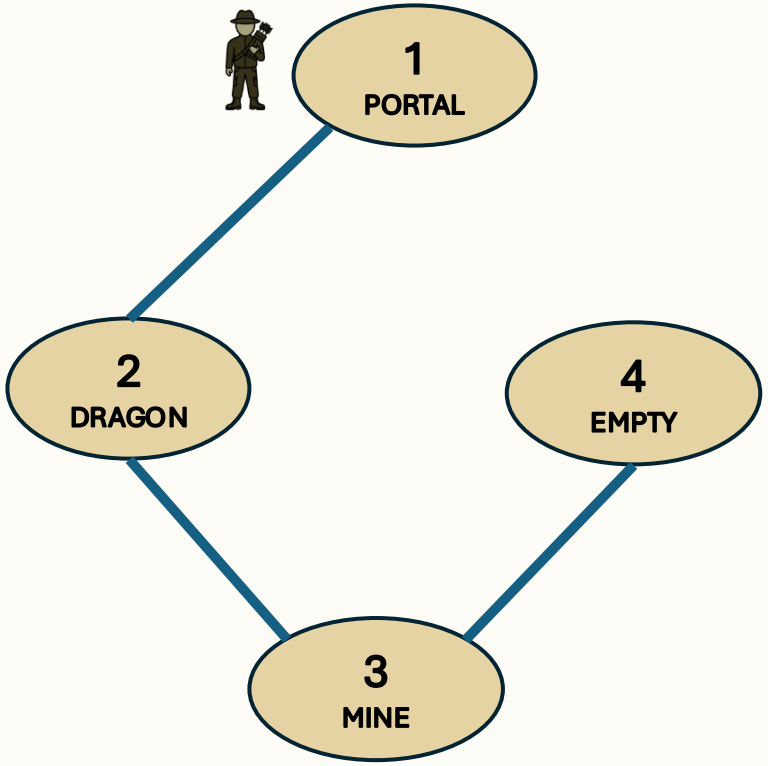
Διαχείριση PORTAL: Σύνδεση νέου κόμβου



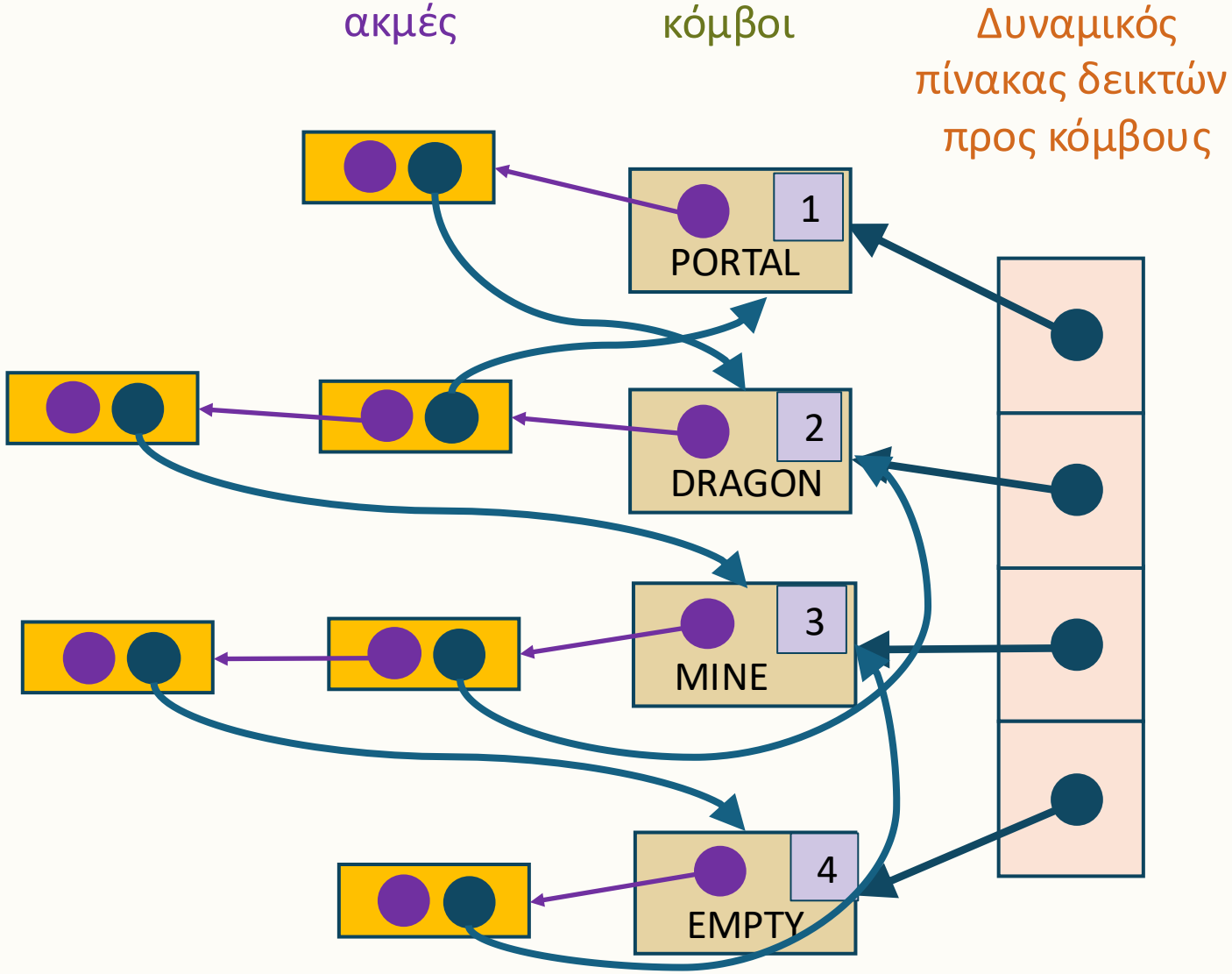


Επιλέγονται στην τύχη δύο άλλοι κόμβοι και συνδέονται με τον νέο κόμβο.
Ας υποθέσουμε ότι επιλέχθηκαν οι 2 και 3.

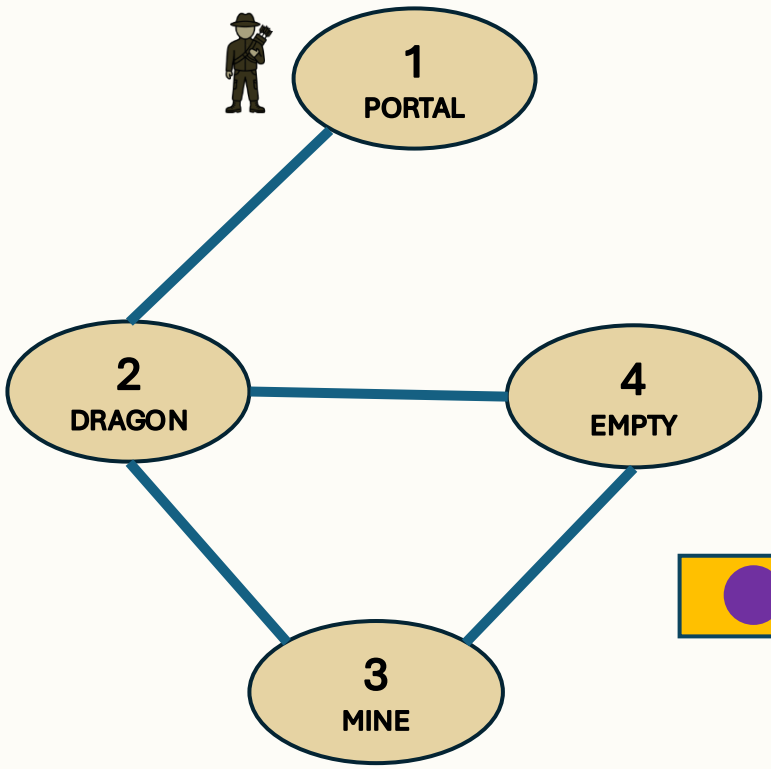
Διαχείριση PORTAL: Σύνδεση νέου κόμβου

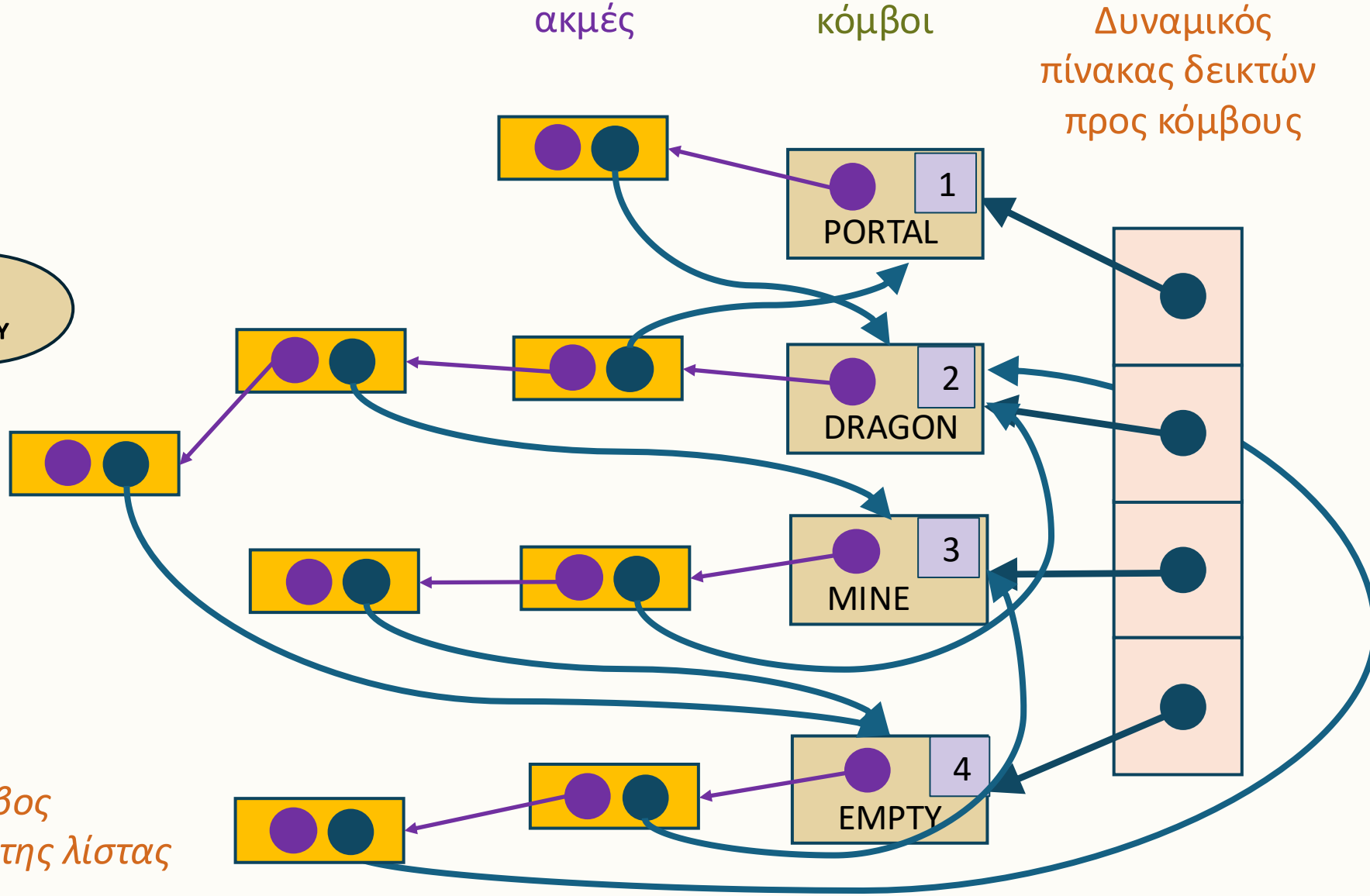


Σύνδεση του 4 με τον 3



Διαχείριση PORTAL: Σύνδεση νέου κόμβου

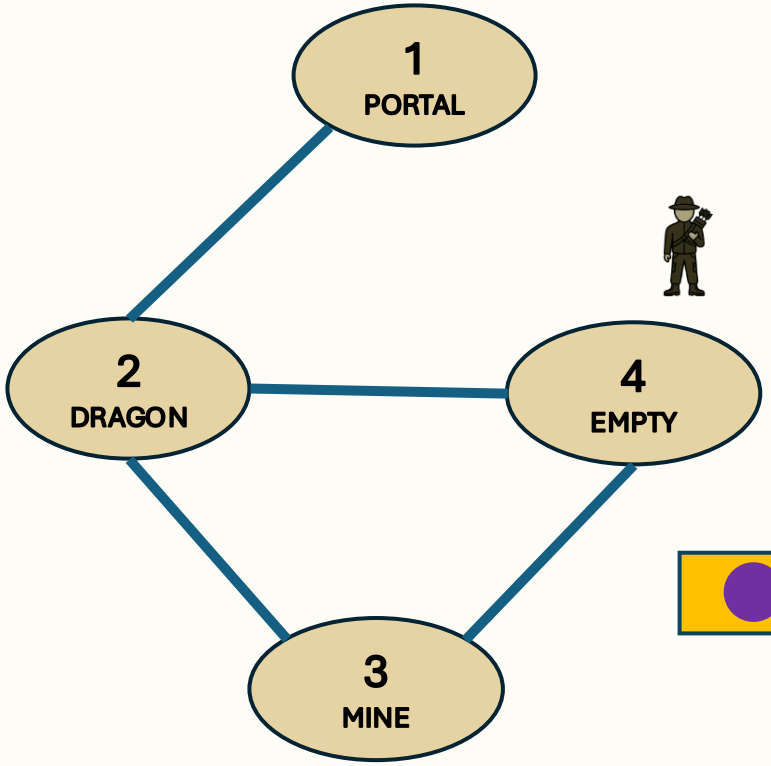


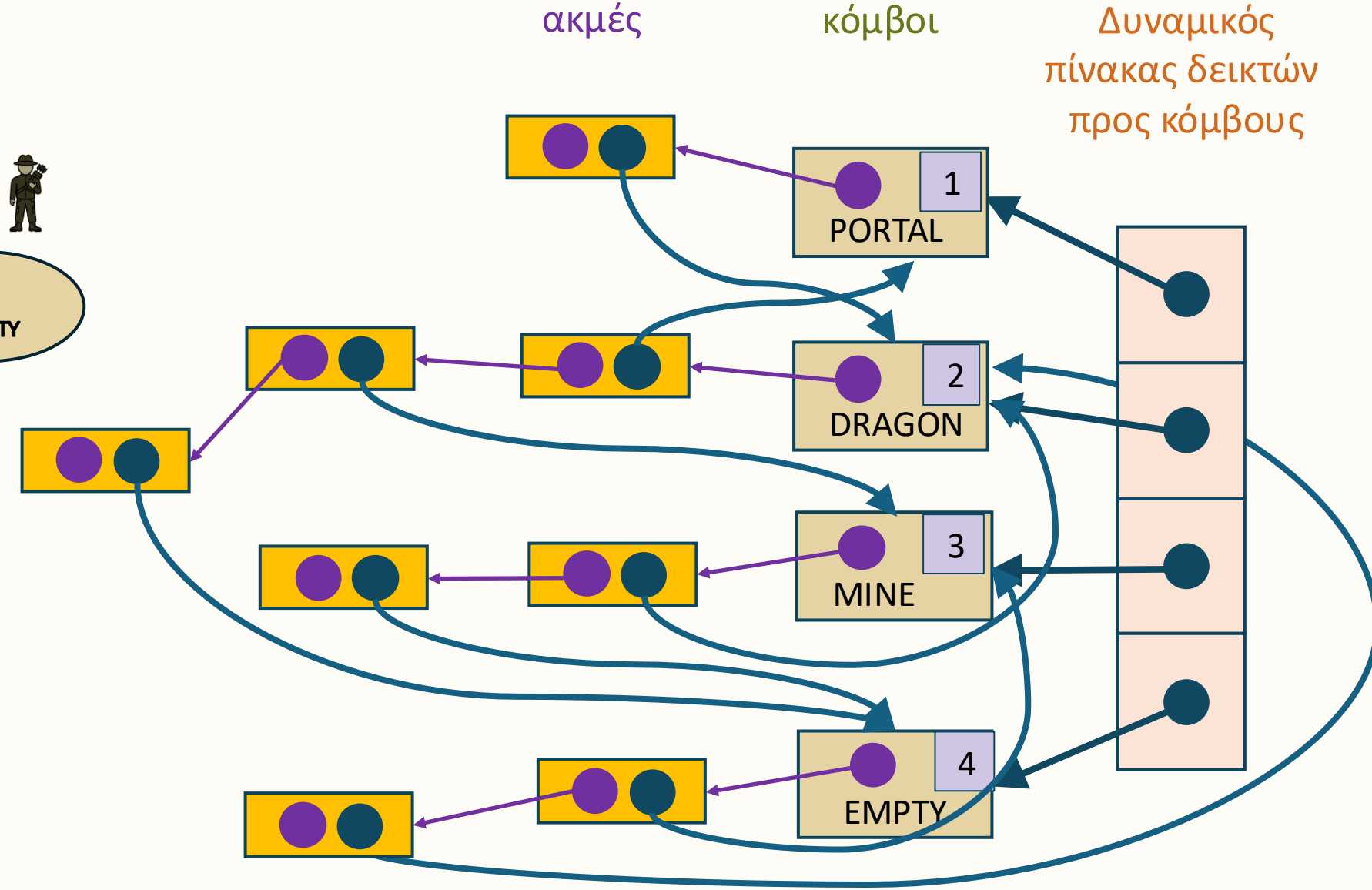


Σύνδεση του 4 με τον 2

Δεν έχει σημασία αν ο νέος κόμβος εισαχθεί στην αρχή ή στο τέλος της λίστας

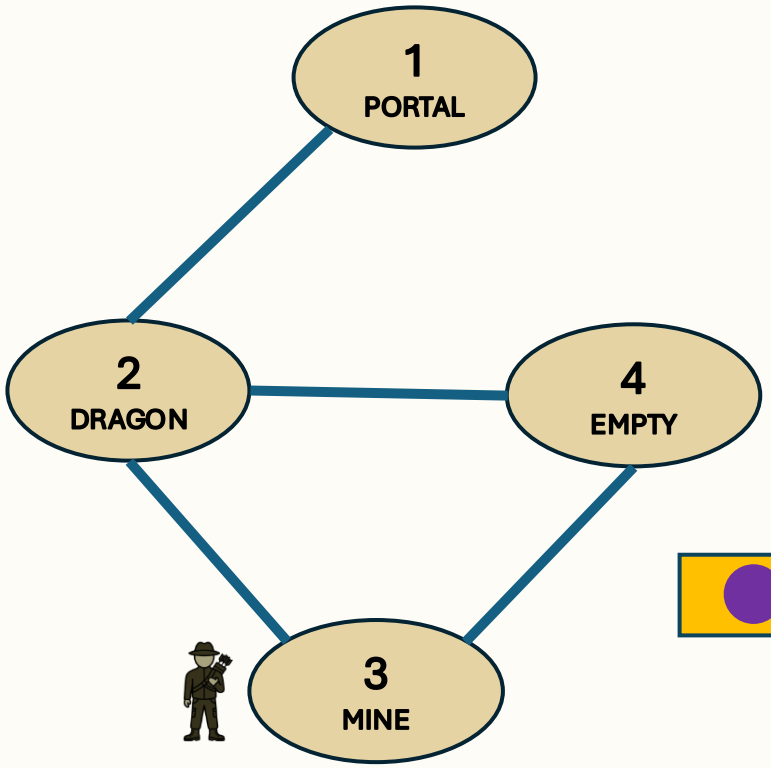
Διαχείριση PORTAL: Μετακίνηση παίκτη

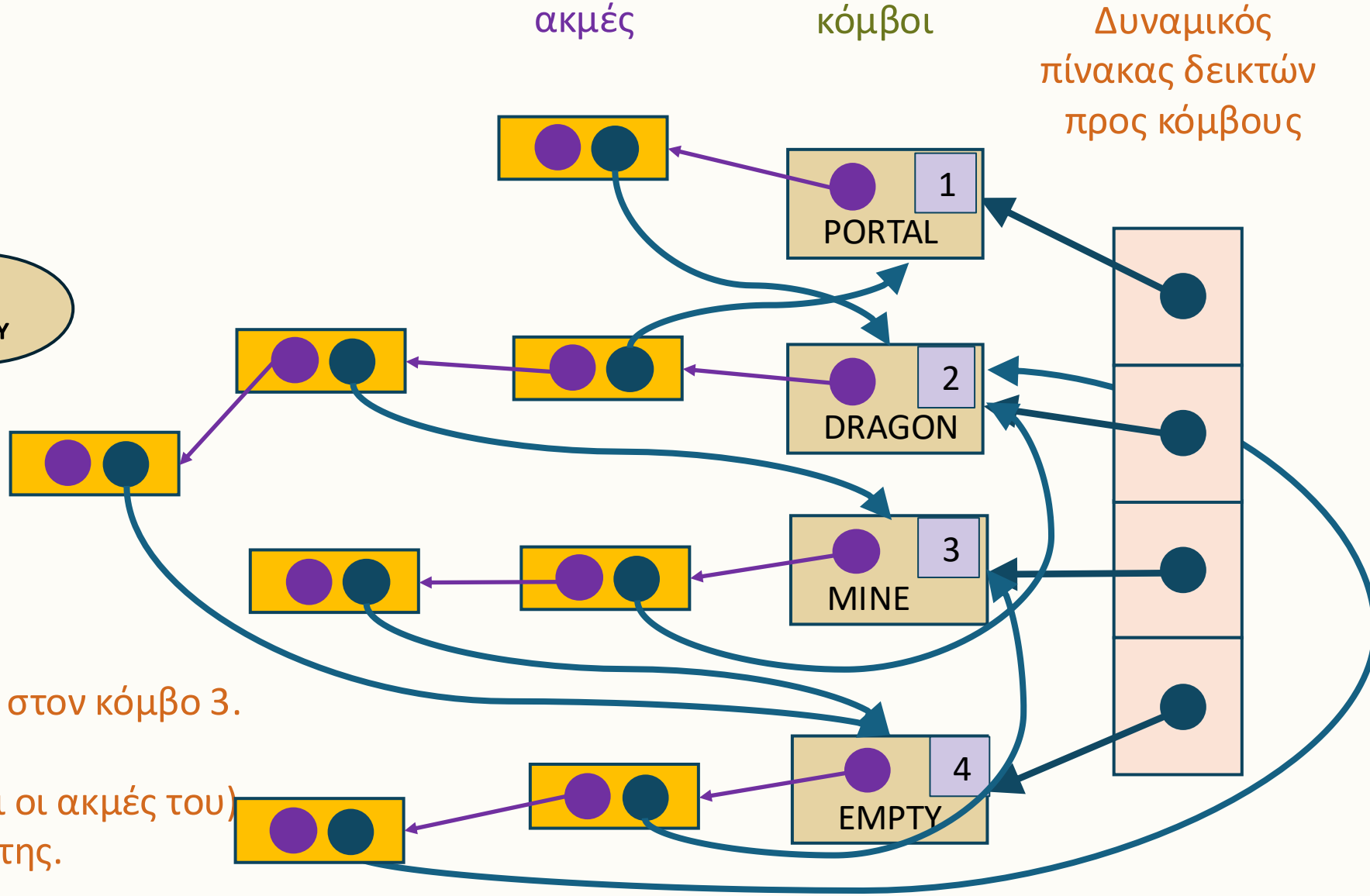




Μετακίνηση στον νέο κόμβο.

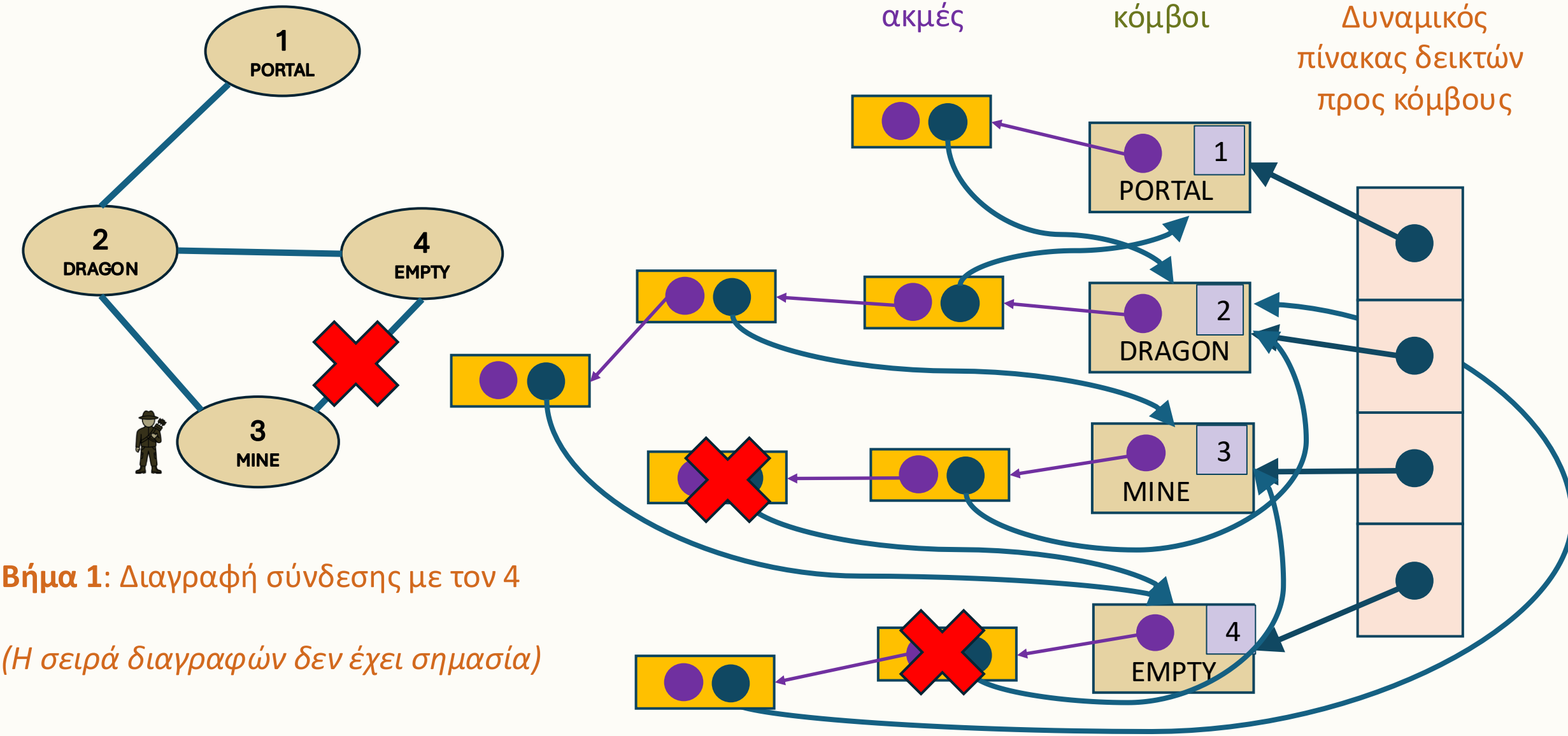
Διαχείριση MINE





Έστω ότι ο παίκτης μετακινείται στον κόμβο 3.
Πρέπει:
1. Να διαγραφεί ο κόμβος 3 (και οι ακμές του)
2. Να μετακινηθεί αλλού ο παίκτης.

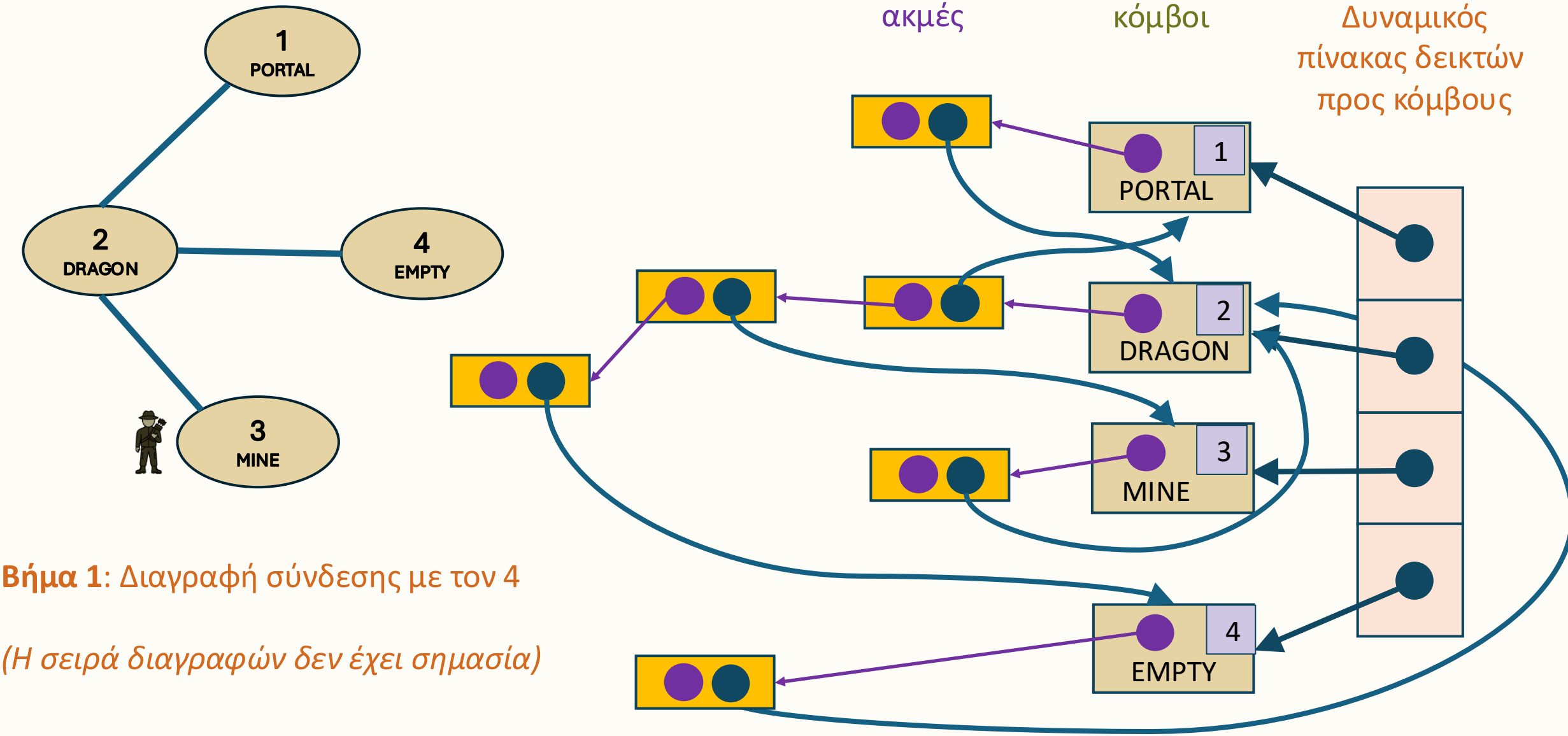
Διαχείριση MINE: Διαγραφή κόμβου 3



Βήμα 1: Διαγραφή σύνδεσης με τον 4

(Η σειρά διαγραφών δεν έχει σημασία)

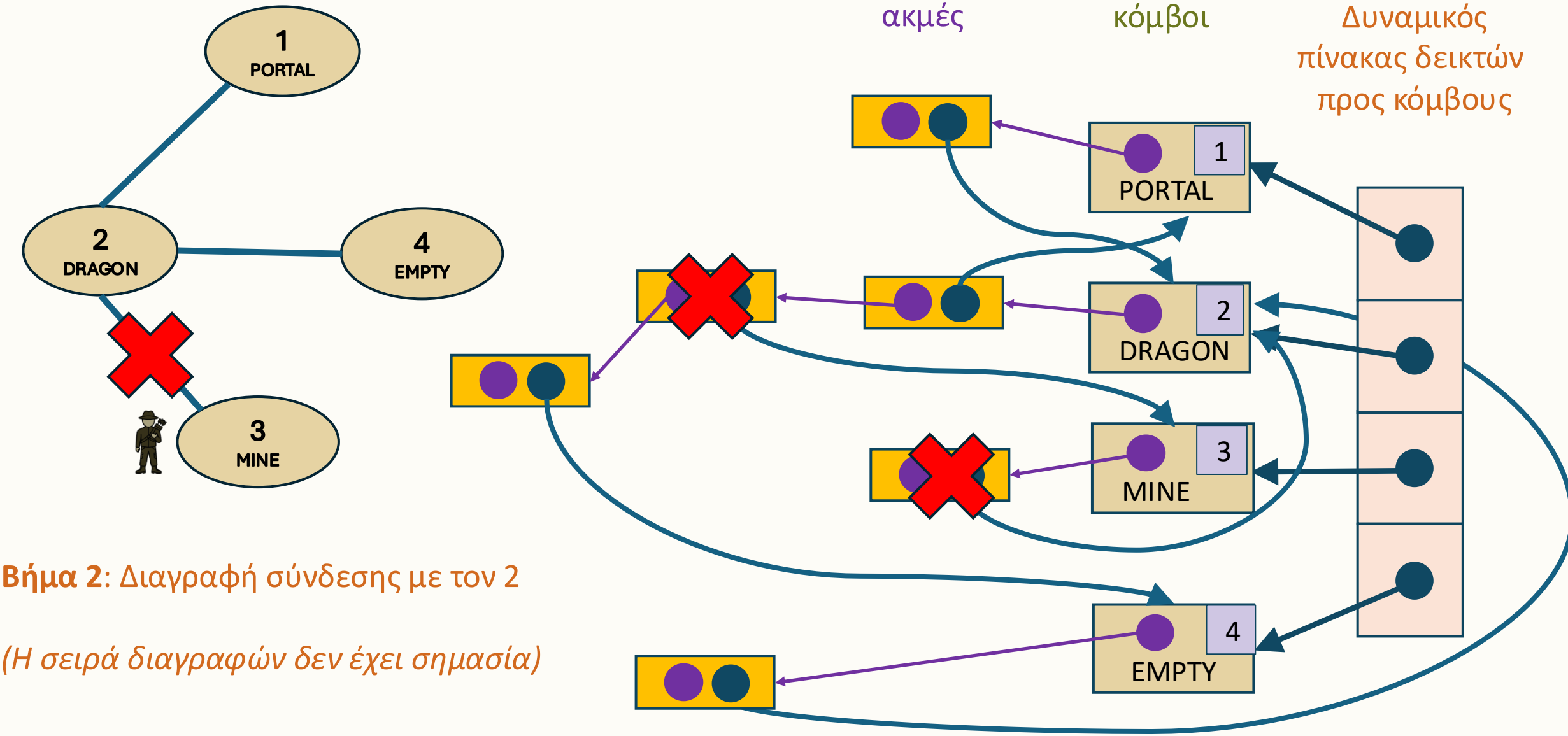
Διαχείριση MINE: Διαγραφή κόμβου 3



Βήμα 1: Διαγραφή σύνδεσης με τον 4

(Η σειρά διαγραφών δεν έχει σημασία)

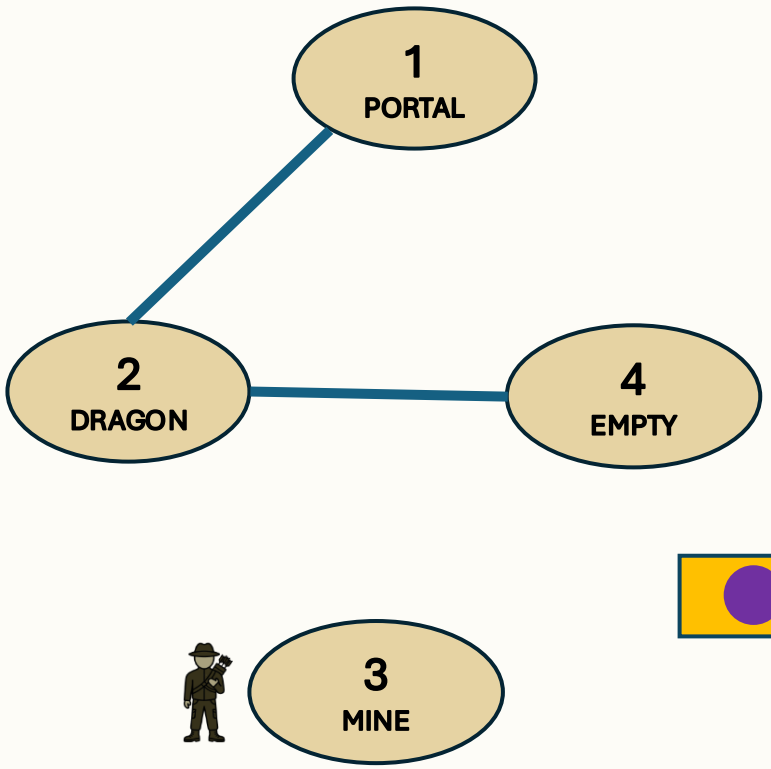
Διαχείριση MINE: Διαγραφή κόμβου 3

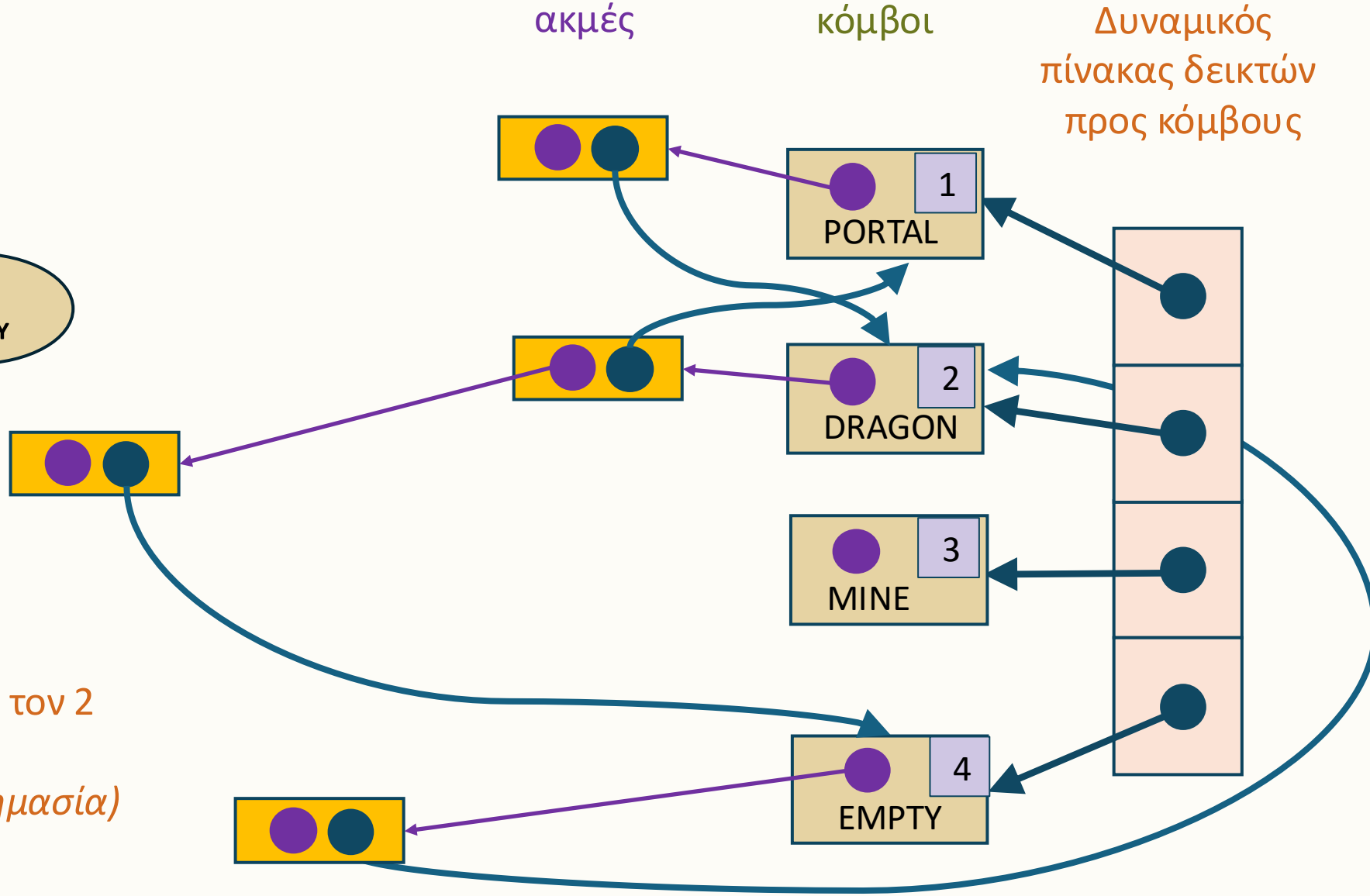


Βήμα 2: Διαγραφή σύνδεσης με τον 2

(Η σειρά διαγραφών δεν έχει σημασία)

Διαχείριση MINE: Διαγραφή κόμβου 3

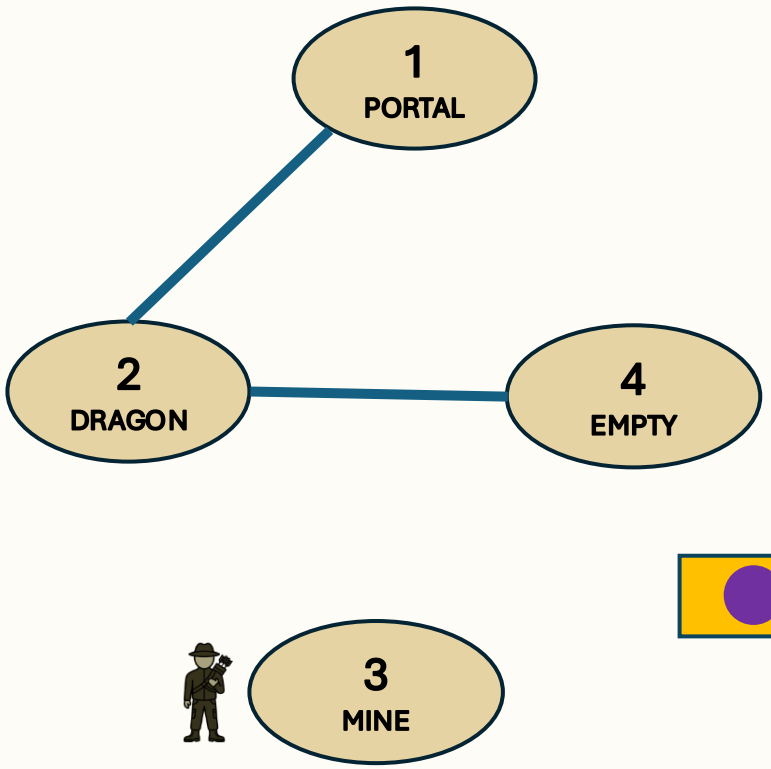


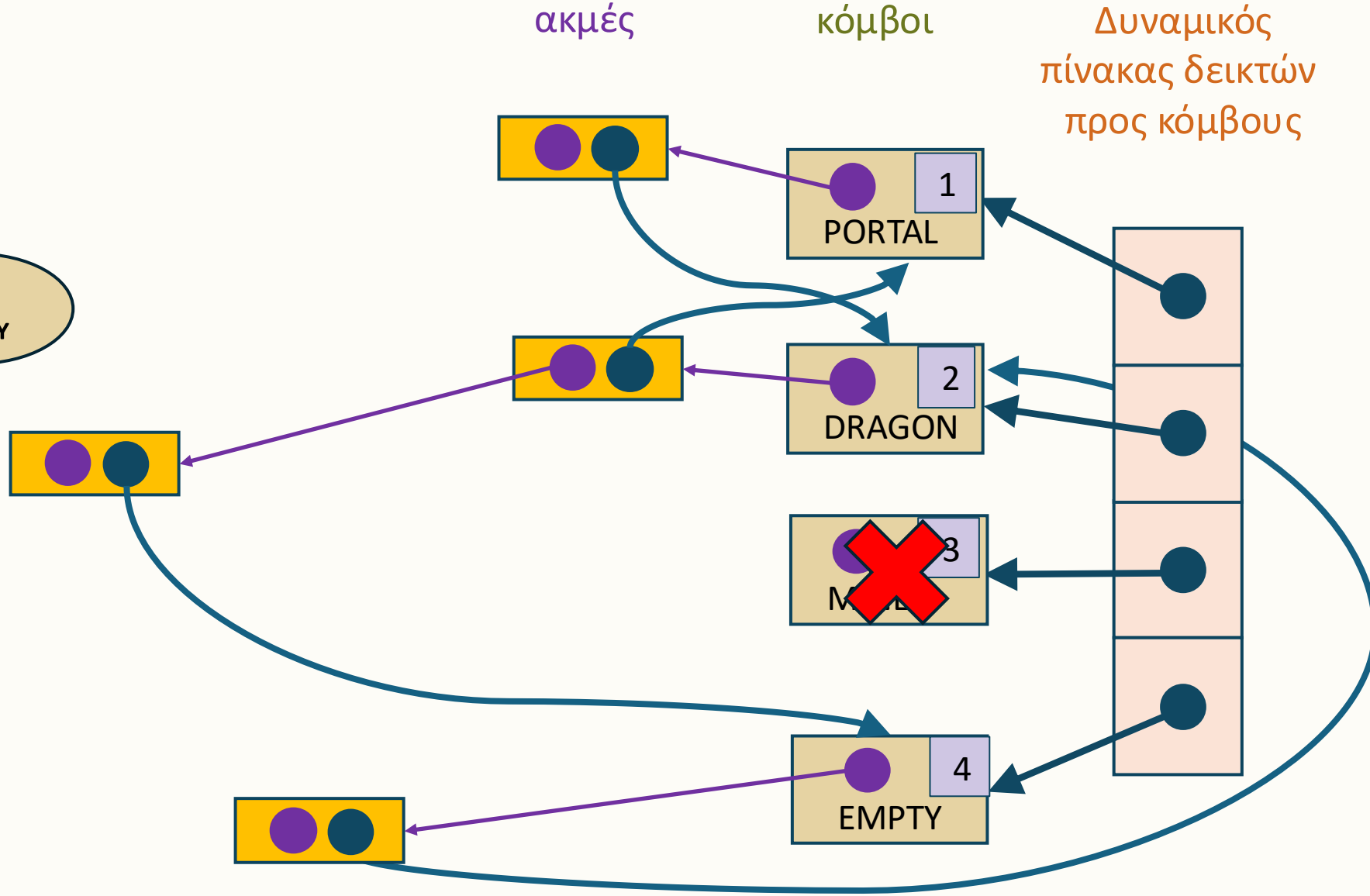


Βήμα 2: Διαγραφή σύνδεσης με τον 2

(Η σειρά διαγραφών δεν έχει σημασία)

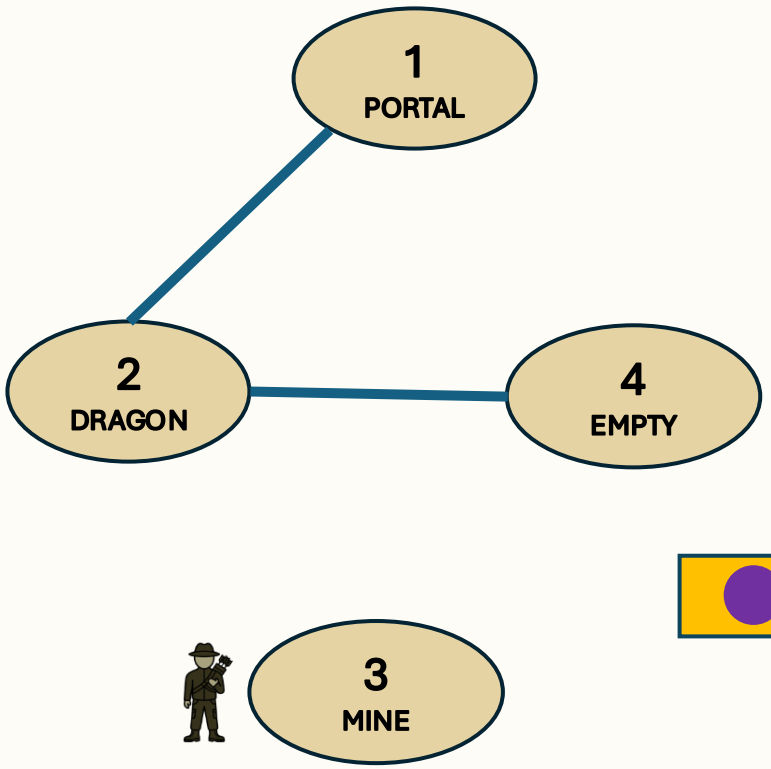
Διαχείριση MINE: Διαγραφή κόμβου 3

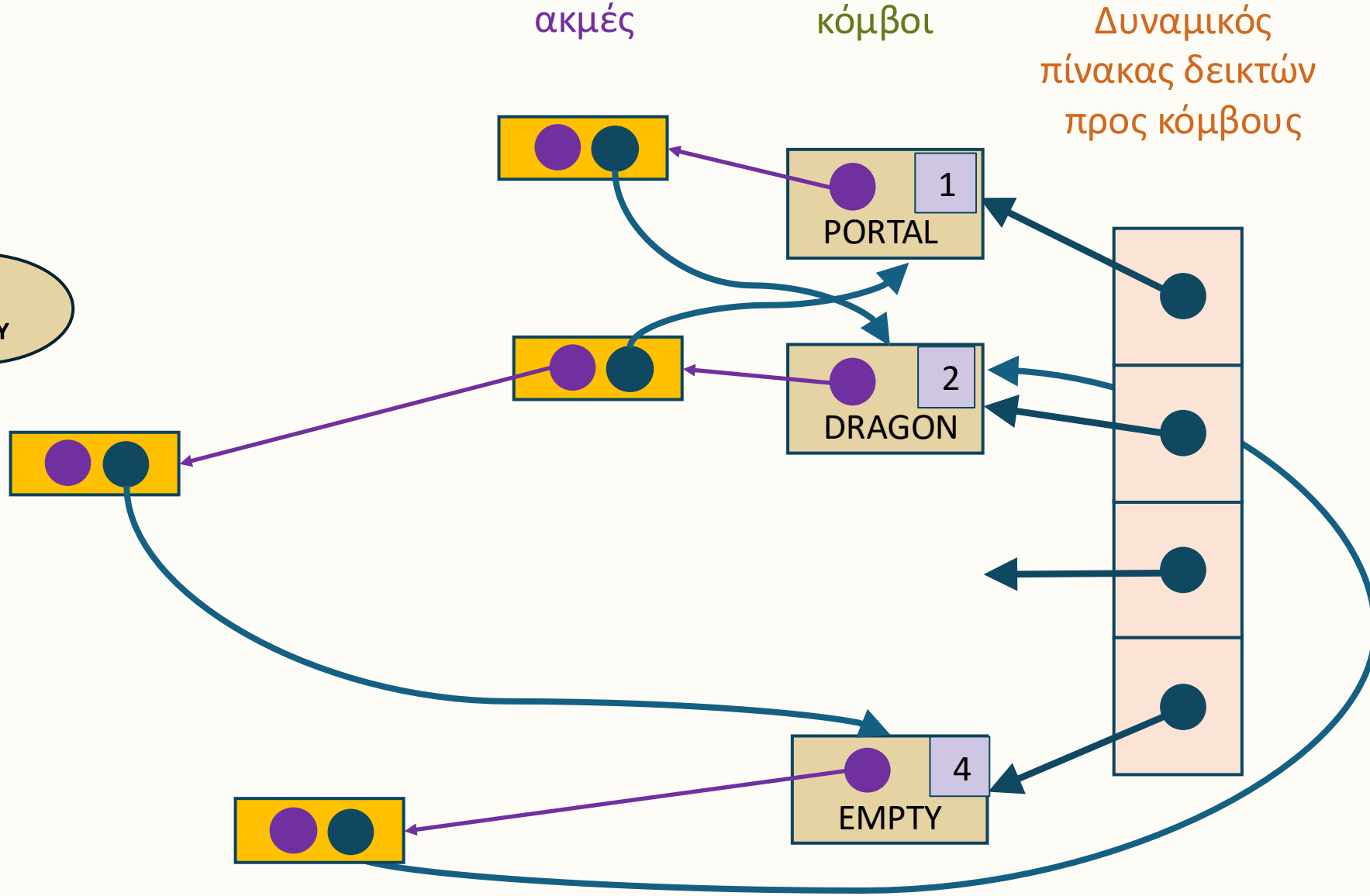




Βήμα 3: Διαγραφή κόμβου

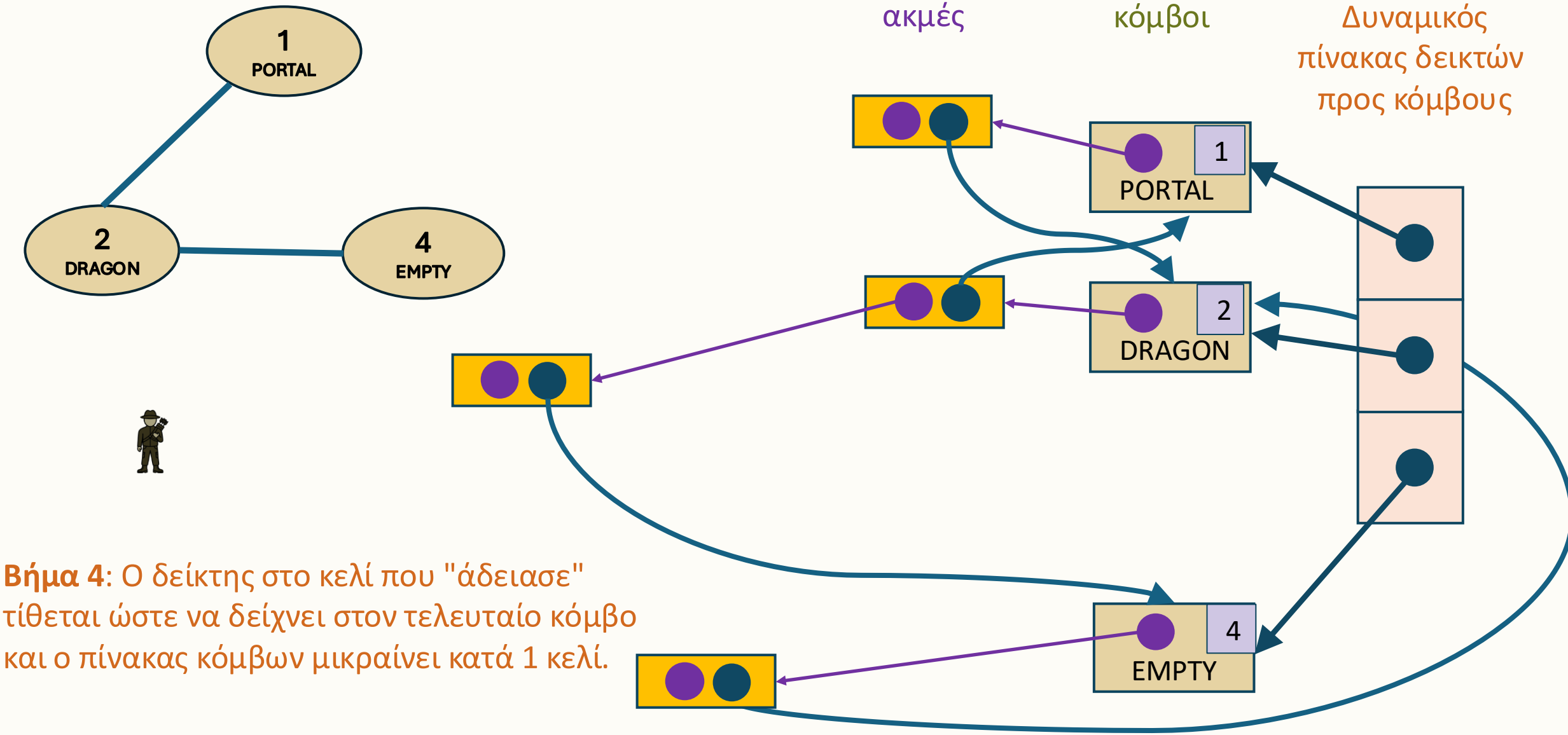
Διαχείριση MINE: Διαγραφή κόμβου 3





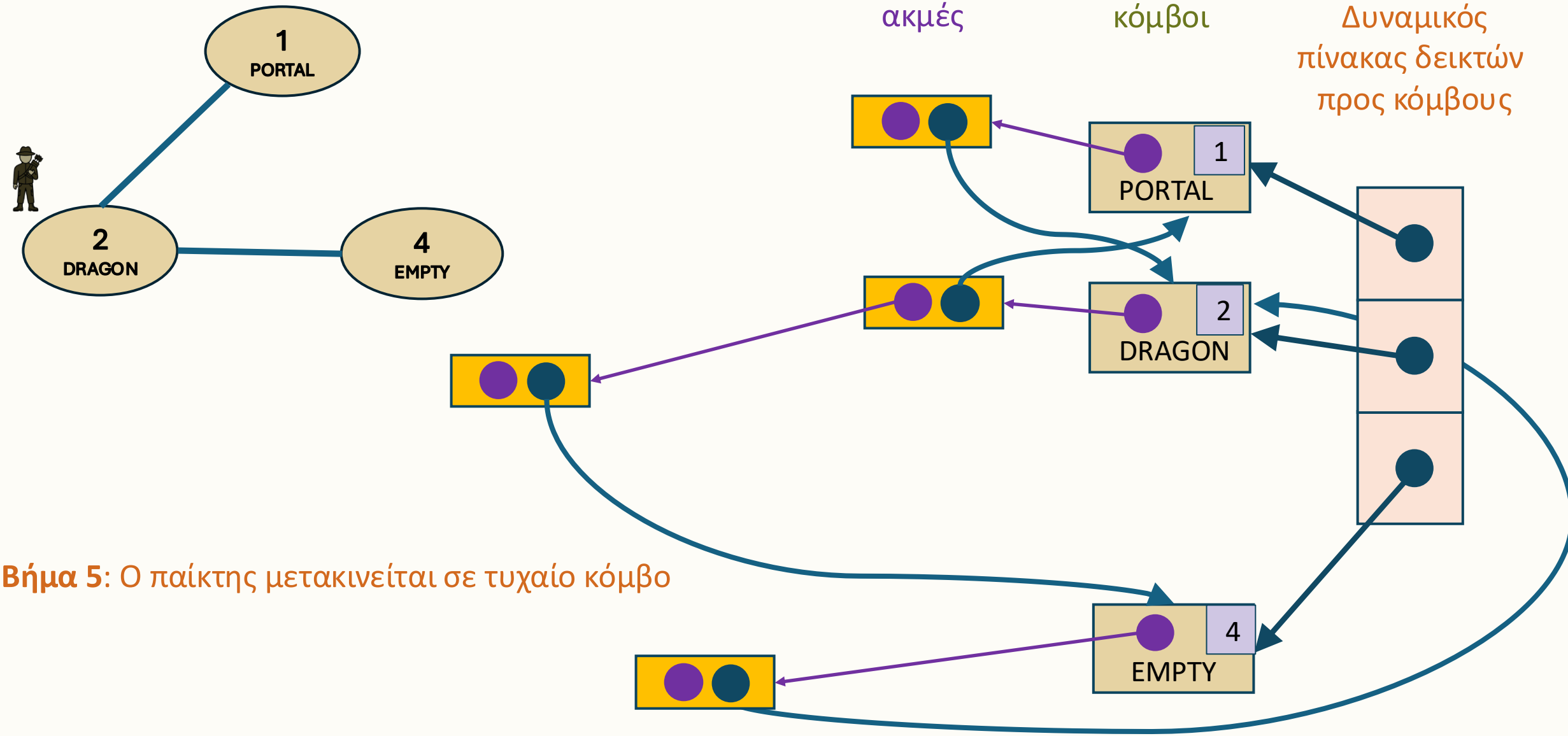
Βήμα 3: Διαγραφή κόμβου

Διαχείριση MINE: Διαγραφή κόμβου 3

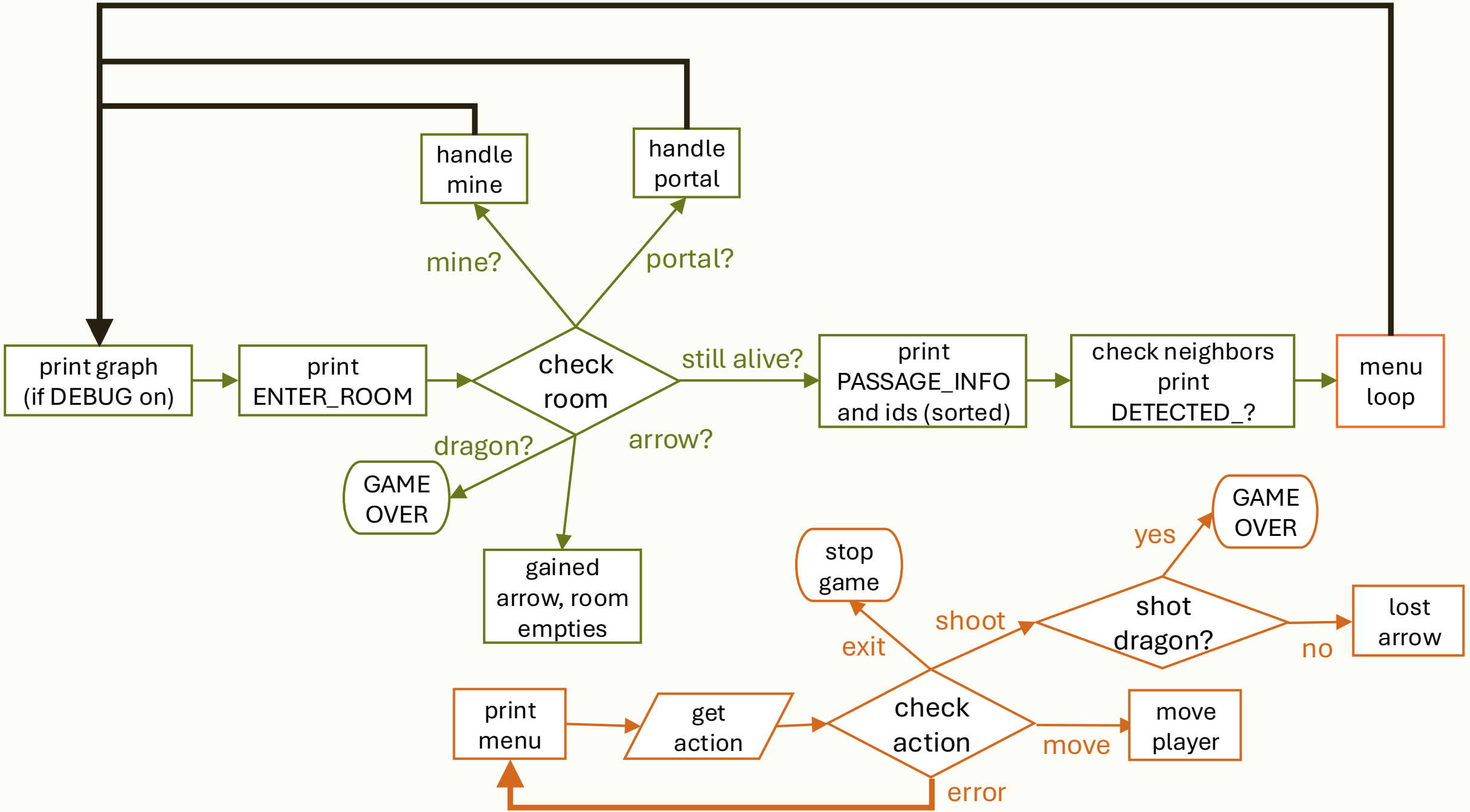


Βήμα 4: Ο δείκτης στο κελί που "άδειασε" τίθεται ώστε να δείχνει στον τελευταίο κόμβο και ο πίνακας κόμβων μικραίνει κατά 1 κελί.

Διαχείριση MINE: Μετακίνηση παίκτη



Βήμα 5: Ο παίκτης μετακινείται σε τυχαίο κόμβο


handle mine
handle portal
mine?
portal?
print graph (if DEBUG on)
print ENTER_ROOM
check room
still alive?
print PASSAGE_INFO and ids (sorted)
check neighbors print DETECTED_?
menu loop
dragon?
arrow?
GAME OVER
gained arrow, room empties
stop game
yes
GAME OVER
shoot
shot dragon?
no
lost arrow
exit
print menu
get action
check action
move
move player
error

Αναγνωριστικά κόμβων

- Κάθε κόμβος έχει ένα αναγνωριστικό (ακέραιος)
- Με αυτό αναφερόμαστε στο θάλαμο που είναι ή που μπορεί να πάει ο παίκτης.
- Διατηρούμε επίσης το μέγιστο αναγνωριστικό που έχει εμφανιστεί στο παιχνίδι (αρχικά μηδέν).
- Αν δημιουργηθεί νέος κόμβος, το αναγνωριστικό του είναι μέγιστο+1
- Αν διαγραφεί ένας κόμβος, το μέγιστο δεν αλλάζει και το αναγνωριστικό του δεν επαναχρησιμοποιείται.
- Το αναγνωριστικό ενός κόμβου δε σχετίζεται με τη θέση του κόμβου στον πίνακα.

Επιλογή "τυχαίου" αριθμού – Λίγη θεωρία

- Δεν είναι δυνατό να παραχθεί αλγοριθμικά ένας πραγματικά τυχαίος αριθμός.
- Αντίθετα, παράγονται "ψευδοτυχαίοι" αριθμοί
  - Μοιάζουν τυχαίοι, αλλά είναι προβλέψιμοι αν γνωρίζουμε την αρχική τιμή.
- Η ιδέα:
  - Ξεκινάμε από έναν αρχικό αριθμό, seed.
  - Εφαρμόζουμε επαναληπτικά ένα μαθηματικό τύπο
  - Κάθε νέο αποτέλεσμα χρησιμοποιείται ως αρχική τιμή για τον επόμενο ψευδοτυχαίο αριθμό.

Επιλογή "τυχαίου" κόμβου στο hw3

- Στο random.c σας δίνουμε την υλοποίηση μιας απλής γεννήτριας ψευδοτυχαίων αριθμών.
  - Δεν είναι σε βιβλιοθήκη, ώστε να δείτε τον κώδικα αν θέλετε.
- Επιπλέον, παρέχουμε τη συνάρτηση
  `get_rand_range(from, to)`
  που επιστρέφει ένα ψευδοτυχαίο ακέραιο στο διάστημα [from, to)
- Στην αρχή της main το seed αρχικοποιείται ώστε να είναι ίσο με το επίπεδο δυσκολίας που επέλεξε ο παίκτης.

Τι δίνεται έτοιμο

- Στο graph.h
  - Τα struct που περιγράφουν κόμβο, ακμή, γράφο
  - Ένα enum που αναπαριστά τα περιεχόμενα ενός θαλάμου (κόμβου)
- Στο msg.h
  - Μια συλλογή από #define που ορίζουν τα κείμενα των μηνυμάτων του παιχνιδιού.
  - Δώστε ιδιαίτερη προσοχή στα μηνύματα για τα οποία απαιτείται επιπλέον παράμετρος στην prinf που θα χρησιμοποιηθούν (π.χ. GREETING_MSG, PASSAGE_INFO_MSG, κτλ.

Τι δίνεται έτοιμο

- Στο game.h
  - Ορισμοί enum για διάφορες καταστάσεις του παιχνιδιού.
  - Ένα struct που αναπαριστά τον παίκτη
  - Prototypes συναρτήσεων εκτύπωσης και διαχείρισης των μενού του παιχνιδιού.
  - Οι υλοποιήσεις των συναρτήσεων βρίσκονται στο game.c
- Στο hw3.c
  - Ο σκελετός της main (με λίγο κρέας)

Τι δίνεται έτοιμο

- Στο random.h
  - Prototypes συναρτήσεων για την παραγωγή ψευδοτυχαίων αριθμών.
  - Οι υλοποιήσεις βρίσκονται στο random.c
- Στο libhw3.a
  - Συνάρτηση δημιουργίας νέου γράφου παιχνιδιού. Δέχεται ως παράμετρο το επίπεδο δυσκολίας
  - Συνάρτηση εκτύπωσης του γράφου
- Στο fileio.h
  - Ο ορισμός του magic number.

Στάδιο 1

- Υλοποιήστε όλες τις λειτουργίες του παιχνιδιού εκτός από:
  - Save/Load
- Ξεκινήστε από τη main
  - Φτιάξτε νέο παιχνίδι, καλέστε μια συνάρτηση play
  - Στο game.c υλοποιήστε την play. Σε κάθε σημείο που απαιτείται μια λειτουργία διαχείρισης του γράφου καλέστε μια αρχικά κενή συνάρτηση (stub – περιέχει μόνο μια ενδεικτική printf και αν χρειάζεται μια return).
  - Στόχος είναι να ολοκληρώσετε τον σκελετό και στη συνέχεια να υλοποιείτε μία – μία τις συναρτήσεις που απαιτούνται.

Save/load

- Στόχος
  - Να αποθηκεύσουμε σε αρχείο την κατάσταση ενός παιχνιδιού που έχει διακοπεί.
  - Να μπορούμε αργότερα (σε διαφορετική εκτέλεση του προγράμματος) να φορτώσουμε την κατάσταση του παιχνιδιού ώστε να συνεχίσουμε.
- Απαίτηση:
  - Η κατάσταση του παιχνιδιού να είναι αποθηκευμένη με τέτοιο τρόπο ώστε να μπορείτε να ανταλλάξετε savefiles μεταξύ σας!
    - Εφόσον δουλεύετε σε συστήματα με ίδια αρχιτεκτονική

Save

- Δείτε στην εκφώνηση την ακριβή δομή του αρχείου
- Το MAGIC_NUMBER απαιτείται για την αναγνώριση του τύπου του αρχείου
  - Εισάγεται σε τρία σημεία στο αρχείο:
    - Στην αρχή
    - Μετά τις βασικές πληροφορίες του γράφου (πλήθος βελών, θέση παίκτη, μέγιστο αναγνωριστικό, πλήθος κόμβων)
    - Στο τέλος
- Χρησιμοποιήστε την εντολή xxd για να δείτε τα περιεχόμενα του αρχείου
  - ΜΗΝ το ανοίξετε με επεξεργαστή κειμένου! Το αρχείο είναι δυαδικό!

xxd

7 κόμβοι

3 βέλη

```
00000000: 4472 3436 304e 3544 334e 0300 0000 0300  Dr460N5D3N......
00000010: 0000 0800 0000 0700 0000 4472 3436 304e  ..........Dr460N
00000020: 3544 334e 0100 0000 0000 0000 0800 0000  5D3N............
00000030: 0100 0000 0300 0000 0000 0000 0400 0000  ................
00000040: 0000 0000 0500 0000 0400 0000 0600 0000  ................
00000050: 0000 0000 0700 0000 0000 0000 0300 0000  ................
00000060: 0600 0000 0700 0000 0800 0000 0200 0000  ................
00000070: 0100 0000 0300 0000 0300 0000 0400 0000  ................
00000080: 0500 0000 0800 0000 0200 0000 0300 0000  ................
00000090: 0700 0000 0200 0000 0300 0000 0600 0000  ................
000000a0: 0300 0000 0100 0000 0500 0000 0700 0000  ................
000000b0: 0300 0000 0100 0000 0400 0000 0600 0000  ................
000000c0: 4472 3436 304e 3544 334e                 Dr460N5D3N
```

ο κόμβος με id 5 περιέχει DRAGON (4 στο enum)

Ο πρώτος κόμβος του πίνακα έχει 3 γείτονες κι αυτοί έχουν αναγνωριστικά 6, 7 και 8.

Ο δεύτερος κόμβος του πίνακα έχει 2 γείτονες κι αυτοί έχουν αναγνωριστικά 1 και 3.

```
==========================================
[      ] <  1 > ---  <  6 > <  7 > <  8 >
[PORTAL] <  8 > ---  <  1 > <  3 >
[ARROW ] <  3 > ---  <  4 > <  5 > <  8 >
[      ] <  4 > ---  <  3 > <  7 >
[DRAGON] <  5 > ---  <  3 > <  6 >
[      ] <  6 > ---  <  1 > <  5 > <  7 >
[      ] <  7 > ---  <  1 > <  4 > <  6 >
==========================================
```

Load

- Πριν κληθεί η συνάρτηση φόρτωσης:
  - Έλεγχος για το αν το αρχείο έχει σωστή δομή
    - Έχει το magic number στα σημεία που πρέπει;
    - Εκμεταλλευτείτε την lseek!
- Εφόσον η δομή του αρχείου είναι σωστή, καλείται η συνάρτηση φόρτωσης
  - Ταυτόχρονα με τη δημιουργία του γράφου από τα δεδομένα του αρχείου, γίνεται έλεγχος ορθότητας των δεδομένων
    - Αρνητικοί ή μη-λογικοί αριθμοί
    - Αναφορές σε αναγνωριστικά που δεν υπάρχουν.
    - Το αρχείο τελειώνει πρόωρα

Load

- Χτίζετε γράφο από την αρχή
  - Ξεκινάτε από άδειο γράφο
  - Προσθέτετε κόμβους
  - Προσθέτετε ακμές
- Επαναχρησιμοποιήστε συναρτήσεις που έχετε ήδη γράψει!
- Αν η λειτουργία αποτύχει, αποδεσμεύετε ό,τι έχει δημιουργηθεί κατά τη διαδικασία της load και παραμένετε με το γράφο που υπήρχε στο παιχνίδι (αν υπήρχε) πριν επιλεγεί LOAD.


GAME OVER

Is it really GAME OVER?

- Μελλοντικές βελτιώσεις
  - Έλεγχος ακεραιότητας αρχείου με checksum
  - Εξασφάλιση φορητότητας αρχείων
    - Σταθερά μεγέθη τύπων
    - Endianness
  - Πολλαπλά αποθηκευμένα παιχνίδια (save slots) με χρονική σήμανση
    - Στο ίδιο αρχείο ή σε διαφορετικά αρχεία εντός καταλόγου
  - Κρυπτογράφηση αρχείου αποθήκευσης
    - xor με σταθερό κλειδί ή με ακολουθία κλειδιών
  - Multiplayer support