



Mini-projet : FizzBuzz

GOMEZ Matias & MOHAMED RAFFIQUE Asfar

Groupe : 1G

06/10/2016

Sujet : Réaliser un programme permettant à un joueur d'affronter un ordinateur dans une partie de FizzBuzz.

Table des matières :

I- Introduction	2
II- Les hypothèses de travail et les choix de programmation	3
III- Les problèmes rencontrés	3
IV- Le jeu d'essais	4-5
V- Conclusion	5
Annexe	6-7-8

I- Introduction

Nous avons réalisé ce mini-projet en binôme à partir du samedi 24 septembre. Le but de ce mini-projet est de créer un programme permettant à un joueur d'affronter un ordinateur dans une partie de Fizz Buzz, le programme de ce projet a été réalisé sur Code::Blocks en 4 jours.

FizzBuzz est un jeu qui demande le nom du joueur au début, puis il demande la valeur minimale et maximale entre lesquelles il souhaite jouer contre l'ordinateur et finalement le programme demande au joueur qui veut commencer entre l'ordinateur et le joueur.

Ainsi le programme demande au joueur et à l'ordinateur de compter de 1 en 1 dans la console chacun à leur tour, à chaque fois que le joueur ou l'ordinateur saisi un multiple de 3, le résultat sera remplacé par le mot « Fizz » et pour un multiple de 5 le résultat sera remplacé par le mot « Buzz » et le mot « FizzBuzz » s'il s'agit d'un multiple de 3 et 5 en même temps.

Un message d'erreur apparaît si le joueur se trompe et donne la bonne réponse, ainsi un système de vies au joueur s'ajoute, au début celui-ci aura 3 vies et pour chaque erreur il perd une vie avec l'affichage du nombre de vies restantes entre parenthèses.

La partie s'arrête si le joueur n'a plus de vies ou lorsque la valeur maximale saisie au début du jeu est atteinte suivie d'un message « victoire » ou « défaite ». Et pour finir une proposition au joueur de recommencer ou quitter le jeu à la fin d'une partie.

II- Les hypothèses de travail et les choix de programmation

Nous avons créé deux conditions, l'une des deux s'enclencherait selon qui entre l'ordinateur et le joueur commencera à jouer.

Le problème de cette manipulation malgré le fait qu'elle fonctionne, elle rendait le programme assez chargé et compliqué à lire, c'est pour cela que nous avons simplifié cette partie du programme de manière à avoir qu'une seule boucle (boucle qui contient le début et la fin de la partie).

III- Les problèmes rencontrés

L'avertissement en cas d'erreur par le joueur n'était pas encore au point, cela était dû à l'ordre dans lequel étaient posées les conditions, on a donc finalement commencé par demander si mult3 et mult5 étaient vrais puis pour mult3, puis pour mult5, et on finit par demander si dans le cas où les conditions précédentes ne sont pas remplies, si la valeur saisie par le joueur est bien bonne.

La conversion de « string » vers « int » n'était pas complètement maîtrisée, cela a donné un peu de retard à la finalisation du programme.

Si la dernière valeur à rentrer était 20 et bien le programme proposait quand même de saisir un nombre à « MAX+1 » ou bien c'était l'ordinateur qui donnait cette valeur alors que la dernière valeur a déjà été saisie. On a remédié à cela en crée une condition qui faisait que le programme continuait seulement si « i » était inférieur ou égal à « MAX ».

IV- Le jeu d'essais

Le but du test		Les valeurs choisies en entrée		Les résultats prévus pour ce test
Test 1	<u>Cas général</u> : vérifier que la valeur minimale est strictement inférieure à la valeur maximale.	MIN = 20	MAX=25	- (La valeur est sauvegardé dans la variable)
	<u>Cas particulier</u> : si le joueur saisie une valeur MIN supérieur à MAX.	MIN = 20	MAX = 5	- Affichage : "Assurez-vous que le maximum soit supérieur au minimum SVP."
Test 2	<u>Cas général</u> : vérifier que le minimum et le maximum sont des valeurs positives.	MIN = 1	MAX=15	- La valeur est sauvegardé dans la variable
	<u>Cas particulier</u> : si le joueur entre MIN et MAX avec des valeurs négatives.	MIN = -1	MAX = -3	- <u>Message affiché</u> : "Assurez-vous que le maximum soit positif" "Assurez-vous que le minimum soit positif"
Test 3	<u>Cas général</u> : vérifier que le joueur entre bien "Fizz" lorsqu'il s'agit d'un multiple de 3.	Valeur courante : 6 Valeur entrée par le joueur : Fizz		-La partie continue.
	<u>Cas particulier</u> : Si le joueur entre un mot différent de « Fizz ».	Valeur entrée par le joueur : 6		- <u>Message affiché</u> : "ERREUR la réponse était Fizz" (-1 vie)
Test 4	<u>Cas général</u> : vérifier que le joueur entre bien "Buzz" lorsqu'il s'agit d'un multiple de 5.	Valeur courante : 10 Valeur entrée par le joueur : Buzz		-La partie continue.
	<u>Cas particulier</u> : Si le joueur ne saisit pas « Buzz ».	Valeur entrée par le joueur : 10		- <u>Message affiché</u> : "ERREUR la réponse était Fizz" (-1 vie)
Test 5	<u>Cas général</u> : vérifier que le joueur entre bien "FizzBuzz" lorsqu'il s'agit d'un multiple de 3 et 5.	Valeur courante = 15 Valeur entrée par le joueur : FizzBuzz		-La partie continue.
	<u>Cas particulier</u> : Si le joueur ne saisit pas « FizzBuzz ».	Valeur entrée par le joueur : 15		- <u>Message affiché</u> : "ERREUR la réponse était FizzBuzz" (-1 vie)

Test 6	<u>Cas particulier</u> : vérifier que le joueur entre la bonne valeur autre multiple de 5 et/ou 3.	Valeur que le joueur doit entrer : 19 Valeur entrée par le joueur : 43	- <u>Message affiché</u> : "ERREUR la réponse était 19 (-1 vie)"
Test 7	<u>Cas général</u> : vérifier qui de l'ordinateur ou du joueur commencera.	Valeur saisie : 0 Valeur saisie : 1	- <u>Message affiché</u> : "OK ! Alors je commence." - <u>Message affiché</u> : "Super! Alors tu commences."
Test 8	<u>Cas général</u> : vérifier si le joueur veut recommencer une partie.	Valeur saisie : 'n' Valeur saisie : 'o'	-La boucle s'arrête - <u>Message affiché</u> : "A la prochaine 😊 » -La boucle recommence - <u>Message affiché</u> : "Viens on joue ! »

V- Conclusion

La finalisation de ce programme bien qu'éprouvant mais très enrichissant nous a donné grande satisfaction.

Les difficultés que nous avons rencontrées lors du développement de notre programme ont été résolues grâce aux différentes recherches.

Ce projet nous a appris à mieux communiquer en binôme, à partager et rassembler nos idées, à raisonner de la manière la plus efficace possible, et bien sûr la réalisation de ce mini-projet a poussé plus loin nos connaissances en algorithmique et programmation C++.

Pour les améliorations, une interface graphique pourrait rendre l'expérience du jeu encore plus agréable.

Annexe

```

1 //GOMEZ MATIAS & MOHAMED RAFFIQUE AS FAR 1G
2 #include <iostream>
3 #include<string>
4 #include <cstdlib>
5 #include <cstdio>
6 using namespace std;
7
8 int main()
9 {
10     string prenom; //nom de joueur qui sera saisi par l'utilisateur
11     string nomPC="Ordinateur"; //nom qui apparaîtra lorsque l'ordinateur jouera
12     int MAX; //nombre Maximum qui sera saisi par le joueur
13     int MIN; //nombre minimum qui sera saisi par le joueur
14     int depart; //variable qui nous permettra de savoir qui de l'ordinateur
ou du joueur commencera à jouer
15     int i;
16     int nbVie=3; //nombre de vie du joueur de depart (3 au total)
17     char rejouer='o'; //variable de depart pour rejouer
18
19     string valeurJEU; //variable pour conversion
20     char tab[250];
21
22     bool mult3; //booléen pour les multiple de 3
23     bool mult5; //booléen pour les multiple de 5
24     bool tourOrdi;
25
26     while(rejouer=='o')
27     {
28         cout << "Viens on joue !\n" << endl; //Premier message afficher
29         cout << "Comment t'appelles-tu ?" << endl; //On demande le prenom à l'utilisateur
30         cin >> prenom; //Saisie du prenom par l'utilisateur
31
32         //On demande de quel nombre à quel nombre l'ordinateur et le joueur joueront
33         cout << "Donne moi la valeur minimale et maximale dans lesquelles nous jouerons." << endl;
34         cout << "MIN= ";
35         cin >> MIN; //Saisie du minimum (du nombre de depart
par l'utilisateur)
36         cout << "MAX= ";
37         cin >> MAX; //Saisie du maximum (du dernier nombre à
saisir dans le jeu)
38
39         /*Tant que l'utilisateur ne saisi pas un minimum ou un maximum
strictement positive, ou que le maximum n'est pas superieur au minimum faire...*/
40         while(MAX<=0 || MIN<=0 || MIN>MAX)
41         {
42             //Si il n'a pas saisi un maximum strictement positive demander à l'utilisateur de le
faire
43             if(MAX<=0)
44             {
45                 cout << "Assurez vous que le maximum soit positive" << endl;
46                 cin >> MAX;
47             }
48             //Si il n'a pas saisi un minimum strictement positive demander à l'utilisateur de le
faire
49             if(MIN<=0)
50             {
51                 cout << "Assurez vous que le minimum soit positive" << endl;
52                 cin >> MIN;
53             }
54             //Si il n'a pas saisi un maximum superieur au minimum demander à l'utilisateur de le
faire
55             if(MIN>MAX)
56             {
57                 cout << "Assurez vous que le maximum soit superieur au minimum SVP." << endl;
58                 cout << "MIN= ";
59                 cin >> MIN;
60                 cout << "MAX= ";
61                 cin >> MAX;
62             }
63         }
64
65         //On demande qui de l'ordinateur ou du joueur sera le premier à jouer
66         cout << "Je commence ou tu commences ?\n Si c'est moi qui commence tape '0' \n Si c'est toi
qui commences tape '1'" << endl;
67         cin >> depart; //saisie par l'utilisateur
68
69         //Tant que l'utilisateur n'a pas taper '0' ou '1' pour faire son choix...
70         while((depart!=0) && (depart!=1))
71         {
72             //Redemander à l'utilisateur de taper '0' ou '1' pour faire son choix
73

```



```

74     cout<<"J'AI DIS !!! Si c'est moi qui commence tape '0' \n Si c'est toi qui commences
tape '1'"<<endl;
75     cin>>depart;           //Saisie par l'utilisateur
76 }
77
78     tourOrdi= depart==0;    //On affecte la valeur '0' qui veut dire que l'ordinateur commence
79
80     if(tourOrdi)            //Si l'ordinateur commence...
81         cout<<"OK ! Alors je commence."<<endl;        //Ecrire message
82     else                    //Dans le cas contraire...
83         cout<<"Super! Alors tu commences."<<endl;      //Ecrire message
84
85     i=MIN;                  //On affecte la valeur minimal à i
86
87     while(i<=MAX)           //Tant que i est inferieur ou égal au maximum entré
88     {
89         mult3= i%3 == 0;    //le booleen mult3 est vrai dans le cas où i est divisible par 3
90         mult5= i%5 == 0;    //le booleen mult5 est vrai dans le cas où i est divisible par 5
91
92         if(tourOrdi)        //Si c'est le tour de l'Ordinateur faire...
93         {
94             if(mult3 && mult5)    //Si le nombre i actuel est divisible par 3 et 5...
95             {
96                 cout<<nomPC<<"> FizzBuzz"<<endl;    //Afficher "Ordinateur> FizzBuzz
97             }
98             else if(mult3)        //Si le nombre i actuel est divisible par 3...
99             {
100                 cout<<nomPC<<"> Fizz"<<endl;        //Afficher "Ordinateur> Fizz
101             }
102             else if(mult5)        //Si le nombre i actuel est divisible par
5...
103             {
104                 cout<<nomPC<<"> Buzz"<<endl;        //Afficher "Ordinateur> Buzz
105             }
106             else                //Si aucun des conditions precedentes ne
sont presentes
107
108                 cout<<nomPC<<"> "<<i<<endl;        //Afficher "Ordinateur> (nombre i actuel de
cette boucle)
109             }
110             else                //Si c'est le tour de l'utilisateur faire...
111             {
112                 cout<<prenom<<" ("<<nbVie<<" vies)"<<" > ";    //Afficher "prenom de l'utilisateur
(nombre de vie)>"
113                 cin>>valeurJEU;    //Saisie de ma valeur que
l'utilsateur pense etre la bonne
114
115
116                 //Si le nombre i actuel est divisible par 3 et 5 et que la valeur rentrer par
l'utilsateur est differrente de "FizzBuzz" faire...
117                 if((mult3 && mult5) && valeurJEU!="FizzBuzz")
118                 {
119                     cout<<"ERREUR la reponse etait FizzBuzz"<<endl;    //Afficher que
l'utilisateur s'est trompé
120                     nbVie=nbVie-1;    //Et enlever une vie
121                 }
122                 //Si le nombre i actuel est divisible par 3 et que la valeur rentrer par
l'utilsateur est differrente de "Fizz" faire...
123                 else if(mult3 && valeurJEU!="Fizz")
124                 {
125                     cout<<"ERREUR la reponse etait Fizz"<<endl;    //Afficher que
l'utilisateur s'est trompé
126                     nbVie=nbVie-1;    //Et enlever une vie
127                 }
128                 //Si le nombre i actuel est divisible par 5 et que la valeur rentrer par
l'utilsateur est differrente de "Buzz" faire...
129                 else if(mult5 && valeurJEU!="Buzz")
130                 {
131                     cout<<"ERREUR la reponse etait Buzz"<<endl;    //Afficher que
l'utilisateur s'est trompé
132                     nbVie=nbVie-1;    //Et enlever une vie
133                 }
134
135                 else if(!(mult3 || mult5))    //Si les conditions precedentes ne sont
pas remplis faire, et que le nombre i actuel n'est pas divisible par 3 ou 5 ni les deux
136                 {
137                     sprintf(tab,"%d",i);    //conversion de int vers string et envoie
le resultat dans tab
138                     if(valeurJEU!=tab)    //Si la valeur entrer par l'utilisateur est
differente du nombre i de cette boucle

```



```

139         {
140             cout<<"ERREUR la reponse etait "<<i<<endl;           //Afficher que
l'utilisateur s'est trompé
141             nbVie=nbVie-1;                                       //Et enlever une vie
142         }
143     }
144     if(nbVie==0)                                                 //Lorsque le nombre de vie du joueur tombe
à 0...
145     {
146         cout<<"Defaite :("<<endl;                               //Ecrire que l'utilisateur vien de perdre
147         i=MAX;                                                  //Et affecter la valeur MAX à i pour ainsi
sortir de la boucle
148     }
149 }
150 i=i+1;                                                         //Lorsque l'ordinateur ou le joueur a finis
son tour rajouter i+1 à la variable i
151 tourOrdi= !tourOrdi;                                           //Affecter un NON(Ordinateur) pour que ce
soit maintenant le tour de l'utilsteur ou de l'ordinateur (selon qui commence)
152 }
153
154 if(nbVie>0)                                                     //Une fois i==MAX on sort de cette boucle, ce qui veut dire que le
Joueur a fini la partie sans perdre toutes ses vies
155 {
156     cout<<"\nVictoire :)\n"<<endl;                             //Afficher la victoire du joueur
157 }
158
159 cout<<"C'etait chouette on rejoue ? (o/n)"<<endl;             //On demande si le joueur veut
rejouer et de repondre par 'o' pour oui ou par 'n' pour non
160 cin>>rejouer;                                                  //Saisie de la reponse par le
joueur
161
162 while(rejouer!='o' && rejouer!='n')                             //Tant que l'utilisateur n'a pas
reponsu par 'o' ou par 'n' faire...
163 {
164     cout<<"Veuillez a repondre par 'o' pour oui ou 'n' pour non, merci."<<endl;
//Demander a saisir correctement par 'o' ou 'n' sa reponse
165     cin>>rejouer;                                               //Saisie de la reponse de l'utilsteur
166
167 }
168 //Si il a repondu 'o' pour rejouer alors la boucle continue pas besoin de mettre de
condition
169
170 if(rejouer=='n')                                               //Si l'utilisateur a repondu 'n' pour ne pas rejouer...
171 {
172     rejouer='n';                                               //On change la valeur de la variable rejouer par 'n', ce qui
termine la boucle
173 }
174 }
175 cout<<"\nA la prochaine :)"<<endl;                             //Et pour finir on dit au revoir au joueur
176 return 0;                                                      //Le programme ce termine
177 }

```