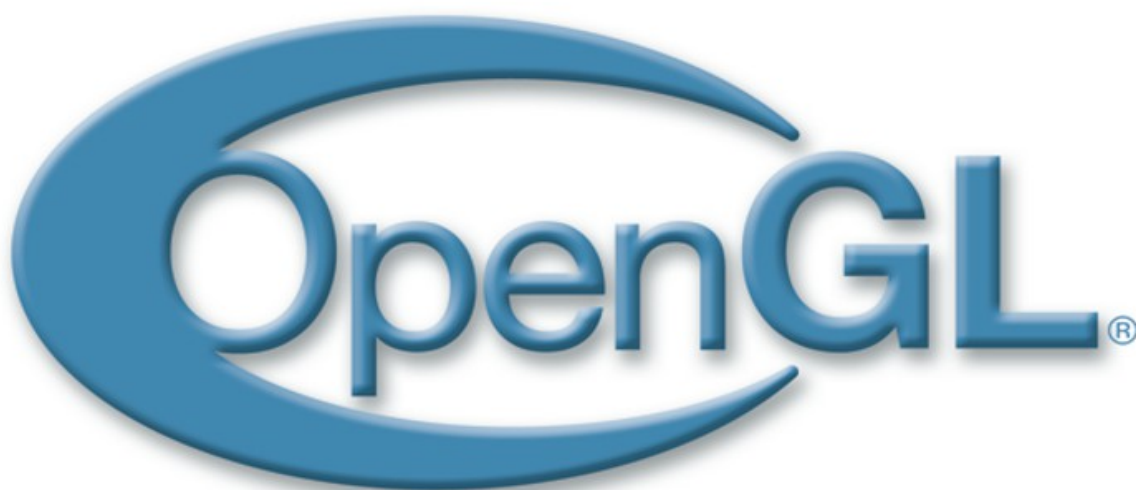


**Γραφικά Υπολογιστών  
και  
Συστήματα Αλληλεπίδρασης**



**Τζιβάρας Βασίλης & Τάσιος Βασίλης**

Πανεπιστήμιο Ιωαννίνων  
Τμήμα Πληροφορικής

# Περιεχόμενα

1. Οργάνωση Αρχείων
2. Γραφική Διεπαφή
3. Γραφική Σχεδίαση
4. Αλγοριθμική Σχεδίαση

## 1. Οργάνωση Αρχείων

Ο κώδικας μας αποτελείται απο το αρχείο main.cpp, βιβλιοθήκες και ένα Makefile για την μεταγλώττιση.

<b>main.cpp</b>	Περιέχει τον “σκελετό” του προγράμματος, τις βιβλιοθήκες και την διεπαφή με το πληκτρολόγιο και την οθόνη.
<b>display.h</b>	Περιέχει όλες τις συναρτήσεις που χρειάζονται για την προβολή των γραφικών.
<b>moves.h</b>	Περιέχει όλες τις συναρτήσεις που χρειάζονται για την μετακίνηση του αντικειμένου.
<b>loadImages.h</b>	Φόρτωση εικόνων
<b>object.h</b>	Περιέχει την κλάση και τις υλοποιήσεις των μεθόδων της
<b>actions.h</b>	Περιέχει λειτουργίες για το παιχνίδι όπως η restartGame().
<b>global.h</b>	Περιέχει τις global μεταβλητές

Η εκτέλεση του κώδικα σε Linux συστήματα γίνεται μεταγλωττίζοντας το Makefile με

\$ make all

και τρέχοντας το εκτελέσιμο

\$ ./vraxapsa

για έναν παίχτη και

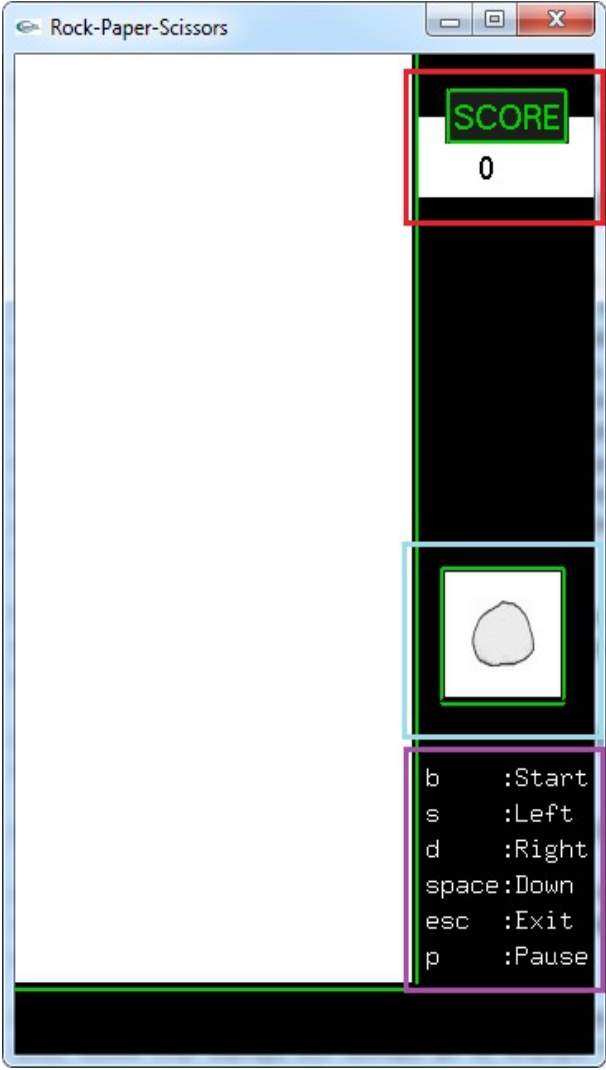
\$ ./vraxapsa 2

για δυο παίκτες.

## 2. Γραφική Διεπαφή

Στο παιχνίδι έχουν υλοποιηθεί τα ερωτήματα I, ii, iii, iv, v, vi και απο τα Bonus το i, ii, iii. Επίσης έχουν προστεθεί επιπλέον λειτουργίες που έχουν και παρόμοια παιχνίδια. Αναλυτικά το παιχνίδι υποστηρίζει επιπλέον τις δυνατότητες

- παύσης με το πλήκτρο <p>
- γρήγορη κίνηση του τρέχων αντικειμένου με το πλήκτρο <space>
- προβολής του επόμενου αντικειμένου που θα έρθει
- προσθήκη ήχου (classic tetris) στο παιχνίδι και ηχητικά εφέ όταν τρώγονται πάνω από 5 και πάνω από 10 αντικείμενα στη σειρά.

	<p>Για το ερώτημα ii το Score αποφασίσαμε να το εμφανίσουμε σε ένα δεξί πάνελ που δημιουργήσαμε (κόκκινο πλαίσιο της φωτογραφίας). Θεωρήσαμε ότι είναι πιο βολικό και λειτουργικό μαζί με τις υπόλοιπες πληροφορίες.</p> <p>Η προβολή του επόμενου αντικειμένου που θα πέσει εμφανίζεται στο γαλάζιο πλαίσιο της φωτογραφίας.</p> <p>Στο μωβ πλαίσιο της φωτογραφίας έχουμε τα πλήκτρα χειρισμού του παιχνιδιού μαζί με τις δικές μας προσθήκες.</p>
--	--

Τέλος, για το ερώτημα iv το “Game over” αποφασίσαμε να το εμφανίσουμε στο κέντρο του παραθύρου.

### 3. Γραφική Σχεδίαση

- Το παράθυρό μας έχει σχεδιαστεί με διαστάσεις 350x560. Το παιχνίδι εξελίσσεται με  $x(0, 240)$  και  $y(40, 560)$ , δηλαδή στο άσπρο πλαίσιο της παραπάνω φωτογραφίας. Το ορθοκανονικό μας σύστημα ορίζεται μέσα στην συνάρτηση `init()` του αρχείου `main.cpp` με  $(0,0)$  την κάτω αριστερή γωνία του παραθύρου.
- Φόρτωση εικόνων  
Η φόρτωση εικόνων υλοποιείται στη βιβλιοθήκη `loadImages.h` και για την αναπαράσταση χρησιμοποιήθηκε η `glDrawPixels()` παίρνοντας ως όρισμα έναν από τους global πίνακες. Για τα κόκκινα, μπλε και τις βόμβες έχουμε προσθέσει αντίστοιχες εικόνες.
- Μετακίνηση εικόνων  
Η μετακίνηση των εικόνων γίνεται με την συνάρτηση `glRasterPos2i()` αυξομειώνοντας τις συντεταγμένες του τρέχον αντικειμένου. Η συνεχής μετακίνηση του αντικειμένου προς τα κάτω γίνεται χρησιμοποιώντας την `glutTimerFunc()` καλώντας έτσι επαναληπτικά την `moveDown()`.
- Προβολή των κειμένων/score  
Η προβολή των κειμένων και του score γίνεται με την `glutBitmapCharacter()` και το μετακινούμε στην θέση που θέλουμε με την `glRasterPos2i()`.

#### 4. Αλγοριθμική Σχεδίαση

- Η βασική δομή στην οποία στηρίζεται το παιχνίδι είναι η κλάση `object` που περιλαμβάνει πληροφορίες για το αντικείμενο όπως αν είναι ενεργό, τον τύπο του, την εικόνα σε έναν δισδιάστατο πίνακα και τις συντεταγμένες του. Επιπλέον περιέχει μεθόδους για τον καθορισμό των παραπάνω πεδίων..
- Έχουμε δημιουργήσει έναν πίνακα τύπου της παραπάνω κλάσης ο οποίος αποθηκεύει τα αντικείμενα που έπεσαν και έμειναν. Ο πίνακας αυτός διατρέχεται όταν χρειάζεται να προβάσουμε τα αντικείμενα που έχουν μείνει στην οθόνη μετακινείται ένα αντικείμενο ώστε να δούμε ποιες θέσεις είναι ήδη κατειλημμένες εκτελείται η λειτουργία `eat()` για να φαγωθούν τα γειτονικά αντικείμενα
- Επίσης χρησιμοποιείται μια λίστα την οποία καταχωρούνται προσωρινά οι γείτονες του κάθε αντικειμένου που τρώγεται. Έπειτα ελέγχουμε έναν έναν του βγάζουμε απο την λίστα και ελέγχουμε αν τρώγονται η όχι. Αν τρώγονται τότε στον πίνακα μας γίνονται ανενεργοί και στην λίστα προστίθενται οι νέοι γείτονες.