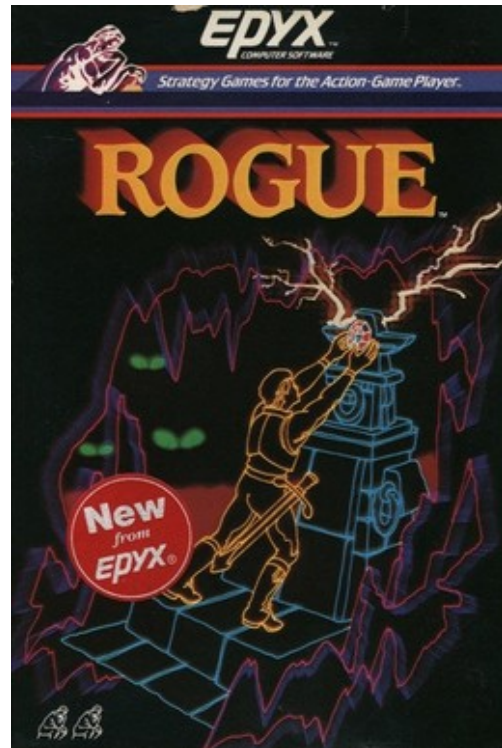
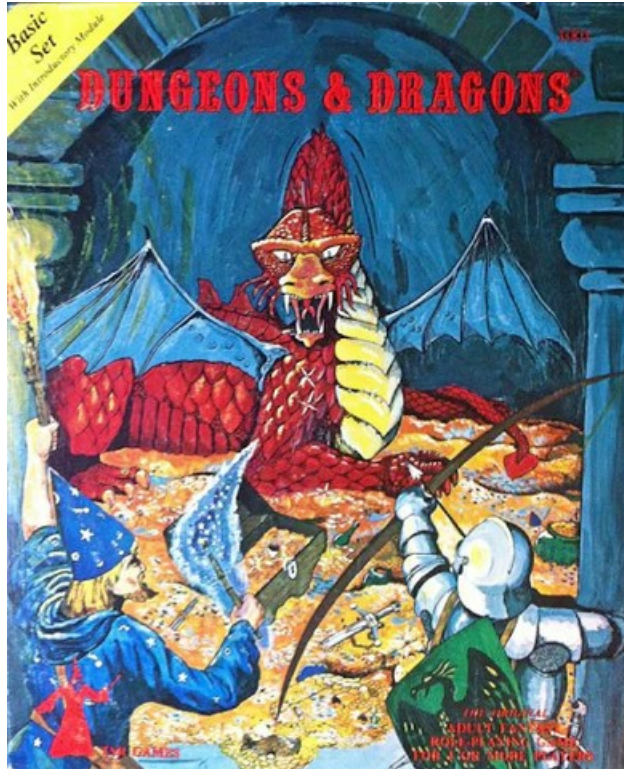


Βιντεοπαιχνίδι κειμένου με python



Εξερεύνηση του μπουντρουμιού

Στο Dungeons and Dragons, ένα απ' τα σημαντικότερα κομμάτια του παιχνιδιού είναι η εξερεύνηση του μπουντρουμιού (dungeon). Εκεί οι ήρωες βρίσκουν ξεχασμένους θησαυρούς, μαγικά αντικείμενα αλλά και τέρατα, παγίδες και κινδύνους. Στα πλαίσια του παιχνιδιού, ο Dungeon master μπορούσε να σχεδιάζει τα μπουντρούμια όπως ήθελε, να αγοράσει μια έτοιμη περιπέτεια ή να αξιοποιήσει κάποιον πίνακα με τυχαίες γεννήτριες ώστε να υπάρχει η αίσθηση της ουδετερότητας και της αγωνίας στο παιχνίδι.

TABLE V.: CHAMBERS AND ROOMS SHAPE AND SIZE (d20)
(Roll for Shape, Size, and Exits; then Contents, Treasure, and how the latter is contained, if applicable.)

Die	Chamber Shape and Area	Room Shape and Area
1-2	Square, 20' x 20'	Square, 10' x 10'
3-4	Square, 20' x 20'	Square, 20' x 20'
5-6	Square, 30' x 30'	Square, 30' x 30'
7-8	Square, 40' x 40'	Square, 40' x 40'
9-10	Rectangular, 20' x 30'	Rectangular, 10' x 20'
11-13	Rectangular, 20' x 30'	Rectangular, 20' x 30'
14-15	Rectangular, 30' x 50'	Rectangular, 20' x 40'
16-17	Rectangular, 40' x 60'	Rectangular, 30' x 40'
18-20	Unusual shape and size — see sub-tables below	

TABLE V. C.: NUMBER OF EXITS (d20)

Die	Room Area	Number of Exits
1-3	up to 600'	1
1-3	over 600'	2
4-6	up to 600'	2
4-6	over 600'	3
7-9	up to 600'	3
7-9	over 600'	4
10-12	up to 1200'	0*
10-12	over 1200'	1
13-15	up to 1600'	0*
13-15	over 1600'	1
16-18	any size	1-4 (d4)
19-20	any size	1 — door in chamber, passage in room

TABLE V. A.: UNUSUAL SHAPE (Roll Separately for Size) (d20)

Die	Shape
1-5	Circular*
6-8	Triangular
9-11	Trapezoidal
12-13	Odd-shaped**
14-15	Oval
16-17	Hexagonal
18-19	Octagonal
20	Cave

* Check once per 10' for secret doors (see TABLE V. D., footnote).

TABLE V. D.: EXIT LOCATION (d20)

Die	Location
1-7	opposite wall
8-12	left wall
13-17	right wall
18-20	same wall

* 1-5 has pool (see TABLE VIII. A. and C. if appropriate), 6-7 has well, 8-10 has shaft, and 11-20 is normal.

TABLE V. F.: CHAMBER OR ROOM CONTENTS (d20)

Die	Contents
1-12	Empty
13-14	Monster only (determine on appropriate table from APPENDIX C: RANDOM MONSTER ENCOUNTERS, Dungeon Encounter Matrix).
15-17	Monster and treasure (see TABLE V.G. below)
18	Special*, or contains stairway up 1 level (1-5), up 2 levels (7-8), down 1 level (9-14), down 2 levels (15-19), or down 3 levels — 2 flights of stairs and a slanting passageway (20).
19	Trick/Trap (see TABLE VII.)
20	Treasure (see TABLE V.G.)

Παραδείγματα γεννητριών δωματίων μπουντρουμιού απ' το Dungeon Master's Guide 1e, 1979

Παιχνίδι προς υλοποίηση

Στα πλαίσια του παιχνιδιού, θα πρέπει να δημιουργήσουμε μέσω τυχαίων γεννητριών το μπουντρούμι που θα εξερευνήσει ο ήρωας. Για να είναι απλούστερο, θα φανταστούμε ότι δεν υπάρχουν διακλαδώσεις σε αυτό το μπουντρούμι και άρα όλα τα δωμάτιά του είναι το ένα μετά το άλλο γραμμικά.

Ο παίχτης κάθε φορά που είναι σε ένα δωμάτιο θα πρέπει να ρωτάται αν θέλει να συνεχίσει στο επόμενο, να επιστρέψει στο προηγούμενο ή αν θέλει να ψάξει το δωμάτιο για να βρει κάποιον κρυμμένο θησαυρό. Αν επιλέξει να προχωρήσει και το δωμάτιο στο οποίο θα μεταβεί δεν το έχει εξερευνήσει ακόμα θα πρέπει τυχαία να καθορίζεται το περιεχόμενό του.



Πώς όμως θα μπορούμε να φτιάξουμε τυχαία τα δωμάτια;

Πολλές φορές στην ρython χρειάζεται να αξιοποιήσουμε έτοιμες συναρτήσεις χωρίς να χρειαστεί να τις φτιάξουμε οι ίδιοι. Για αυτό το λόγο υπάρχουν **τα αρθρώματα (modules)**. Ένα άρθρωμα είναι ουσιαστικά κώδικας ρython στον οποίο έχουμε ορίσει μεταβλητές, συναρτήσεις κτλ τα οποία μπορεί να χρησιμοποιήσει κάποιο άλλο πρόγραμμα ρython αρκεί να έχει εισάγει το άρθρωμα αυτό στον κώδικά του. Αυτό γίνεται με την εντολή **import**.

Για τη δημιουργία τυχαίων αριθμών στο παιχνίδι μας θα αξιοποιήσουμε το module **random**. Στο πρώτο μέρος του φύλλου εργασίας υπάρχουν παραδείγματα χρήσης αρθρωμάτων στα οποία θα πρέπει να προσπαθήσετε να προβλέψετε την απάντηση στις ερωτήσεις και έπειτα να τρέξετε τον κώδικα για να τις επιβεβαιώσετε.

ΜΕΡΟΣ Α

Παράδειγμα 1:

```
import math
print math.pi
print math.sqrt(81)
```

Ερώτηση 1:

Τι πιστεύετε ότι θα εμφανιστεί στην εντολή `math.pi`;

Ερώτηση 2:

Γιατί υπάρχει η τελεία (.) ανάμεσα στο `math` και το `pi` και αντίστοιχα ανάμεσα στο `math` και το `sqrt`;

Ερώτηση 3:

Τι πιστεύετε ότι κάνει η συνάρτηση `sqrt()`; Πώς μπορεί και εκτελείται ενώ την καλούμε χωρίς να την έχουμε ορίσει πρώτα;

Παράδειγμα 2

```
import random
print random.choice(['apple', 'pear', 'banana'])
print random.randint(1,6)
```

Ερώτηση 1:

Τι πιστεύετε ότι κάνει η `random.choice` και η `random.randint(1,6)`; Τι περιμένετε να εμφανιστεί στην οθόνη;

ΜΕΡΟΣ Β

Τώρα που είδαμε πώς λειτουργούν τα modules και μιας και ο ήρωάς μας έχει αγοράσει τον εξοπλισμό του, ήρθε η ώρα να δημιουργήσουμε το dungeon! Αρχικά, θα πρέπει να δημιουργήσουμε συναρτήσεις που θα κάνουν τη δουλειά των ζαριών του παιχνιδιού. Επομένως θα χρειαστεί να φτιάξετε τις συναρτήσεις:

1) d4(), d6(), d10(), d20() που θα πραγματοποιούν τη ρίψη 4πλευρου, 6πλευρου, 10πλευρου και 20πλευρου ζαριού αντίστοιχα, επιστρέφοντας έναν ακέραιο αριθμό στο κατάλληλο εύρος (1-4, 1-6 κ.ο.κ.).

2) Τη συνάρτηση `dungeon_crawling()` η οποία θα ρωτάει τον παίκτη “What would you like to do? 1) Move to the next room, 2) Move to the previous room, 3) Search the current room” και θα διαβάζει την απάντησή του, που θα είναι ένας αριθμός απ’ το 1 έως το 3. Για όση διάρκεια περιφέρεται ο ήρωας στο dungeon θα πρέπει να γνωρίζουμε σε ποιο δωμάτιο βρίσκεται. Μιας και το dungeon θα είναι γραμμικό, θα χρησιμοποιήσουμε μια **λίστα** για να το αναπαραστήσουμε. Κάθε στοιχείο της λίστας θα αντιπροσωπεύει ένα δωμάτιο του dungeon. Τα πρώτα 10 (θέσεις 0-9) θα είναι το πρώτο επίπεδο του dungeon, τα επόμενα 10 (10-19) θα είναι το δεύτερο κ.ο.κ. Κάθε μετακίνηση του ήρωα στο dungeon είναι ένας γύρος. Θα πρέπει μόνιμα να εμφανίζει ένα μήνυμα του τύπου “The hero is in the level *level* in the room *room*. We are on turn *turn*”. Αν ο ήρωας βρίσκεται στο δωμάτιο 0 θα εμφανίζεται και η επιλογή εξόδου απ’ το dungeon. Είναι το μόνο δωμάτιο απ’ το οποίο μπορεί να φύγει απ’ το dungeon.

3) Σε περίπτωση που ο ήρωας μπει δωμάτιο που δεν έχει ξαναεξερευνήσει θα καλείται η συνάρτηση `room_generator()` η οποία θα χρησιμοποιεί το d20 και ανάλογα με το αποτέλεσμα το δωμάτιο θα είναι Empty (1-12), Monster (13-15), Monster and Treasure (16-18), Treasure (18-20). Θα επιστρέφει τη λέξη Empty, Monster, Monster and Treasure ή Treasure ανάλογα και θα αποθηκεύεται στη λίστα.

Καλή επιτυχία!!!