

Omada:Argirokastritis Polidoros 2196

Το πρόγραμμα τρέχει σωστά, με μια μικρή καθυστέρηση όταν υπάρχουν τριάδες ίδιου τύπου. Αντιμετώπισα πρόβλημα με την λειτουργία reshape του παραθύρου και δεν κατάφερα να το λύσω. Ακόμα, όταν το έτρεχα στο δικό μου μηχάνημα έτρεχε μια χαρά, ενώ όταν το έτρεχα σε μηχάνημα της σχολής και πατούσα "Esc", έβγαζε core dumped.

Ο κώδικας ξεκινάει με τις δηλώσεις κάποιων global μεταβλητών:

- winWidth και winHeight , οι οποίες είναι οι διαστάσεις για το μήκος και το πλάτος του παραθύρου του παιχνιδιού αντίστοιχα.
- gfPosX1, gfPosX2, gfPosY1, gfPosY2 και gfDelta, οι οποίες είναι συντεταγμένες στους άξονες x,y για το παράθυρο του παιχνιδιού και χρειάζονται για να τοποθετηθούν σωστά τα τετράγωνα στο παράθυρο.
- Counter είναι μια βοηθητική μεταβλητή η οποία ελέγχει για πόσες τριάδες ίδιου τύπου θα εκτελείται μόνο του το πρόγραμμα(αν υπάρχει αυτός ο αριθμός των τριάδων). Μόλις μηδενιστεί η μεταβλητή, μπορεί να παίξει ο χρήστης. (Στο πρόγραμμα μου ισχύει Counter=5)
- Count είναι μια βοηθητική μεταβλητή για να εμφανίζεται το μήνυμα SCORE:0 μόνο στην αρχή του παιχνιδιού.
- Mx και my, οι οποίες είναι μεταβλητές για να κρατάμε τις συντεταγμένες x,y ενός πατήματος του ποντικιού στο παράθυρο.
- Mtimes,th1,th2 και th είναι μεταβλητές οι οποίες βοηθούν στο να βρεθούν τα τετράγωνα τα οποία επιλέξαμε με το πάτημα του αριστερού κουμπιού του ποντικιού.
- totalMoves και moves, στις οποίες αποθηκεύεται ο αριθμός κινήσεων που μπορεί να εκτελέσει ο παίχτης.
- Repeat η οποία είναι μια μεταβλητή που χρησιμοποιείται στην μέθοδο threeOfAKind() και βοηθάει στο να βρίσκω τις τριάδες ίδιου τύπου που δημιουργούνται στο παιχνίδι.

- `textureR, textureS, textureP` είναι πίνακες `float` τιμών, τους οποίους γεμίζω και στέλνω στην συνάρτηση `DrawPixels(...)` για να εμφανιστούν στο παράθυρο οι εικόνες «ψαλίδι», «βράχος» και «Χαρτί».
- `arrayOfQuad`, ο οποίος είναι πίνακας `int` τιμών και κρατάω τους τύπους των τετραγώνων. Βοηθάει σχεδόν σε όλες τις μεθόδους του προγράμματος.
- `Root, current, first, second, root2` και `current2` είναι αντικείμενα της τάξης `quad` και βοηθούν στην απεικόνιση των τετραγώνων στο παράθυρο.
- `Score`, η οποία είναι μια μεταβλητή για την διατήρηση του σκορ του παίχτη.

Στη συνέχεια του κώδικα ακολουθούν οι μέθοδοι. Θα αναλυθούν με λίγα λόγια όλες οι μέθοδοι, ξεκινώντας από την `main()`.

Η `main()` εμφανίζει ένα μήνυμα στον τερματικό για να επιλέξει ο παίχτης σε πόσες κινήσεις τελειώνει το παιχνίδι. Αφού εισάγει μια τιμή ο παίχτης και αποθηκευτεί στην μεταβλητή `totalMoves`, δεσμεύω χώρο στην μνήμη για τα αντικείμενα που θα χρειαστούν και δημιουργώ την αρχή της λίστας με τα αντικείμενα τύπου `quad`. Μετά δημιουργείται το παράθυρο του παιχνιδιού, ορίζοντας αρχικά την θέση του και τις συντεταγμένες του στους άξονες `x, y` και καλώντας τελικά την μέθοδο `initialize()`. Τέλος, καλώ τις μεθόδους που χρειάζονται για την εκτέλεση του παιχνιδιού και πριν τελειώσει το πρόγραμμα αποδεσμεύω τον χώρο που είχα δεσμεύσει για τα αντικείμενα.

Η `keyboardFunc()` είναι η μέθοδος που βοηθά στην αλληλεπίδραση του πληκτρολογίου με το παράθυρο του προγράμματος. Όταν ο παίχτης πατάει το κουμπί “b”, ξεκινά το παιχνίδι ή μπορεί, αν έχει ξεκινήσει ήδη το παιχνίδι, να κάνει `restart` του παιχνιδιού, ενώ όταν πατάει το κουμπί “Esc”, βγαίνει από το παιχνίδι, σταματάει να τρέχει το πρόγραμμα και κλείνει το παράθυρο.

Η `display()` καλεί τη συνάρτηση `grid()`, η οποία με την σειρά της δημιουργεί το πλέγμα και το εμφανίζει καθ’ όλη την διάρκεια του

παιχνιδιού και εμφανίζει τα πρώτα μηνύματα στο παράθυρο. Τα μηνύματα εμφανίζονται με την μέθοδο `drawText()`.

Η `image()` είναι υπεύθυνη για την δημιουργία των εικόνων. Διαβάζει τα αρχεία `rgm` και αποθηκεύει τις τιμές που υπάρχουν στα αρχεία στους κατάλληλους πίνακες ώστε να μπορούν να χρησιμοποιήσουν τους πίνακες οι μέθοδοι `drawPixels()`.

Η μέθοδος `display2()` καλεί τις μεθόδους `image()` και `start()`. Η `start()` αρχικά δημιουργεί τα τετράγωνα και τα τοποθετεί τυχαία. Τέλος καλεί την μέθοδο `threeOfAKind()`, η οποία βρίσκει, τις κάθετες πρώτα και στη συνέχεια τις οριζόντιες, τριάδες. Αφού τις βρει, καταστρέφει την τριάδα και σε απόσταση 1,2 και 3 τα τετράγωνα που πρέπει να καταστραφούν και παράλληλα υπολογίζει και το σκορ του παίχτη. Η μέθοδος `threeOfAKind()` βρίσκει τον αριθμό των τριάδων που έχω ορίσει, τυχαία, στην μεταβλητή `counter`. Πιο συγκεκριμένα, η `while()` που υπάρχει μέσα στη μέθοδο ελέγχει τον αριθμό των τριάδων που θα βρεθούν και κάθε φορά που βρίσκει μια τριάδα και γίνεται η «έκρηξη», μειώνεται η μεταβλητή `counter`. Η `while()` εκτελείται μέχρις ότου η μεταβλητή `counter` μηδενιστεί ή δεν υπάρχει άλλη τριάδα.

Η `redisplay()` δημιουργεί τα νέα τετράγωνα και ελέγχει αν τα νέα τετράγωνα που δημιουργούνται γειτνιάζουν με τετράγωνα τα οποία «τρώνε». Αν βρει τέτοια τετράγωνα, καταστρέφει όλα τα γειτονικά τετράγωνα που «τρώει» το νέο. Τέλος, η μέθοδος εμφανίζει το σκορ του παίχτη στο κάτω δεξιά μέρος του παραθύρου.

Η μέθοδος `fallOfQuad()` είναι υπεύθυνη για να «πέφτουν» τα τετράγωνα στις κενές θέσεις, οι οποίες δημιουργήθηκαν από τις εκρήξεις.

Τέλος, οι μέθοδοι `move()` και `exchange()` είναι υπεύθυνες για την μετακίνηση των τετραγώνων που επέλεξε ο παίχτης, χρησιμοποιώντας για τον σκοπό αυτό και τις μεθόδους `threeOfAKind()`, `redisplay()` και `mousebutton()`. Η `mousebutton()` βρίσκει τις συντεταγμένες του πατήματος του αριστερού κουμπιού πάνω στο παράθυρο του παιχνιδιού. Μια μετακίνηση

ενός τετραγώνου μπορεί να γίνει μόνο αν ο παίχτης επιλέξει μια γειτονική θέση του τετραγώνου που επιθυμεί να μετακινήσει. Διαφορετικά, αν επιλέξει δύο φορές το ίδιο τετράγωνο δεν εκτελείται κίνηση, δεν μειώνεται ο αριθμός των διαθέσιμων κινήσεων και στη συνέχεια μπορεί να εκτελέσει μια διαφορετική κίνηση. Επίσης, αν ο παίχτης επιλέξει μη γειτονικό τετράγωνο του τετραγώνου που θέλει να μετακινήσει, και σε αυτή την περίπτωση δεν εκτελείται κίνηση, μέχρις ότου να επιλέξει ένα γειτονικό του τετράγωνο.