Σχέδιο Δραστηριότητας: Παιχνίδι: Πέτρα – Ψαλίδι – Χαρτί

Αντί Εισαγωγής

Ο σκοπός αυτής της δραστηριότητας είναι η υλοποίηση του γνωστού παιχνιδιού Πέτρα – Ψαλίδι – Χαρτί το οποίο παίζουν δυο παίκτες, εσείς και ο υπολογιστής. Σε περίπτωση που δεν έχετε ξαναπαίξει θυμίζουμε ότι οι δυο παίκτες επιλέγουν ένα αντικείμενο και η Πέτρα κερδίζει το Ψαλίδι, το Ψαλίδι κερδίζει το Χαρτί το οποίο κερδίζει την Πέτρα. Θα συμφωνήσουμε ότι ο χρήστης (δηλαδή εσείς) θα δίνει τους αγγλικούς χαρακτήρες R, P, S αντί για Πέτρα, Χαρτί και Ψαλίδι αντίστοιχα.

Το αναγκαίο κακό

Δυστυχώς θα πρέπει αρχικά να γράψουμε κάποιες δηλώσεις τις οποίες δεν είναι απαραίτητο να καταλάβουμε ακριβώς τι σημαίνουν. Αρκεί να ξέρετε ότι με αυτές τις δηλώσεις ενημερώνετε τον μεταγλωττιστή ότι θα χρησιμοποιήσετε κάποια χαρακτηριστικά από συγκεκριμένες βιβλιοθήκες. Για παράδειγμα για να χρησιμοποιήσουμε τα cin, cout πρέπει να προσαρτήσουμε στο πρόγραμμά μας τη βιβλιοθήκη iostream, για την παραγωγή τυχαίων αριθμών τη βιβλιοθήκη stdlib.h και για την ώρα την time.h .
Όλα αυτά φαίνονται παρακάτω όπου έχουμε προσθέσει μια αρχική οθόνη του παιχνιδιού:

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
using  namespace  std;

int main ()
{
   cout << "**********************************" << endl;
   cout << "*   Rock-Paper-Scissors Game Ver. 1.0   *" << endl;
   cout << "**********************************" << endl;
   return 0;
}
```

Τώρα ήρθε η σειρά σας να γράψετε κώδικα

Δοκιμάστε να εκτελέσετε το παραπάνω πρόγραμμα και αφού βεβαιωθείτε ότι τρέχει σωστά προσθέστε ακόμα μια γραμμή έτσι ώστε να εμφανίζεται και το όνομά σας ως developer ή programmer.

Παίζοντας στην τύχη

Θέλουμε ο υπολογιστής να επιλέξει στην τύχη ένα εκ των Πέτρα/Ψαλίδι/Χαρτί. Για αυτό τον λόγο θα χρησιμοποιήσουμε τη συνάρτηση rand( ) η οποία επιστρέφει έναν τυχαίο ακέραιο αριθμό. Πειραματιστείτε λίγο με την rand καλώντας την μερικές φορές όπως φαίνεται παρακάτω:

```
   srand (time(NULL));  // Αρχικές ετοιμασίες για να ξεκινήσει η παραγωγή τυχαίων
                        // αριθμών,( που δεν θέλετε να ξέρετε ☺). Χρειάζεται μια
                        //φορά στην αρχή του προγράμματος

   cout << rand( ) << endl;      // Θα εμφανιστούν τρεις αριθμοί
   cout << rand( ) << endl;
   cout << rand( ) << endl;
```

Αν εκτελέσετε το πρόγραμμά σας 2-3 φορές θα ανακαλύψετε ότι κάθε φορά οι τρεις αριθμοί είναι διαφορετικοί.

Θέτοντας όρια

Οι αριθμοί που παράγονται από την rand ναι μεν είναι ακέραιοι που προκύπτουν με τυχαίο τρόπο, αλλά εμείς θέλουμε να επιλέξουμε με τυχαίο τρόπο όχι μεταξύ οποιωνδήποτε ακέραιων αλλά μόνο μεταξύ τριών, για την ακρίβεια 0 ή 1 ή 2. Το σκεπτικό είναι ότι κάθε αριθμός θα αντιστοιχιστεί σε μια επιλογή του παιχνιδιού (R, P, S). Για να το πετύχουμε αυτό πρέπει να σκεφτούμε μια πράξη η οποία αν γίνει σε οποιοδήποτε ακέραιο θα επιστρέφει 0,1 ή 2. Αυτή η πράξη είναι το υπόλοιπο της ευκλείδειας διαίρεσης το οποίο υλοποιείται στη C++ με τον τελεστή %.

Έτσι το παρακάτω τμημά κώδικα μας εξασφαλίζει ότι ο υπολογιστής θα επιλέξει στην τύχη έναν από τους αριθμούς 0, 1 ή 2, εκχωρώντας τον στη μεταβλητή choice.

```
choice = rand( ) % 3;
```

Η σειρά του υπολογιστή

Ο υπολογιστής θα παίζει σε κάθε γύρο καλώντας τη rand( ) και υπολογίζοντας το υπόλοιπο της διαίρεσης με το 3. Αν αυτό είναι 0 σημαίνει ότι ο υπολογιστής παίζει Πέτρα (**R**), 2 σημαίνει Ψαλίδι (**S**) και 1 Χαρτί (**P**). Να συμπληρώσετε το παρακάτω τμήμα κώδικα έτσι ώστε στη μεταβλητή computer να καταχωρείται το γράμμα που αντιστοιχεί στον αριθμό.

```
choice = rand( ) % 3;

if (choice == 0)
    computer = 'P';

else if _____________ :
    ___________________

else :
    ___________________
```

Βελτιστοποίηση: Μπορείτε να σκεφτείτε έναν τρόπο αντιστοίχησης των 0,1,2 στα 'R, 'P', 'S' χωρίς να χρησιμοποιήσετε την εντολή if; **Υπόδειξη**: Χρησιμοποιήστε ένα πίνακα με τα τρία γράμματα.

Τώρα είναι η σειρά σου

Τώρα είναι η σειρά του παίκτη. Το πρόγραμμα θα ζητάει από τον παίκτη την επιλογή του η οποία θα είναι 'R', 'P', ή 'S'. Με χρήση της cout εμφανίζει σχετικό μήνυμα ενώ η cin διαβάζει την επιλογή του χρήστη και την καταχωρεί στη μεταβλητή player:

```
cout << " Keep Calm, it is your turn now " << endl;
cout << "Enter choice : P: Paper, S: Scissors and R: Rock : " << endl;
cin >> player;
```

Στη συνέχεια ελέγχοντας όλους τους συνδυασμούς μεταξύ computer και player το πρόγραμμα θα εμφανίζει κατάλληλο μήνυμα για το νικητή ή την ισοπαλία του γύρου, αλλά πρώτα θα εμφανίζει τις επιλογές και των δυο, ώστε να φαίνεται ότι δεν ..... κλέβει.

Παρακάτω δίνεται μια οθόνη εκτέλεσης του παιχνιδιού.

```
**************************************
*       Rock-Paper-Scissors Game Ver. 0.1       *
*          programmed by Euripides              *
**************************************

Keep Calm, it is your turn now
Enter choice : P: Paper, S: Scissors and R: Rock
P
You : P
Computer : S
```

Όταν πρέπει να ελέγξουμε πολλές περιπτώσεις

Τώρα ήρθε η στιγμή να υλοποιήσουμε την καρδιά του αλγορίθμου, δηλαδή τον έλεγχο όλων των πιθανών συνδυασμών των επιλογών των παικτών και τον καθορισμό του νικητή του γύρου. Απλά θα πρέπει να συμπληρώσετε το παρακάτω τμήμα κώδικα ώστε να ελέγχει όλες τις περιπτώσεις.

```
if  (player == 'S'' &&  computer == 'P' )
   cout << "I won this round!!!" << endl;

else if _____ :
   ……
   ……
```

Όλοι οι συνδυασμοί είναι 3x3 = 9, όμως μπορείτε με μια απλή παρατήρηση να ελέγξετε δυο λιγότερους.

Επαναλαμβανόμενοι Γύροι

Μέχρι στιγμής έχουμε σχεδιάσει ένα παιχνίδι το οποίο όμως έχει ένα μόνο γύρο. Εμείς όμως θέλουμε να παίζουμε συνέχεια μέχρι να βαρεθούμε. Για να το πετύχουμε αυτό θα πρέπει να βρούμε έναν τρόπο να επαναλαμβάνεται η παραπάνω διαδικασία μέχρι να δώσουμε εμείς ένα μήνυμα ότι βαρεθήκαμε και θέλουμε να σταματήσουμε. Ένας εύκολος τρόπος θα ήταν να θέσουμε όλη τη διαδικασία που υλοποιήσαμε προηγουμένως μέσα σε μια επανάληψη while όπως φαίνεται παρακάτω:

```
while  ()  {

   ……

   ……
}
```

Τώρα μπορείτε να παίξετε ΠΨΧ όσες φορές θέλετε.

Πότε σταματάμε;

Όπως σύντομα θα διαπιστώσετε έχουμε δυο μικρά προβληματάκια. Το πρώτο είναι ότι ο παραπάνω αλγόριθμος δεν σταματάει ποτέ, και το δεύτερο ότι πρέπει να κρατάμε το σκορ στο χαρτί. Ας ασχοληθούμε με το πρώτο. Το πρόβλημα είναι η εντολή while (true) { } η οποία θα πρέπει να αντικατασταθεί από μια συνθήκη η οποία θα διαπιστώνει αν εμείς θέλουμε να συνεχίσουμε το παιχνίδι ή όχι. Καλείστε λοιπόν τώρα να ρωτήσετε τον χρήστη αν θέλει να συνεχίσει :

```
cout << " Continue? (Y/N) " ;
cin >> answer;
```

Από την απάντηση θα κριθεί ο τερματισμός ή όχι της επανάληψης της οποίας τη συνθήκη θα πρέπει να τροποποιήσετε κατάλληλα σε σχέση με την απάντηση του παίκτη.

Υπόδειξη 1: Σκεφτείτε σε ποιο σημείο του προγράμματος θα τοποθετήσετε την παραπάνω εντολή ώστε όταν ο χρήστης βαρεθεί να μην εκτελείται άλλος γύρος.

Υπόδειξη 2: Θυμηθείτε ότι μπορούμε να χρησιμοποιήσουμε μεταβλητές και με λογικές τιμές στην C++ ☺.

Για παράδειγμα το παρακάτω τμήμα προγράμματος διαβάζει αριθμούς μέχρι να δοθεί 0 και υπολογίζει και εμφανίζει το πλήθος των θετικών.

```
int positives = 0;                    // ακόμα δεν έχει δοθεί κανένας θετικός
cin >> number;                        // διαβάζει έναν αριθμό από το πληκτρολόγιο
while (number != 0) {                 // Όσο number ≠ 0 Επανάλαβε τις παρακάτω εντολές
  if ( number > 0 ) {
     positives++ ;          // αυξάνει την positives κατά 1 αφού ήρθε ένας ακόμα θετικός
  }
  cin >> number;
}
cout << positives;
```

Κρατάμε σκορ;

Ας λύσουμε τώρα το δεύτερο προβληματάκι που διαπιστώσαμε, ότι δηλαδή ενώ έχουμε τη δυνατότητα να παίζουμε πολλούς γύρους, στο τέλος δεν φαίνεται το σκορ. Για να κρατάμε το σκορ αρκεί να μετράμε τις νίκες κάθε παίκτη και στο τέλος του παιχνιδιού να εμφανίζουμε το τελικό σκορ. Άρα χρειαζόμαστε δυο μεταβλητές player_wins και computer_wins. Η κάθε μεταβλητή θα αυξάνεται ανάλογα με το αποτέλεσμα του αγώνα. Αν για παράδειγμα κερδίσουμε εμείς θα πρέπει στην αντίστοιχο μπλοκ if να μπει η παρακάτω εντολή:

player_wins = player_wins + 1 ή η ισοδύναμή της *player_wins++*

η οποία αυξάνει τη μεταβλητή player_wins κατά 1.

Στο τέλος θα εμφανίσουμε το τελικό σκορ .

Ένα προβληματάκι ακόμα

Τι θα συμβεί αν ο χρήστης δεν δώσει κάποιο από τα κεφαλαία γράμματα R,P,S. Πως μπορούμε να αντιμετωπίσουμε αυτό το πρόβλημα;