

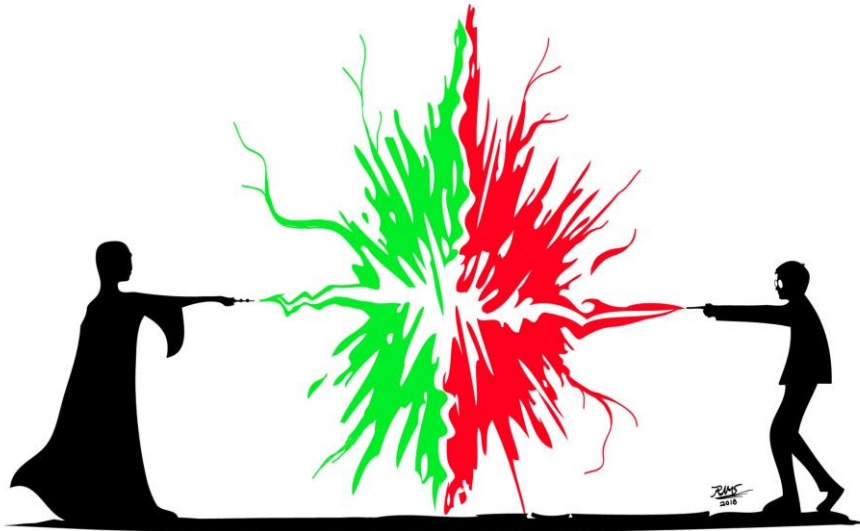
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ – ΗΥ352

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2019
ΔΙΔΑΣΚΩΝ: ΑΝΤΩΝΙΟΣ ΣΑΒΒΙΔΗΣ

ΕΡΓΑΣΙΑ (ομάδες δύο ατόμων)

Ανάθεση: Πέμπτη 12 Δεκεμβρίου 2019

Παράδοση: Πριν την τελική εξέταση (ακριβής ημερομηνία θα ανακοινωθεί όταν θα πλησιάζουμε σε εκείνη την ημερομηνία)



Θέμα – Κατασκευή γλώσσας για προσομοίωση μάχης Harry Potter

Θέμα της εργασίας είναι η κατασκευή μιας γλώσσας ειδικού σκοπού μέσα από την οποία ο προγραμματιστής θα μπορεί να προσομοιώσει μάχες εμπνευσμένες από την ταινία Harry Potter. Η γλώσσα που θα κατασκευάσετε (περιγράφεται παρακάτω) θα γίνεται compiled ως C++ οπότε θα χρειαστείτε ένα ή περισσότερα header files με κατάλληλους ορισμούς ώστε το πρόγραμμά σας που είναι γραμμένο στη γλώσσα σας να αντιστοιχεί σε valid C++ κώδικα και να κάνει compile σωστά. Όπως και στη C++ τα whitespaces θα αγνοούνται. Ένα πρόγραμμα λοιπόν θα **πρέπει** να έχει την εξής μορφή:

```
#include <Hogwarts.h>
```

```
BEGIN_GAME
```

```
...
```

```
END_GAME
```

Στοιχεία της γλώσσας

Ορισμός μάγων

Η δήλωση μάγων γίνεται με τα παρακάτω συντακτικά (προσέξτε ότι στο τέλος κάθε δήλωσης δεν υπάρχει semicolon):

```
CREATE WIZARD {
    NAME: "name",
    HOUSE: "house",
    HP: health_points
}

CREATE WIZARDS [
    WIZARD {
        NAME: "name",
        HOUSE: "house",
        HP: health_points
    },
    WIZARD {
        NAME: "name",
        HOUSE: "house",
        HP: health_points
    },
    ...
]
```

Το *name* είναι το αναγνωριστικό του μάγου ώστε να μπορούμε να αναφερθούμε σε αυτό στη συνέχεια, το *house* δείχνει σε ποιο house ανήκει ο μάγος (*Gryffindor*, *Hufflepuff*, *Ravenclaw*, *Slytherin*) και το *health_points* είναι η ζωή που θα έχει ο μάγος. Παρακάτω φαίνονται 3 παραδείγματα για την δημιουργία μάγων.

Προσοχή: Τα ονόματα των μάγων θα **μπορούν** να περιέχουν και κενό.

Παραδείγματα:

```
CREATE WIZARD {
    NAME: "Harry Potter",
    HOUSE: "Gryffindor",
    HP: 120
}

CREATE WIZARD {
    NAME: "Draco Malfoy",
    HOUSE: "Slytherin",
    HP: 100
}

CREATE WIZARDS [
    WIZARD{
        NAME: "Hermione Granger",
        HOUSE: "Gryffindor",
        HP: 120
    },
    WIZARD{
        NAME: "Luna Lovegood",
        HOUSE: "Ravenclaw",
        HP: 85
    }
]
```

Ορισμός *spell*

Η δήλωση ενός *spell* γίνεται με τα παρακάτω συντακτικά (προσέξτε ότι στο τέλος κάθε δήλωσης δεν υπάρχει semicolon):

```
CREATE SPELL {
    NAME: "spell_name",
    ACTION: START
    ...
END
}

CREATE SPELLS [
    SPELL {
        NAME: "spell_name",
        ACTION: START
        ...
    END
    },
    SPELL {
        NAME: "spell_name",
        ACTION: START
        ...
    END
    },
]
```

Το *spell_name* είναι το αναγνωριστικό του spell ώστε να μπορούμε να αναφερθούμε σε αυτό στη συνέχεια.

Προσοχή: Τα ονόματα των spells **δεν θα μπορούν** να περιέχουν κενό.

● *Action*

Περιέχει τον κώδικα που θα εκτελεστεί όταν κάποιος μάγος χρησιμοποιήσει το συγκεκριμένο spell. Μπορούμε να αναφερθούμε σε αυτόν που κάνει το spell με το keyword **ATTACKER** και σε αυτόν που δέχεται το spell με το keyword **DEFENDER**.

Οι εντολές που μπορούν να γραφτούν μέσα στο Action ενός spell είναι οι εξής:

- Εντολή **DAMAGE DEFENDER/ATTACKER number** κατα την οποία ένας μάγος κάνει damage είτε στον εαυτό του (ATTACKER) είτε στον αντίπαλο μάγο (DEFENDER).
- Εντολή **HEAL DEFENDER/ATTACKER number** κατά την οποία ένας μάγος κάνει heal είτε στον εαυτό του (ATTACKER) είτε στον αντίπαλο μάγο (DEFENDER).
- Εντολή **EQUIP DEFENDER/ATTACKER wand_value** όπου:
 - **EQUIP DEFENDER ---α** κάνει equip στον defender το ραβδί του
 - **EQUIP DEFENDER _** βγάζει το ραβδί του defender
 - **EQUIP ATTACKER ---α** κάνει equip στον attacker το ραβδί του
 - **EQUIP ATTACKER _** βγάζει το ραβδί του attacker
- Εντολή **GET_HP (DEFENDER/ATTACKER)** η οποία επιστρέφει το HP του αντιπάλου (DEFENDER) ή του εαυτού του (ATTACKER).
- Εντολή **GET_HOUSE (DEFENDER/ATTACKER)** η οποία επιστρέφει το house που ανήκει ο αντίπαλος μάγος (DEFENDER) ή ο ίδιος (ATTACKER).
- Εντολή **GET_NAME (DEFENDER/ATTACKER)** η οποία επιστρέφει το όνομα του αντιπάλου (DEFENDER) ή του εαυτού του (ATTACKER).
- Εντολή **HAS_WAND (DEFENDER/ATTACKER)** η οποία επιστρέφει αν ο αντίπαλος μάγος (DEFENDER) ή ο ίδιος (ATTACKER) έχει κάνει equip το ραβδί του.

Σύγκριση, λογικές πράξεις και εντολές ελέγχου μέσα στο Action :

- Συγκρίσεις μεταξύ τιμών (==, !=, >, >=, <, <=)

Παραδείγματα:

- `GET_HOUSE(DEFENDER) == "Gryffindor"`
- `GET_HP(ATTACKER) <= 60`

- Λογικές πράξεις (**AND**, **OR**, **NOT**), όπου το *AND*, *OR* έχει τουλάχιστον 2 στοιχεία ενώ το *NOT* έχει ένα.

Παραδείγματα:

- `AND (GET_HOUSE(ATTACKER) == "Gryffindor", GET_HP(ATTACKER) > 20)`
- `OR (GET_HOUSE(DEFENDER) == "Slytherin",
NOT (GET_HOUSE(ATTACKER) == "Ravenclaw"),
GET_HP(DEFENDER) <= 20
)`
- `NOT (AND (GET_HP(DEFENDER) > 20, GET_HP(DEFENDER) < 70))`

- Εντολή **IF Condition DO Commands ELSE_IF Condition DO Commands ELSE Commands END**.

Παραδείγματα:

- `IF AND (GET_HP(DEFENDER) > 30, GET_HP(DEFENDER) < 70) DO
DAMAGE DEFENDER 20
END`
- `IF GET_HP(DEFENDER) <= 20 DO
DAMAGE DEFENDER 10
ELSE_IF GET_HP(DEFENDER) <= 50 DO
DAMAGE DEFENDER 20
ELSE
DAMAGE DEFENDER 30
END`
- `IF GET_HP(ATTACKER) <= 30 DO
IF GET_HOUSE(ATTACKER) != "Ravenclaw" DO
HEAL ATTACKER 20
ELSE
HEAL ATTACKER 30
END`

- Εντολή επανάληψης **FOR rounds_number ROUNDS DO Commands END** η οποία εκτελεί τα *Commands* για τα επόμενα *rounds_number* rounds.

Παράδειγμα:

- `FOR 5 ROUNDS DO
IF AND (GET_HP(DEFENDER) > 10, GET_HP(DEFENDER) < 80)
DO
DAMAGE DEFENDER 20
END
END`

- Εντολή **AFTER rounds number ROUNDS DO Commands END** η οποία εκτελεί τα *Commands* μετά από *round_number* rounds.

Παράδειγμα:

```
○ AFTER 2 ROUNDS DO
  EQUIP ATTACKER ---α
END
```

- Εντολή **SHOW** *output* η οποία εκτυπώνει τα *output* στην κονσόλα.

Παραδείγματα:

```
○ SHOW GET_HP(ATTACKER)
○ SHOW "Name: " << GET_NAME(DEFENDER) << "House: " <<
  GET_HOUSE(DEFENDER)
```

Παραδείγματα για δημιουργία Spells (μαζί με Actions):

```
CREATE SPELL {
  NAME: "Anapneo",
  ACTION: START
    IF GET_HP(ATTACKER) < 30 DO
      HEAL ATTACKER 25
    ELSE
      HEAL ATTACKER 15
    END
  END
}

CREATE SPELL {
  NAME: "Levicorpus",
  ACTION: START
    EQUIP ATTACKER ---α
    DAMAGE DEFENDER 20
  END
}

CREATE SPELLS[
  SPELL{
    NAME: "Anapneo",
    ACTION: START
      IF GET_HP(ATTACKER) < 30 DO
        HEAL ATTACKER 25
      ELSE
        HEAL ATTACKER 15
      END
    END
  },
  SPELL{
    NAME: "Expulso",
    ACTION: START
      DAMAGE DEFENDER 22
    END
  }
]
1
```

Εκμάθηση spells σε μάγο

Ένας μάγος για να μπορέσει να επιτεθεί χρειάζεται να γνωρίζει spells. Η εκμάθηση ενός spell στον μάγο γίνεται με το παρακάτω συντακτικό (προσέξτε ότι στο τέλος κάθε δήλωσης δεν υπάρχει semicolon):

```
MR/MRS "wizard name" LEARN [  
    SPELL_NAME(spell_name)  
    SPELL_NAME(spell_name)  
    ...  
]
```

Το *wizard name* αντιστοιχεί στο όνομα του μάγου που θέλουμε να μάθει τα spells, το *spell name* αντιστοιχεί στο όνομα του spell.

Προσοχή: Το *spell name* δεν περικλείεται από τα σύμβολα `""`. Επίσης τα στοιχεία της λίστας των SPELL_NAME ΔΕΝ χωρίζονται με κόμμα.

Παραδείγματα:

```
MR "Harry Potter" LEARN [  
    SPELL_NAME(Sectumsemptra)  
    SPELL_NAME(Expelliarmus)  
]
```

```
MRS "Hermione Granger" LEARN [  
    SPELL_NAME(WinguardiumLeviosa)  
    SPELL_NAME(ConfundusCharm)  
    SPELL_NAME(Stupefy)  
    SPELL_NAME(Obliviate)  
]
```

Houses

Κάθε μάγος ανήκει σε κάποιο συγκεκριμένο house. Λόγω αντιπαλότητας ανάμεσα στα houses οι μάγοι έχουν αναπτύξει κάποιες ειδικές δυνάμεις για να μπορούν να ανταποκριθούν στη μάχη.

Πιο συγκεκριμένα:

- Οι μάγοι που ανήκουν στο house **Gryffindor** δέχονται 30% **λιγότερο** damage από μάγους του **Slytherin** και 20% **λιγότερο** damage από τα **άλλα houses**.
- Οι μάγοι που ανήκουν στο house **Slytherin** κάνουν 20% **περισσότερο** damage σε μάγους του **Gryffindor** και 15% **περισσότερο** damage σε μάγους των **άλλων houses**.
- Οι μάγοι που ανήκουν στο house **Hufflepuff** κάνουν 7% **περισσότερο** damage σε όλους και δέχονται 7% **λιγότερο** damage από όλους τους μάγους.

- Οι μάγοι που ανήκουν στο house **Ravenclaw** στα rounds που είναι **περιττοί** αριθμοί κάνουν 7% **περισσότερο** damage σε όλους και στα rounds που είναι **άρτιοι** αριθμοί δέχονται **αυτόματα** heal 5% του max health τους κατά την έναρξη του round.

Η Μάχη

Η μάχη μεταξύ δύο μάγων γίνεται από την **κονσόλα** και αναπαρίσταται από την εντολή **DUEL**

Κατά την έναρξη της μάχης:

- Θα πρέπει να εμφανίζεται η λίστα με τους μάγους που έχουν δηλωθεί
- Οι 2 παίκτες επιλέγουν από την λίστα τον μάγο που θέλουν να χειρίζονται
- Προσοχή, μπορούν και οι δύο παίκτες να επιλέξουν τον **ίδιο** μάγο
- Κατά την έναρξη της μάχης και οι δύο μάγοι κρατάνε ραβδί

Κατά τη διάρκεια ενός γύρου:

- Θα πρέπει να εμφανίζεται η λίστα με τα διαθέσιμα spells του μάγου που είναι η σειρά του από την οποία θα επιλέγει το spell του. Αντίστοιχα, και για τον δεύτερο παίκτη.
Διαθέσιμα είναι όλα τα spells που έχει μάθει ο μάγος (με την εντολή LEARN) κατά τη διάρκεια του παιχνιδιού.
Σε περίπτωση που κάποιος μάγος δε κρατάει ραβδί, δε χρειάζεται το παραπάνω αλλά πρέπει να τυπώνεται κατάλληλο μήνυμα (δες Screenshot).
- Αφού ένας μάγος εκτελέσει το spell του θα πρέπει να εκτυπώνεται στην κονσόλα η κατάσταση και των 2 μάγων. π.χ.

Name: *Harry Potter*

HP: *100*

House: *Gryffindor*

Η μάχη τερματίζεται με την ήττα ενός από τους δύο μάγους (έχει χαθεί όλο το HP), που αμέσως αναδεικνύει το αντίπαλο μάγο σε νικητή.

Ακολουθεί **ολοκληρωμένο παράδειγμα** μάχης, Screenshot με το **output** που **ΠΡΕΠΕΙ** να ακολουθήσετε καθώς και μερικά **hints**.

Ολοκληρωμένο Παράδειγμα:

```
#include "Hogwarts.h"
```



```

BEGIN_GAME

CREATE SPELL {
    NAME: "Expelliarmus",
    ACTION: START
        //Αφοπλίζει τον αντίπαλο μάγο (DEFENDER)
        EQUIP DEFENDER
        // Μετά απο 2 γύρους επιστρέφει το ραβδί στον αντίπαλο
        (DEFENDER)
        AFTER 2 ROUNDS DO
            EQUIP DEFENDER ---α
        END
    END
}

CREATE SPELL {
    NAME: "Sectumsempra",
    ACTION: START
        // Για 5 γύρους κάνει 8 damage στον αντίπαλο (DEFENDER)
        FOR 5 ROUNDS DO
            DAMAGE DEFENDER 8
        END
    END
}

CREATE SPELL {
    NAME: "Expulso",
    ACTION: START
        // Κάνει 22 damage στον αντίπαλο (DEFENDER)
        DAMAGE DEFENDER 22
    END
}

CREATE SPELL{
    NAME: "Anapneo",
    ACTION: START
        // Κάνει heal στον εαυτό του (ATTACKER)
        HEAL ATTACKER 30
    END
}

CREATE WIZARD {
    NAME: "Harry Potter",
    HOUSE: "Gryffindor",
    HP: 100
}

CREATE WIZARD {
    NAME: "Draco Malfoy",
    HOUSE: "Slytherin",
    HP: 90
}

MR "Harry Potter" LEARN [
    SPELL_NAME(Expelliarmus)
    SPELL_NAME(Sectumsempra)
    SPELL_NAME(Expulso)

```

```
        SPELL_NAME (Anapneo)
    ]

    MR "Draco Malfoy" LEARN [
        SPELL_NAME (Expelliarmus)
        SPELL_NAME (Expulso)
    ]

    DUEL

END_GAME
```

Screenshot από παράδειγμα μάχης:

```
-----HARRY POTTER THE GAME-----

Player1 select wizard:
-----
DracoMalfoy
HarryPotter
-----
DracoMalfoy

Player2 select wizard:
-----
DracoMalfoy
HarryPotter
-----
HarryPotter

~~~~~
Round 1
~~~~~
DracoMalfoy(Player1) select spell:
-----
Expelliarmus
Expulso
-----
Expulso

#####
Name: DracoMalfoy
HP: 90
Wand equipped
#####

#####
Name: HarryPotter
HP: 81
Wand equipped
#####
```

HarryPotter(Player2) select spell:

Anapneo
Expelliarmus
Expulso
Sectumsempra

Expelliarmus

#####

Name: DracoMalfoy

HP: 90

Wand not equipped

#####

#####

Name: HarryPotter

HP: 81

Wand equipped

#####

~~~~~

Round 2

~~~~~

DracoMalfoy(Player1) has not a wand so he can't cast a spell.

HarryPotter(Player2) select spell:

Anapneo
Expelliarmus
Expulso
Sectumsempra

Hints

Για να μετατρέψετε το συντακτικό της γλώσσας σε *valid C++* χρησιμοποιήστε:

- ✧ Τη δυνατότητα για operator overloading που σας προσφέρει η C++, δίνοντας μεγάλη προσοχή στην προτεραιότητα των operators.
 - `operator[]`
 - Για την εκμάθηση ξορκιών σε μάγους.
 - `operator,`
 - Για να μαζεύετε εκφράσεις που έχουν κόμμα ανάμεσα τους.
- ✧ Δημιουργία προσωρινών στιγμιότυπων ως επιστρεφόμενα αποτελέσματα, αλλά και ως βοηθητικά στιγμιότυπα σε εκφράσεις.
- ✧ Αρκετά τον preprocessor αφού λέξεις κλειδιά της γλώσσας όπως *CREATE*, *WIZARD*, *DAMAGE*, κτλ. θα είναι macros που θα κρύβουν μετατροπές σε strings, κλήσεις συναρτήσεων, κάποιους operators ή και βοηθητικά προσωρινά στιγμιότυπα.
- ✧ Τους initializers της C++11 –{ }– για κλήση constructors και αρχικοποίηση μεταβλητών.