 



**Mini-projet :**

**FizzBuzz**

## **GOMEZ Matias & MOHAMED RAFFIQUE Asfar**

## **Groupe : 1G**

## Sujet : Réaliser un programme permettant à un joueur

## d’affronter un ordinateur dans une partie de FizzBuzz.

06/10/2016

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 | Cas général : vérifier que la valeur minimale est strictement inférieure à la valeur maximale.Cas particulier : si le joueur saisie une valeur MIN supérieur à MAX. | MIN = 20 MAX=25MIN = 20 MAX = 5 | - (La valeur est sauvegardé dans la variable)- Affichage :"Assurez-vous que le maximum soit supérieur au minimum SVP." |
| Test 2 | Cas général : vérifier que le minimum et le maximum sont des valeurs positives.Cas particulier : si le joueur entre MIN et MAX avec des valeurs négatives. | MIN = 1 MAX=15MIN = -1 MAX = -3 | - La valeur est sauvegardé dans la variable- Message affiché : "Assurez-vous que le maximum soit positif""Assurez-vous que le minimum soit positif" |
| Test 3 | Cas général : vérifier que le joueur entre bien "Fizz" lorsqu'il s'agit d'un multiple de 3.Cas particulier : Si le joueur entre un mot différent de « Fizz ». | Valeur courante : 6Valeur entrée par le joueur : FizzValeur entrée par le joueur : 6 | -La partie continue.-Message affiché : "ERREUR la réponse était Fizz"(-1 vie) |
| Test 4 | Cas général : vérifier que le joueur entre bien "Buzz" lorsqu'il s'agit d'un multiple de 5.Cas particulier : Si le joueur ne saisit pas « Buzz ». | Valeur courante : 10Valeur entrée par le joueur : BuzzValeur entrée par le joueur : 10 | -La partie continue.-Message affiché : "ERREUR la réponse était Fizz"(-1 vie) |
| Test 5 | Cas général : vérifier que le joueur entre bien "FizzBuzz" lorsqu'il s'agit d'un multiple de3 et 5.Cas particulier : Si le joueur ne saisit pas « FizzBuzz ». | Valeur courante = 15Valeur entrée par le joueur : FizzBuzzValeur entrée par le joueur : 15 | -La partie continue.-Message affiché : "ERREUR la réponse était FizzBuzz" (-1 vie) |
| Test 6 | Cas particulier : vérifier que le joueur entre la bonne valeur autre multiple de 5 et/ou 3. | Valeur que le joueur doit entrer : 19Valeur entrée par le joueur : 43 | -Message affiché :" ERREUR la réponse était 19 (-1 vie)" |
| Test 7 | Cas général : vérifier qui de l'ordinateur ou du joueur commencera. | Valeur saisie : 0Valeur saisie : 1 | -Message affiché: "OK ! Alors je commence."-Message affiché: "Super! Alors tu commences." |
| Test 8 | Cas général : vérifier si le joueur veuxrecommencer une partie. | Valeur saisie : 'n'Valeur saisie : ‘o’ | -La boucle s’arrête-Message affiché: "A la prochaine ☺ »-La boucle recommence-Message affiché: "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

