

# A TEXT ADVENTURE GAME

ΕΡΓΑΣΙΑ 3 - ΛΟΓΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.

ΤΖΙΛΙΒΑΚΗΣ ΔΗΜΗΤΡΗΣ : p14179

# ΠΑΡΑΔΟΧΉΣ ΣΤΗΝ ΥΛΟΠΟΙΪΣΗ.

Αρχικά θα πρέπει να τονιστεί ότι η υλοποίηση δεν ακολουθεί το ενδεικτικό παράδειγμα που δόθηκε κάτω από την εκφώνηση, αυτό έγινε επίτηδες, όχι για λόγους ευκολίας, ειδικά από την στιγμή που η υλοποίηση της εργασίας αυτής είναι αρκετά πιο περίπλοκη. Παρόλα αυτά υλοποιούνται ακριβώς όλα τα ζητούμενα της εκφώνησης, προσθέτοντας επιπλέον και πολλά περισσότερα στοιχεία.

Το παράδειγμα στην εκφώνηση δίνει ως ενδεικτικά queries

?- go.

```
>> goto(gas_station). You are in the gas_station. >> goto(car). You can't get there from here. >> open(car_door). >> open(gas_station_door). >> take(soda). You now have a soda. >> goto(car). You are in the car Thanks for getting the newspaper.
```

Από την στιγμή που είναι text adventure game, δεν υπάρχει λόγος ο χρήστης να πληκτρολογεί queries, επομένως η εργασία υλοποιήθηκε με το πρόγραμμα να περιμένει ο χρήστης να πληκτρολογήσει την κατάλληλη εντολή από μια λίστα εντολών αναλόγως σε ποια τοποθεσία βρίσκεται.

Παρόλα αυτά επειδή φαντάζομαι ο λόγος που το παράδειγμα δίνει queries είναι για να χρησιμοποιηθούν λειτουργίες database της swi-prolog, **έχει υλοποιηθεί** με αυτόν τον τρόπο λοιπόν το Inventory, δηλαδή τα αντικείμενα που έχει πάρει ο χρήστης. Τα οποία προσθέτονται δυναμικά.

Επιπλέον ενσωματώθηκε και μία υποτυπώδης ιστορία. Περισσότερες λεπτομέρειες στην συνέχεια όπου ακολουθεί ένα παράδειγμα εκτέλεσης.

Τέλος η εργασία αναπτύχθηκε χρησιμοποιώντας το online compiler της swi-prolog. <http://swish.swi-prolog.org/> Επομένως τα screenshot της εκτέλεσης είναι από τον εκτελεσμένο κωδικά στο συγκεκριμένο περιβάλλον.

# ΕΝΔΕΙΚΤΙΚΉ ΕΚΤΈΛΕΣΗ.

Πληκτρολογούμε το query ?- goto(gas\_station).

Όπου δηλώνουμε ότι θέλουμε να πάμε στο gas\_station αυτομάτως το πρόγραμμα ξέρει ότι μπορείς να πας εκεί μέσω του car. Επιπλέον όπως είναι λογικό ο χρήστης πρέπει να έχει και car\_keys, άρα στο inventory εμφανίζονται αρχικά τα car\_keys.

```
goto(gas_station).
you are at: car
===== ACTIONS =====
available actions:
Type 1 to open_car_door | New Location : gas_station
Type 2 to leave_gas_station | New Location : road
1
=====INVENTORY=====
car_keys,
===== STORY =====
you opened the door of your car door, and walked out. "GOD! this thirst! i need a drink something.",you grumble.
you are at: gas_station
===== ACTIONS =====
available actions:
Type 1 to open_gas_station_door | New Location : gas_station_store
Type 2 to open_car_door | New Location : car
1
=====INVENTORY=====
car_keys,
===== STORY =====
You opened the door and you enter the gas station, in front of you there is the gas station store. Sprakling cool soda is sold, and there is a newspaper stand. There is one problem though, nobody is there, "What happened" you wonder, there were also no traffic on the road, as you
droived there. strange
you are at: gas_station_store
===== ACTIONS =====
available actions:
Type 1 to open_gas_station_door | New Location : gas_station
Type 2 to take_soda | New Location : gas_station_store
Type 3 to take_newspaper | New Location : gas_station_store
2
=====INVENTORY=====
car_keys, soda,
===== STORY =====
"THAT THIRST! OH MY GOD SODA!", you shout, you immediately take a soda and drink it, glu,glu,glu. "This is how a man should live"
you are at: gas_station_store
===== ACTIONS =====
available actions:
Type 1 to open_gas_station_door | New Location : gas_station
Type 2 to take_soda | New Location : gas_station_store
Type 3 to take_newspaper | New Location : gas_station_store
3
=====INVENTORY=====
```

```

=====INVENTORY=====
car_keys, soda, newspaper,
===== STORY =====
You take and check the newspaper. there is some text in it ,but something is completely off, You notice there are gaps in the flow of text in the
articles. Some people names are missing, its like someone wrote an article about someone else and forgot his name. There were also blank pages
like there was something supposed to be written on but never published. "what the hell is going on" you wonder, its like i woke up in a computer
program, and there is a huge bug, causing everything to glitch
you are at: gas_station_store
===== ACTIONS =====
available actions:
Type 1 to open_gas_station_door | New Location : gas_station
Type 2 to take_soda | New Location : gas_station_store
Type 3 to take_newspaper | New Location : gas_station_store
1
=====INVENTORY=====
car_keys, soda, newspaper,
===== STORY =====
You feel something is not right there, so you immediately storm out of the gas_station_store.
you are at: gas_station
===== ACTIONS =====
available actions:
Type 1 to open_gas_station_door | New Location : gas_station_store
Type 2 to open_car_door | New Location : car
2
=====INVENTORY=====
car_keys, soda, newspaper,
===== STORY =====
You have returned to your car.
you are at: car
===== ACTIONS =====
available actions:
Type 1 to open_car_door | New Location : gas_station
Type 2 to leave_gas_station | New Location : road
2
=====INVENTORY=====
car_keys, soda, newspaper,
===== STORY =====
You sit on the driver's chair, and open the radio, listening to a random music program, nothing seems to be wrong. Suddenly the music stops. You
hear a lot of noise coming through the speaker. and a mysterious voice says: "C-C-CODE IS MALFUNCTIONING, THE BUG BUG B-B-B-BUG IS C-
C-COMING". You are terrified, and you immediately turn the car's engine on, leaving this place once and for all. ...To be continue...
- .

```

Κατά την διάρκεια της εκτέλεσης, αναλόγως το μέρος που βρίσκετε ο χρήστης, εμφανίζονται οι αντίστοιχες επιλογές. και ο αριθμός που πρέπει να πληκτρολογήσει ο χρήστης για να την εκτελέσει καθώς και το που θα καταλήξει.

Επιπλέον στο Inventory εμφανίζονται τα αντικείμενα που έχει στην διάθεση του. Το πρόγραμμα είναι σε θέση να καταλάβει πότε πρέπει να προσθέσει κάποιο αντικείμενο αλλά και ποιο αντικείμενο. Θα μπορούσε να είχε υλοποιηθεί και κάποια ενέργεια που να ο χρήστης να πετάει κάποιο αντικείμενο που πήρε, δεν κρίθηκε απαραίτητο όμως

Τέλος εμφανίζει ένα μικρό κείμενο σαν μικρή ιστορία ανάλογα με το που βρίσκετε. Το πρόγραμμα τελειώνει (αποτυγχάνει εάν και μόνο αν ο χρήστης βρίσκετε στο αυτοκίνητο και επιλέξει να φύγει από το gas station. Μπορεί να μεταβεί σε προηγούμενες καταστάσεις σε κάθε μέρος.

# ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ

```
:-dynamic item/1.
state(car,[action('open_car_door', gas_station,
    'you opened the door of your car door, and walked out. "GOD! this thirst!
    i need a drink something.",you grumple.',null),
    action('leave_gas_station', road ,
    'You sit on the driver's chair,
    and open the radio, listening to a random music program, nothing seems to be wrong.
    Suddenly the music stops. You hear a lot of noise coming through the speaker. and a mysterious voice says:
    "C-C-CODE IS MALFUNCTIONING, THE BUG BUG B-B-B-BUG IS C-C-COMING". You are terrified,
    and you immediately turn the car's engine on, leaving this place once and for all. ...To be continue...',null)
    ]),

% actions at gas station.
state(gas_station,[
    action('open_gas_station_door',gas_station_store,
    'You opened the door and you enter the gas station, in front of you there is the gas station store.
    Sprakling cool soda is sold, and there is a newspaper stand. There is one problem though, nobody is there
    "What happened" you wonder, there were also no traffic on the road, as you drived there. strange',null),
    action('open_car_door',car,'You have returned to your car.',null)]),

%actions at gas station store.
state(gas_station_store, [
    action('open_gas_station_door',gas_station,
    'You feel something is not right there, so you immediately
    storm out of the gas_station_store.',null),
    action('take_soda',gas_station_store,
    '"THAT THIRST! OH MY GOD SODA!", you shout, you immediately take a soda and drink it,
    glu,glu,glu. "This is how a man should live" ',soda),
    action('take_newspaper',gas_station_store,
    'You take and check the newspaper.there is some text in it ,but something is completely off,
    You notice there are gaps in the flow of text in the articles. Some people names are missing,
    its like someone wrote an article about someone else and forgot his name. There were also blank
    pages like there was something supposed to be written on but never published.
    "what the hell is going on" you wonder, its like i woke up in a computer program, and there is a hug
    bug, causing everything to glitch',newspaper)
    ]).
```

Ορίζετε ένα δυναμικό γεγονός item (η σύνταξη αυτή είναι απαραίτητη για την swi-prolog)  
**state/2**

Αρχικά ορίστικαν 3 κατηγορήματα state/2 που δηλώνουν μια τοποθεσία καθώς και τις διαθέσιμες ενέργειες Το πρώτο όρισμα δηλώνει την τοποθεσία, και το δεύτερο μια λίστα όπου περιέχει κατηγορήματα action.

## action/4

Το πρώτο όρισμα είναι το όνομα της ενέργειας. Το δεύτερο η τοποθεσία που θα βρεθεί ο χρήστης εάν επιλέξει την ενέργεια αυτή, και τέλος η ιστορία που θα εμφανιστεί αφού ο χρήστης επιλέξει την αντίστοιχη ενέργεια. Το τρίτο ορίζει το αντικείμενο που θα αποκτηθεί. null, αν δεν υπάρχει κάποιο αντικείμενο.

Στο car υπάρχει η εντολή leave\_gas\_station όπου καταλήγει στο road όπου έχει ορίσσει ως τερματική κατάσταση (σε κομμάτι του κώδικα πιο κάτω)

```

%to go to gas_station you will start from car.
stateInit(gas_station, car).

%the predicates will fail when you reach the road.
stateEnd(road).

```

## stateInit\2

Στο κατηγορήμα αυτό ορίζεται η τοποθεσία που πρέπει να αρχίσει για να πάει στο gas station

## stateEnd/1

τερματική κατάσταση.

```

%show all actions of a list with action predicates. write a number for each action based on position
showActions([action(A,B,_,_)|T],Counter):-
    write('Type '),write(Counter),write(' to '),write(A),write(' | New Location : '),write(B),nl,
    C is Counter+1,
    showActions(T,C).
isItem(Item):-
    Item = null.
isItem(Item):-
    Item \= null,
    assert(item(Item)).
s(X):-write(X),write(' '),
inventory():-
    write('=====INVENTORY====='),nl,
    forall(item(X),s(X)).

```

## showActions/2

Ένα κατηγορήμα το οποίο διατρέχει μια λίστα με κατηγορήματα action και εμφανίζει το μήνυμα για την κάθε εντολή. περνώντας και το νούμερο της θέσης της που πρέπει να πληκτρολογήσει αργότερα ο χρήστης

## isItem/1

προσθέτει το αντικείμενο Item στην βάση Δεδομένων ως γεγονός item() εφόσον το Item δεν είναι null, αλλιώς δεν εκτελεί τίποτα και απλώς πετυχαίνει.

## inventory/0

εμφανίζει τα αντικείμενα του χρήστη, ψάχνοντας όλα τα γεγονότα items



```

goto(Place):-
    assert(item(car_keys)),
    stateInit(Place,Through),
    next(Through).

%read action input and go to the next state. This will be done recursively until stateEnd succeed.
next(Place):-
    \+stateEnd(Place),
    write('you are at: '),write(Place),nl,
    state(Place,Actions),
    write('===== ACTIONS ====='),nl,
    write('available actions: '),nl,
    \+showActions(Actions,1),read(UserAction),
    nth1(UserAction,Actions,action(_,NextThrough,Message,Item)),
    isItem(Item),inventory(),nl,
    write('===== STORY ====='),nl,
    write(Message),nl,
    next(NextThrough).

```

### goto/1

Απλώς επιστρέφει την αρχική κατάσταση και έπειτα αρχικοποιεί το next, επιπλέον προσθέτει στην βάση και το item(car\_keys) (πώς, οδηγώ το αυτοκίνητο χωρίς κλειδιά)

### next/1

Το πιο σημαντικό κατηγορημα στο πρόγραμμα καθώς καθορίζει την ροή εκτέλεσης.

Πρώτα από όλα θα πρέπει να ελέγχεται αν η κατάσταση που βρίσκετε ο χρήστης είναι η τελική κατάσταση. Εάν ναι τότε το next/1 θα πρέπει να αποτύχει, συνεπώς αυτό θα γίνει όταν το stateEnd/1 πετύχει. (επομένως χρησιμοποιείται το λογικό not (\+) μπροστά του.

Οι εντολές write δεν προσθέτουν κάποια προγραμματιστική χρηστικότητα και απλώς βρίσκονται εκεί για να εμφανιστούν με μια πιο ευανάγνωστη μορφή κάποια δεδομένα στον χρήστη, όπως οι ενέργειες, το που βρίσκετε και η ιστορία.

Έπειτα αναζητούμε αν αληθεύει το κατηγορημα state(Place,Actions) όπου Place είναι το μέρος που βρίσκεται ο χρήστης και επιστρέφουμε την λίστα με τα action, και την περνάμε στο showActions/1 αρχικοποιώντας τον μετρητή με 1. Πρέπει να σημειωθεί ότι όταν διατρέχουμε μια λίστα χρησιμοποιώντας τον τρόπο που έχει υλοποιηθεί στην showActions, δηλαδή αναδρομικά να περνάμε την ουρά στην κεφαλή της λίστας, όταν θα φτάσει στο τέλος της λίστας θα δοκιμάσει να περάσει την άδεια ουρά στην κεφαλή με αποτέλεσμα να αποτύχει. για αυτό χρησιμοποιείται το λογικό not (\+) ώστε πάντα το κατηγορημα να πετυχαίνει.

διαβάζεται μια επιλογή από τον χρήστη που θα είναι ο αριθμός εμφάνισης του action στην λίστα, και τέλος χρησιμοποιείτε το nth/3 που επιστρέφει ένα στοιχείο μιας λίστας με βάση την θέση, ώστε να πάρουμε την επόμενη τοποθεσία, καθώς και το αντίστοιχο κομμάτι της ιστορίας και το αντικείμενο που θα αποκτήσει ο χρήστης το οποίο περνιέται στο isItem

Καλείτε αναδρομικά το next δίνοντας του ως παράμετρο την επόμενη τοποθεσία.

