



Συστήματα Γνώσης

Πρακτικό Κομμάτι Μαθήματος
Πρόγραμμα Πέτρα – Ψαλίδι - Χαρτί

Νίκος Βασιλειάδης, Αναπλ. Καθηγητής
Τμήμα Πληροφορικής



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ



Πρόγραμμα Πέτρα – Ψαλίδι - Χαρτί

Πέτρα – Ψαλίδι - Χαρτί

- Παιδικό παιχνίδι
 - Παίζουν δύο παίκτες (άνθρωπος vs. CLIPS) οι οποίοι προτείνουν (εκφωνούν) μία από τις λέξεις «πέτρα – ψαλίδι - χαρτί»
- Βαθμολογία: Κάθε φορά που ο ένας παίκτης εκφωνεί μία λέξη που «κερδίζει» την άλλη παίρνει ένα βαθμό
 - Η Πέτρα στομώνει το Ψαλίδι.
 - Το Χαρτί τυλίγει την Πέτρα.
 - Το Ψαλίδι κόβει το Χαρτί.
 - Όταν οι 2 παίκτες εκφωνούν την ίδια λέξη είναι ισοπαλία (δεν παίρνει βαθμό κανένας)



Πρότυπα Γεγονότων

```
(deftemplate win-totals
  (slot human (type INTEGER) (default 0))
  (slot computer (type INTEGER) (default 0))
  (slot ties (type INTEGER) (default 0)))
```

```
(deftemplate results
  (slot winner (type SYMBOL)
    (allowed-symbols rock paper scissors))
  (slot loser (type SYMBOL)
    (allowed-symbols rock paper scissors))
  (slot why (type STRING)))
```



Αρχικά Γεγονότα

```
(defacts information
  (results (winner rock) (loser scissors)
    (why "Rock smashes scissors"))
  (results (winner scissors) (loser paper)
    (why "Scissors cut paper"))
  (results (winner paper) (loser rock)
    (why "Paper covers rock"))
  (valid-answer rock r rock)
  (valid-answer paper p paper)
  (valid-answer scissors s scissors))
```



Αρχικός Κανόνας

```
(defrule startup
```

```
=>
```

```
(printout t "Lets play a game!" crlf crlf)
```

```
(printout t "You choose rock, paper, or scissors,"  
crlf)
```

```
(printout t "and I'll do the same." crlf crlf)
```

```
(printout t "Rock smashes scissors!" crlf)
```

```
(printout t "Paper covers rock!" crlf)
```

```
(printout t "Scissors cut paper!" crlf crlf)
```

```
(set-strategy random)
```

```
(assert (win-totals))
```

```
(assert (get-human-move)) )
```



Κανόνες «Κίνησης» Ανθρώπου

- Εισαγωγή «κίνησης»

```
(defrule get-human-move
```

```
  (get-human-move)
```

```
=>
```

```
  (printout t "Rock (R) , Paper (P) ,  
or Scissors (S) ? ")
```

```
  (assert (human-choice (read))))
```



Κανόνες «Κίνησης» Ανθρώπου

- Σωστή εισαγωγή «κίνησης»

```
(defrule good-human-move
  ?f1 <- (human-choice ?choice)
  (valid-answer ?answer $? =(lowercase ?choice) $?)
  ?f2 <- (get-human-move)
=>
  (retract ?f1 ?f2)
  (assert (human-choice ?answer))
  (assert (get-computer-move)))
```



Κανόνες «Κίνησης» Ανθρώπου

- Λανθασμένη εισαγωγή «κίνησης»

```
(defrule bad-human-move
  ?f1 <- (human-choice ?choice)
  (not (valid-answer ?answer $? =(lowcase
?choice) $?))
  ?f2 <- (get-human-move)
=>
  (retract ?f1 ?f2)
  (assert (get-human-move)))
```



Κανόνες «Κίνησης» Υπολογιστή

- Επιλογή «Πέτρας» (random strategy)

```
(defrule computer-picks-rock
  ?f1 <- (get-computer-move)
  =>
  (printout t "Computer chooses
  rock" crlf)
  (retract ?f1)
  (assert (computer-choice rock))
  (assert (determine-results)))
```



Κανόνες «Κίνησης» Υπολογιστή

```
(defrule computer-picks-paper
  ?f1 <- (get-computer-move)
  =>
  (printout t "Computer chooses paper" crlf)
  (retract ?f1)
  (assert (computer-choice paper))
  (assert (determine-results)))

(defrule computer-picks-scissors
  ?f1 <- (get-computer-move)
  =>
  (printout t "Computer chooses scissors"
  crlf)
  (retract ?f1)
  (assert (computer-choice scissors))
  (assert (determine-results)))
```



Κανόνες Ανάδειξης Νικητή

```
(defrule computer-wins
  ?f1 <- (determine-results)
  ?f2 <- (computer-choice ?cc)
  ?f3 <- (human-choice ?hc)
  (results (winner ?cc) (loser ?hc)
           (why ?explanation))
  ?w <- (win-totals (computer ?cw))
=>
  (retract ?f1 ?f2 ?f3)
  (modify ?w (computer (+ ?cw 1)))
  (format t "%s%n" ?explanation)
  (printout t "Computer wins!" t)
  (assert (determine-play-again)))
```



Κανόνες Ανάδειξης Νικητή

```
(defrule human-wins
  ?f1 <- (determine-results)
  ?f2 <- (computer-choice ?cc)
  ?f3 <- (human-choice ?hc)
  (results (winner ?hc) (loser ?cc)
           (why ?explanation))
  ?w <- (win-totals (human ?hw))
=>
  (retract ?f1 ?f2 ?f3)
  (modify ?w (human (+ ?hw 1)))
  (format t "%s%n" ?explanation)
  (printout t "You win!" t)
  (assert (determine-play-again)))
```



Κανόνες Ανάδειξης Νικητή

```
(defrule tie
  ?f1 <- (determine-results)
  ?f2 <- (computer-choice ?cc)
  ?f3 <- (human-choice ?cc)
  ?w <- (win-totals (ties ?nt))
=>
  (retract ?f1 ?f2 ?f3)
  (modify ?w (ties (+ ?nt 1)))
  (printout t "We tie." t)
  (assert (determine-play-again)))
```



Κανόνας Επανάληψης Παιχνιδιού

```
(defrule play-again
  ?f1 <- (determine-play-again)
  (win-totals (computer ?ct) (human ?ht)
             (ties ?tt))
=>
  (retract ?f1)
  (if (yes-or-no-p "Play again? ")
      then
        (assert (get-human-move))
      else
        (printout t crlf "You won " ?ht " game(s)." crlf)
        (printout t "Computer won " ?ct " game(s)." crlf)
        (printout t "We tied " ?tt " game(s)." crlf crlf)
        (halt)))
```



Συνάρτηση Ερώτησης

```
(deffunction yes-or-no-p (?question)
  (bind ?x bogus)
  (while (and
          (neq ?x yes) (neq ?x y)
          (neq ?x no) (neq ?x n))
    (format t "%s(Yes or No) " ?question)
    (bind ?x (lowercase (sym-cat (read))))))
  (if (or (eq ?x yes) (eq ?x y))
      then
        TRUE
      else
        FALSE) )
```





Τέλος Ενότητας

Επεξεργασία: Εμμανουήλ Ρήγας

Θεσσαλονίκη, 17/3/2014



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ