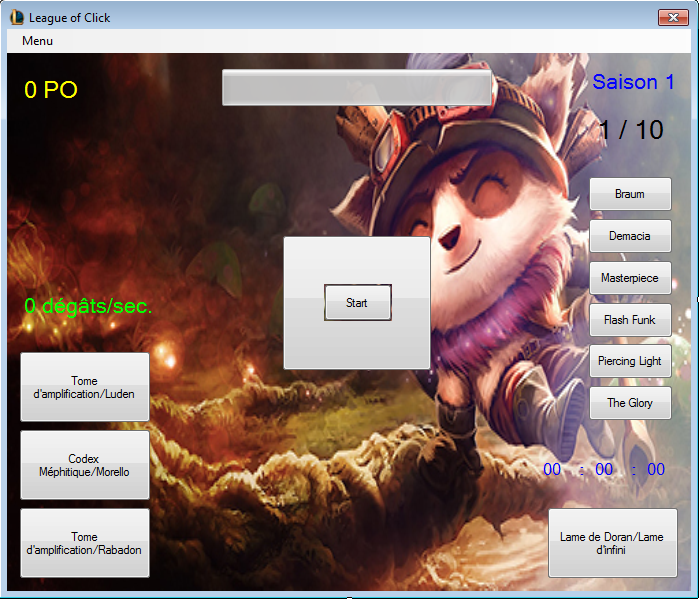
League of Click



Fazzi Enzo – TPA1

ETML

Durée : 11 semaines

Chef de Projet : Monica Rodrigues

Expert : Cindy Hardegger et Pierre Aubert

Table des matières

[1 Spécifications 5](#_Toc483309093)

[1.1 Titre 5](#_Toc483309094)

[1.2 Description 5](#_Toc483309095)

[1.3 Matériel et logiciels à disposition 5](#_Toc483309096)

[1.4 Prérequis 5](#_Toc483309097)

[1.5 Fonctionnalités demandées 5](#_Toc483309098)

[1.6 Cahier des charges 7](#_Toc483309099)

[1.6.1 Objectifs et portée du projet (objectifs SMART) 7](#_Toc483309100)

[2 Planification Initiale – tâches principales 7](#_Toc483309101)

[3 Analyse 8](#_Toc483309102)

[3.1 Opportunités 8](#_Toc483309103)

[3.2 Analyse et conception 8](#_Toc483309104)

[3.3 Conception des tests 10](#_Toc483309105)

[3.4 Planification détaillée – tâches détaillées 10](#_Toc483309106)

[3.4.1 Annexe: 10](#_Toc483309107)

[4 Réalisation 10](#_Toc483309108)

[4.1 Réalisation factuel 10](#_Toc483309109)

[4.2 Déclaration Variables : 11](#_Toc483309110)

[4.2.1 Variables principales 11](#_Toc483309111)

[4.2.2 Variables Niveaux 11](#_Toc483309112)

[4.2.3 Variables des armes 12](#_Toc483309113)

[4.2.4 Variables pour les musiques 12](#_Toc483309114)

[4.2.5 Variables pour le chronomètre 13](#_Toc483309115)

[4.3 Initialisation des objets : 13](#_Toc483309116)

[4.3.1 Initialisation des boutons 13](#_Toc483309117)

[4.3.2 Initialisation Autres 13](#_Toc483309118)

[4.4 Bouton Start 14](#_Toc483309119)

[4.5 Menu Tools 15](#_Toc483309120)

[4.5.1 Menu aide 15](#_Toc483309121)

[4.5.2 Menu de la musique 16](#_Toc483309122)

[4.5.3 Menu Code de Triche 18](#_Toc483309123)

[4.5.4 Menu Réinitialiser 19](#_Toc483309124)

[4.5.5 Menu pour Quitter 21](#_Toc483309125)

[Demande au joueur si il veut vraiment quitter la partie. 21](#_Toc483309126)

[4.6 Calcul des dégâts 21](#_Toc483309127)

[4.7 Gestion des dégâts 22](#_Toc483309128)

[4.7.1 Armes par click 22](#_Toc483309129)

[4.7.2 Armes passive (Luden) 24](#_Toc483309130)

[4.7.3 Armes passive (Morello) 25](#_Toc483309131)

[4.7.4 Armes passive (Rabadon) 26](#_Toc483309132)

[4.8 Bouton Musique 27](#_Toc483309133)

[4.9 Gestion du blocage des armes 28](#_Toc483309134)

[4.9.1 Arme Active 28](#_Toc483309135)

[4.9.2 Arme Passive (Luden) 29](#_Toc483309136)

[4.9.3 Arme Passive (Morello) 30](#_Toc483309137)

[4.9.4 Arme Passive (Rabadon) 31](#_Toc483309138)

[4.10 Gestion Niveau et Gain PO 32](#_Toc483309139)

[4.10.1 Niveau 1 32](#_Toc483309140)

[4.10.2 Niveau Boss 33](#_Toc483309141)

[4.11 Fermeture programme 34](#_Toc483309142)

[4.12 Chrono Tick 34](#_Toc483309143)

[4.12.1 Chronomètre 34](#_Toc483309144)

[4.12.2 Actualisation Label 35](#_Toc483309145)

[4.13 Code de Triche 35](#_Toc483309146)

[4.13.1 Tuer un Champion 35](#_Toc483309147)

[4.13.2 Gain 10K PO 36](#_Toc483309148)

[4.13.3 Gain 10M PO 37](#_Toc483309149)

[4.13.4 Aller Niveau 1 38](#_Toc483309150)

[5 Modifications – Problèmes rencontrés 39](#_Toc483309151)

[5.1 Dégâts par click 39](#_Toc483309152)

[5.2 Variables Vie Champion 39](#_Toc483309153)

[5.3 Changement Switch 40](#_Toc483309154)

[6 Tests 40](#_Toc483309155)

[6.1 Dossier des tests 40](#_Toc483309156)

[7 Conclusion 41](#_Toc483309157)

[7.1 Bilan des fonctionnalités demandées 41](#_Toc483309158)

[7.2 Bilan de la planification 42](#_Toc483309159)

[7.3 Bilan personnel 42](#_Toc483309160)

[8 Divers 43](#_Toc483309161)

[8.1 Journal de travail 43](#_Toc483309162)

[8.2 Webographie 47](#_Toc483309163)

[9 Annexes 48](#_Toc483309164)

[9.1 Planification Détaillé 48](#_Toc483309165)

[9.2 Cahier des charges 54](#_Toc483309166)

[54](#_Toc483309167)

[9.3 Documents Annexes 58](#_Toc483309168)

# Spécifications

## Titre

League of Click

## Description

Ce projet sera réalisé sur Visual Studio 2010 en C# Objet.

Le but de ce projet est de créer un « clicker » qui tuera des « champions » avec des armes automatiques et manuelles. Il y aura quatre armes à activer, améliorables, qui feront plus ou moins de dégâts par millisecondes, en même temps, selon leur nombre d’activation. C’est-à-dire que tous les cinquante millièmes de secondes, le joueur inflige des dégâts sans rien faire selon la somme des boutons cliqués depuis le début de la partie (items rouge, vert et bleu). Il y aura aussi les items qui font des dégâts par clic fait par le joueur (items violet). Le joueur récoltera plus ou moins de PO[[1]](#footnote-1) par champions tués selon le niveau en cours. Il y aura dix niveaux différents et les champions auront de plus en plus de vie au fur et à mesure de l’avancement. Il faudra tuer dix champions par niveau. Du premier niveau au neuvième, cette règle est respectée, mais au niveau dix, il y aura un Boss avec cent milliards de points de vie. Ce dernier comportera un gros Boss (Veigar, inspiré de League of Legends) avec un temps limité de trente secondes pour le tuer, sous peine de redescendre au niveau neuf. A chaque fois que le joueur a battu tous les champions du niveau, un bouton s’affiche pour passer au niveau suivant.

## Matériel et logiciels à disposition

* Word : Rapport
* Excel : Planification et Journal de travail
* Photoshop
* Internet
* Visual Studio 2010

## Prérequis

* Bonne connaissance dans le langage C# objet
* Avoir fait le stage 1 et 2 en informatique

## Fonctionnalités demandées

|  |  |  |
| --- | --- | --- |
| Nom item | Coût de l’item (PO) | Points de dégâts |
| Items qui inflige des dégâts par clique | | |
| Lame de Doran | 10 PO | 5 |
| Epée longue | 100 PO | 50 |
| Sabre vicié | 200 PO | 100 |
| Phage | 1’000 PO | 500 |
| Pioche | 5’000 PO | 10’000 |
| Dernier souffle | 25’000 PO | 50’000 |
| BF Glaive | 300’000 PO | 1’000’000 |
| Soif-de-sang | 4'000'000 PO | 50'000’000 |
| Lame d’infini | 200'000'000 PO | 500'000’000 |
| Items qui inflige des dégâts constant | | |
| Tome d’amplification | 250 PO | 5 |
| Baguette explosive | 250’000 PO | 500 |
| Baguette trop grosse | 10'000’000 PO | 50’000 |
| Coiffe de Rabadon | 500'000’000 PO | 1'000’000 |
| Codex Méphitique | 50 PO | 2 |
| Tome d’amplification | 50'000 PO | 250 |
| Chapitre perdu | 1'000'000 PO | 12’500 |
| Morellonomicon | 50'000'000 PO | 300’000 |
| Tome d’amplification | 20 PO | 1 |
| Feu Follet éthéré | 5'000 PO | 100 |
| Baguette trop grosse | 250'000 PO | 5’000 |
| Echo de Luden | 20'000'000 PO | 100’000 |

|  |
| --- |
| * Un bouton permettant le clic (dégâts aux champions/click). * Un système de niveau (1,2,3 … 10) qui définit la difficulté pour tuer les champions (+ de vie). * Niv 1 : 10 pts de vie/champions et rapporte 2 PO/champions tué * Niv 2 : 250 pts de vie/champions et rapporte 50 PO/champions tué * Niv 3 : 1’000 pts de vie/champions et rapporte 250 PO/champions tué * Niv 4 : 5’000 pts de vie/champions et rapporte 1’000 PO/champions tué * Niv 5 : 100’000 pts de vie/champions et rapporte 25’000 PO/champions tué * Niv 6 : 1'000’000 pts de vie/champions et rapporte 250’000 PO/champions tué * Niv 7 : 5'000’000 pts de vie/champions et rapporte 1'000’000 PO/champions tué * Niv 8 : 25'000’000 pts de vie/champions et rapporte 5'000’000 PO/champions tué * Niv 9 : 250'000’000 pts de vie/champions et rapporte 50'000’000 PO/champions tué * Niveau 10 : 100'000'000’000 pts de vie du Boss final (tuable en 30 secondes) et le jeu se fini quand il est mort * Une barre de vie du champions en cours de combat. * 3 boutons d’amélioration des items des dégâts passif qui change de nom à chaque clique pour dire au joueur à quel niveau de l’item il en est.   - Tome d’amplification < Baguette explosive < Baguette trop grosse < Coiffe de Rabadon  - Codex Méphitique < Tome d’amplification < Chapitre perdu < Morellonomicon  - Tome d’amplification < Feu Follet éthéré < Baguette trop grosse < Echo de Luden   * Un bouton d’amélioration de l’item de dégâts par clique qui change de nom à chaque clique pour dire au joueur à quel niveau de l’item il en est.   - Lame de Doran < Epée longue < Sabre vicié < Phage < Pioche < Dernier souffle < BF Glaive < Soif-de-sang < Lame d’infini   * Un label qui affiche le nombre de PO que l’on a. * Un label pour afficher les dégâts infligé passivement tous les 50 millièmes de secondes. * Un label pour afficher le niveau auquel on se trouve.      * Les paramètres (menu).   - Aide (affiche ce que le joueur doit faire pour gagner)  - Musique (le joueur a la possibilité de mettre une musique ou non)  - Réinitialiser (Réinitialise tout le jeu au complet)  - Quitter (demande au joueur si il veut vraiment quitter le jeu)     * Ajout d’un chronomètre. |

## Cahier des charges

### Objectifs et portée du projet (objectifs SMART)

Le projet était à réaliser en 11 semaines réparti en 990 quarts d’heures. Cela fait donc 330 périodes. Ce projet sera ensuite évaluer par des experts et Madame Rodrigues elle-même.

# Planification Initiale – tâches principales

La planification initiale m’a pris 2 jours. Elle a été réalisée après que le cahier des charges soit valider.

# 

# Analyse

## Opportunités

***Les compétences :***

* Connaissances plus approfondies du C# objet.
* Compétence de la « progressbar » à connaître.
* La gestion des différents niveaux de champions et des 10 niveaux.
* La gestion du code derrière les labels.

***Liste du matériel :***

* Ordinateur, clavier et souris
* Visual Studio
* Internet
* Word
* Excel

***Recherche d’information :***

* Recherche pour le fonctionnement de la « progressbar ».
* Recherche pour le système de niveau (champions et difficultés).
* Recherche pour le système de dégâts passifs.

## Analyse et conception

**Visual Studio 2010 :**



Microsoft Visual Studio est une suite de logiciels de développement créé par Microsoft pour fonctionner sur Windows. Visual Studio 2017 est la version la plus récente à ce jour qui est aussi disponible sur MacOSX, mais ce projet sera fait sur Windows, sur le logiciel Visual Studio 2010 vu que cette version est préinstallé sur l’ordinateur. Cette version de Visual Studio est sortie en avril 2010.

Visual Studio est un ensemble d’outils de développement permettant de générer des ***Applications WEB et des applications pour la bureautique et pour les mobiles.***

Ce logiciel sert au programmeur tout particulièrement. Il est possible de programmer en plusieurs langages. Par exemple : C++, C#, F# et Visual Basic).

**C# Objet :**

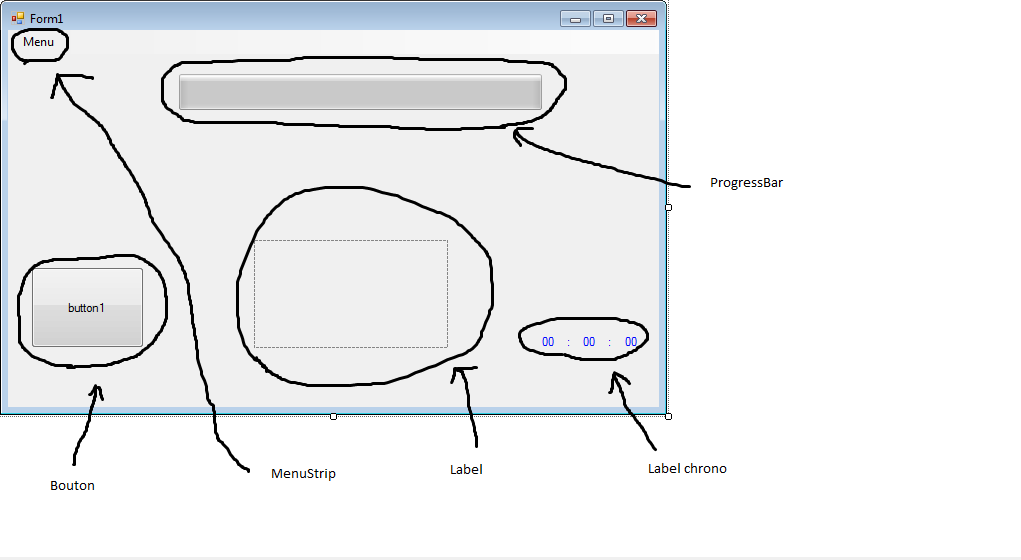


Le C# a été créé en 1998 par Microsoft.

Il a influencé plusieurs langages comme : le Swift, le Vala, le F#, le Kotlin et le TypeScript. Et a été influencé par le C++ et le Java.

Le C# apporte un typage de haute qualité. Le typage permet d’améliorer la qualité de la programmation. Dans les langages avec des typages statique, l’objectif principale est d’intercepter les erreurs de *type de données* lors de la compilation (lancement du programme).

Pourquoi avoir choisi le C# objet comme langage pour ce projet et pas un autre? Le C# objet permet de créer facilement des objets comme ci-dessous. C’est donc beaucoup plus pratique. Vu que ce projet est basé sur la création de labels et boutons, c’est ce qu’il y a de mieux. Et il est facile à utiliser et à comprendre.



Pour ce projet, il faudra utiliser les différents objet dans la boîte à outils.

De base, ce projet aurait dû être coder en Unity, mais c’était inadéquat. Vu que sur Unity c’est principalement des formes à gérer et très peu de code (pour un clicker), et vu que ce projet est « un petit » projet, il a fallu le faire sur Visual Studio.

## Conception des tests

Liste des fonctionnalités qu’il faudra tester:

* Progressbar (vie du champions)
* Résultat du label de PO
* Résultat des dégâts/secondes
* Label dégâts par click (etc.)
* Système de niveau 1-10
* Système de niveau du champions
* Les boutons d’amélioration des armes passives
* Le bouton d’amélioration d’arme aux dégâts par click
* Les paramètres

## Planification détaillée – tâches détaillées

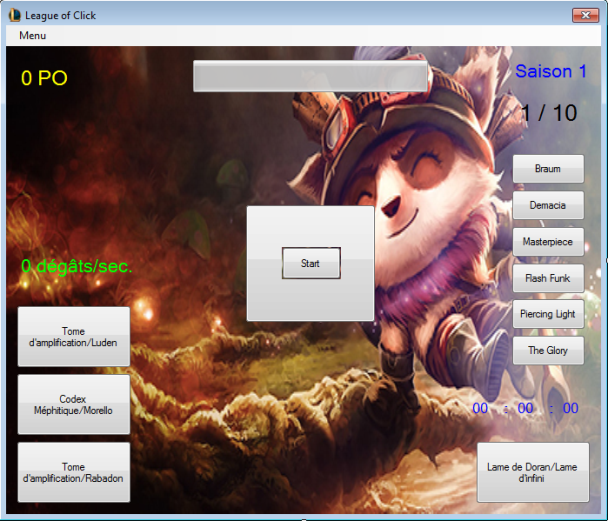
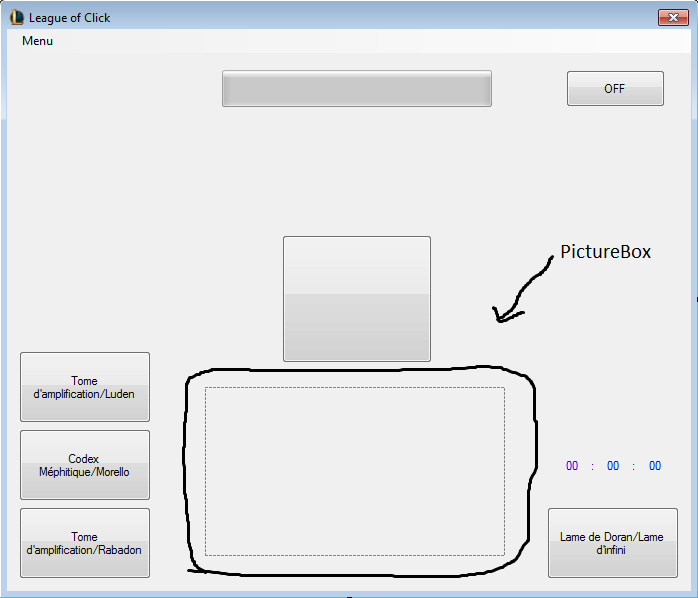
### Annexe:

[INF\_PAPP\_PlanifDétal-JDT\_fazzien.xlsm](file:///\\setmseb2k80002\PAPP_Eleves$\TPA\1\fazzien\PAPP\Projet%20Personnel\INF_PAPP_PlanifDétal-JDT_fazzien.xlsm)

# Réalisation

## Réalisation factuel

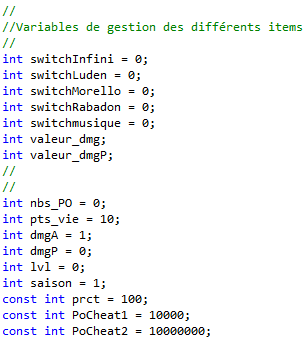
Avant Après



Ceci était le design de base (Avant), mais l’image qui était dans la picture box n’était pas très bien représenté. Du coup, suite à une modification, elle a été mise en arrière-plan pour mieux la distingué. L’image en arrière-plan dans l’image de droite était dans la « PictureBox » de l’image de gauche.

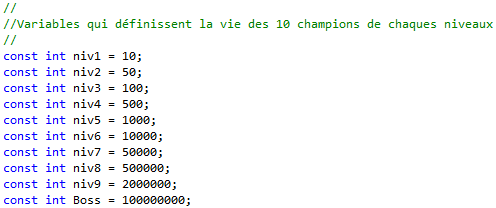
## Déclaration Variables :

### Variables principales



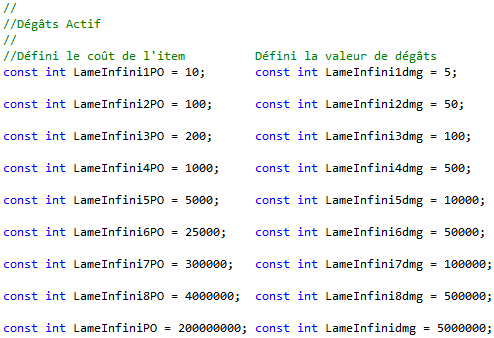
La déclaration des variables principales est très important dans les code. Elles servent à beaucoup d’endroit et à beaucoup de chose. Les switchs permettent notamment de remplacer tous les if, else if qui étaient présent de base.

### Variables Niveaux

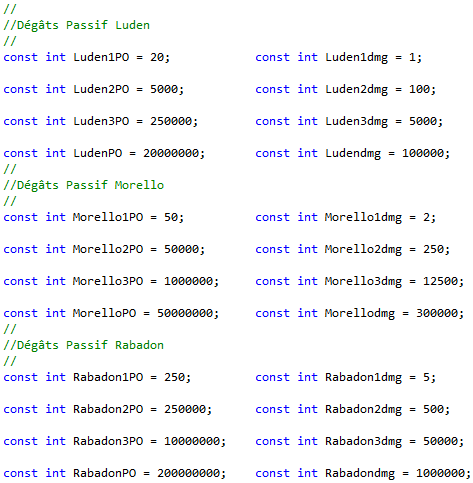


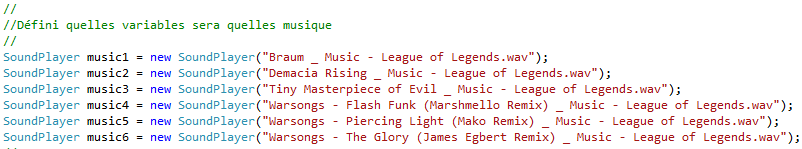
Ces variables définissent la vie des différents champions de chaque niveaux. Il y a eu modification des points vie par rapport au cahier des charges.

### Variables des armes



Ces variables définissent le nombre de PO obligatoire pour pouvoir acheter un item au dégâts actif et les dégâts que chaque items actif inflige.



Ces variables définissent comme l’image au-dessus les PO nécessaire et dégâts que cela inflige mais pour les items passif.

### Variables pour les musiques

Chaque SoundPlayer défini quelle musique sera défini à quelle variable.

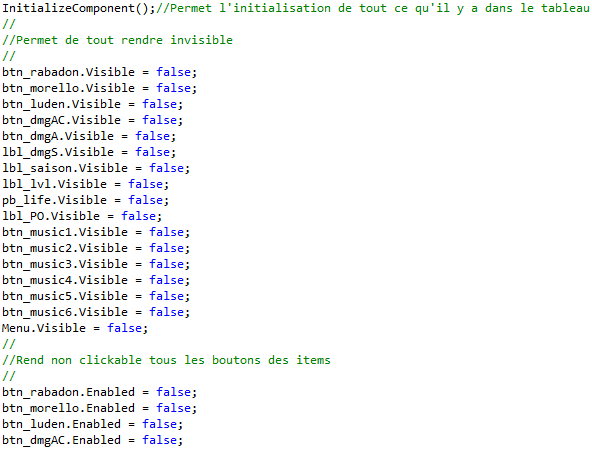
### Variables pour le chronomètre

Le but ici est d’initialiser la fonctionnement du chronomètre.

## Initialisation des objets :

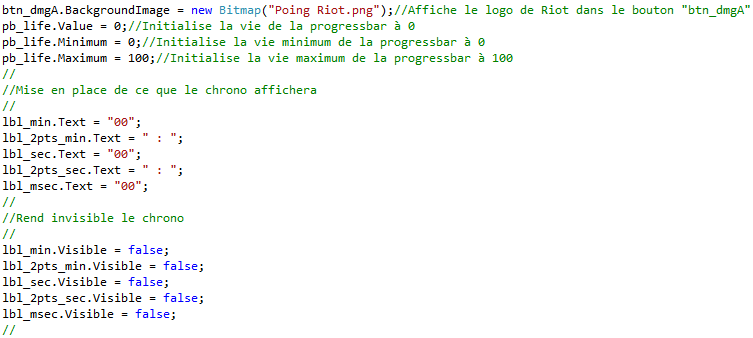
Cette fonction permet de tout initialiser dans le form.

### Initialisation des boutons



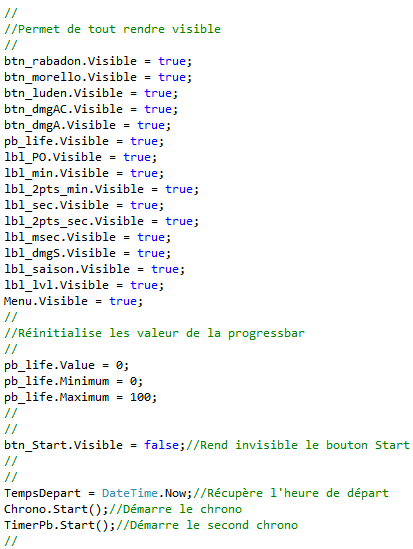
Au démarrage du programme, tout est invisible et le programme n’affiche que le bouton Start. Il a fallu rendre également non-cliquable les 4 boutons de dégâts, car au démarrage du programme, si il n’y avait pas ces 4 commandes, nous pouvions acheté un objet dès le début du jeu sans avoir de PO. Il suffisait d’appuyer sur les 2 Enter présent sur le clavier en même temps au moment du bouton Start et cela achetait un objet.

### Initialisation Autres



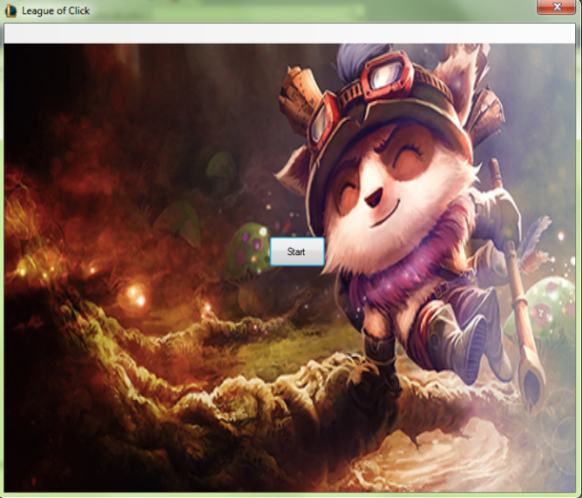
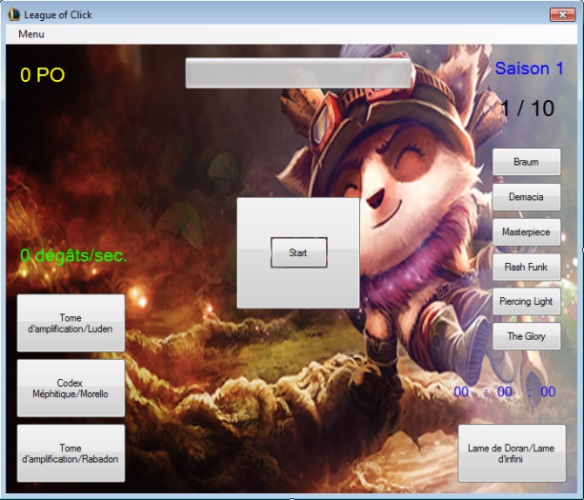
L’initialisation autres permet ici de définir les valeurs de base de la progressbar. Mais aussi l’initialisation de la partie chronomètre.

## Bouton Start



Le bouton Start permet au démarrage de n’afficher que le bouton Start et au moment où l’on clique sur le bouton, il devient invisible et affiche tout le reste du programme (label, bouton dégâts, menu et progressbar). Il démarre également les 2 chronos.

Avant click Après click

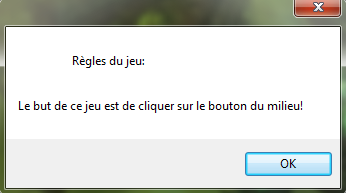
 

## Menu Tools

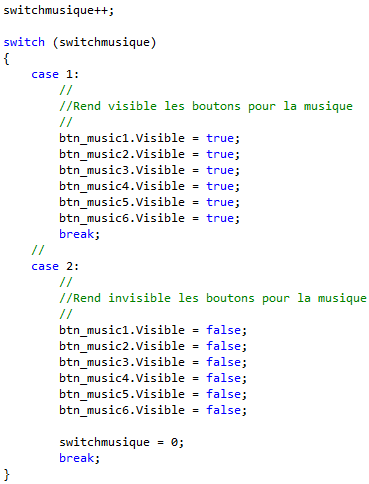
### Menu aide



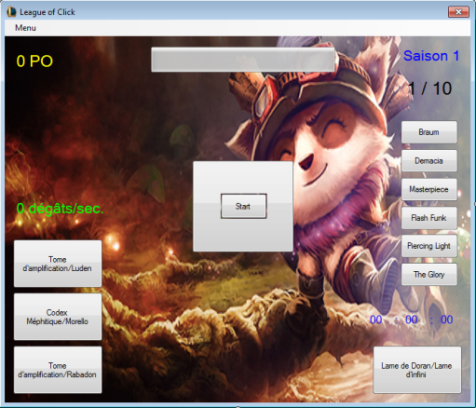
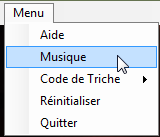
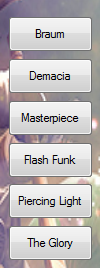
Le menu aide afficher une messagebox qui explique brièvement les règles du jeu.



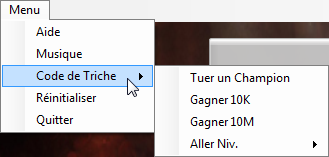
### Menu de la musique



De base, les 6 boutons de la musique ne sont pas affichés mais dès que le joueur appuie dans le menu pour la musique, cela affiche tous les boutons liée à la musique.

### Menu Code de Triche



Comme son nom l’indique, les codes de triche de sont pas censé être utilisé dans le jeu, se sont bien des codes de triche et non des capacités.

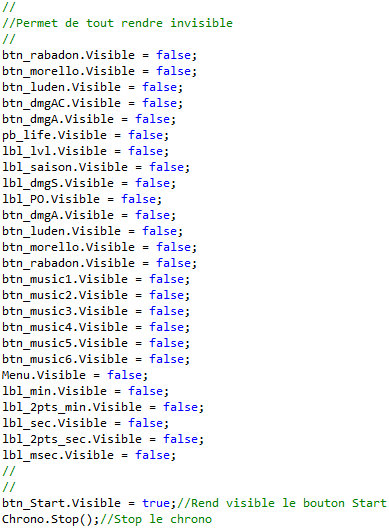
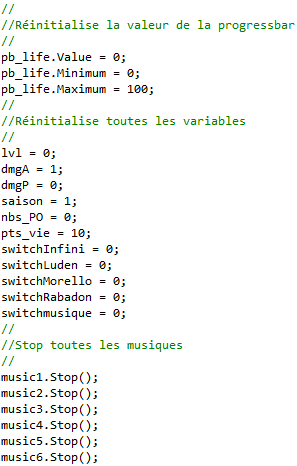
Tuer un champion : Cela permet de tuer n’importe quel champion du jeu, un champion du niveau comme le Boss.

Gagner 10K : Le joueur gagne instantanément 10 mille PO.

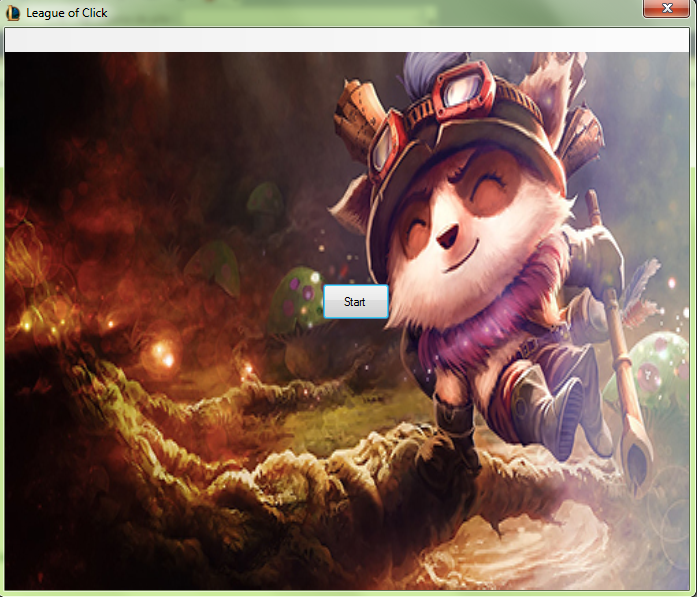
Gagner 10M : Le joueur gagne instantanément 10 millions de PO.

Aller Niv. : Le joueur peut choisir de directement aller à un niveau en particuliers.

### Menu Réinitialiser

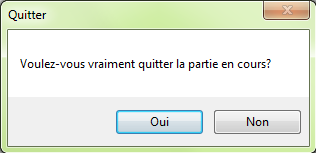
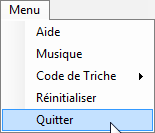


Ce menu sert à réinitialiser complètement le jeu. Il réinitialise toutes variables, stop les musiques, rend invisible tous les bouton du jeu et affiche le bouton Start.

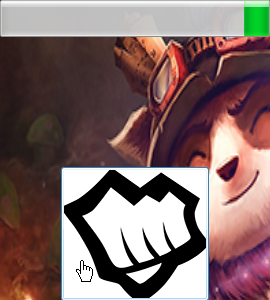
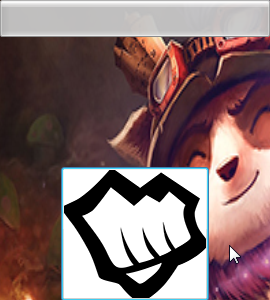
### Menu pour Quitter

### Demande au joueur si il veut vraiment quitter la partie.



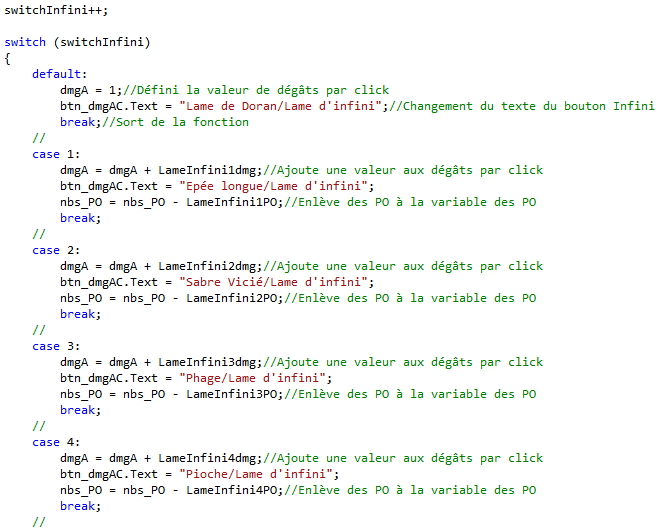
## Calcul des dégâts

Il a fallu créé une variable valeur\_dmg pour faire un calcul et faire des dégâts en pourcentage. Cela permet une meilleur fluidité au programme et un déplacement de la progressbar en continue et non par cran.



## Gestion des dégâts

### Armes par click

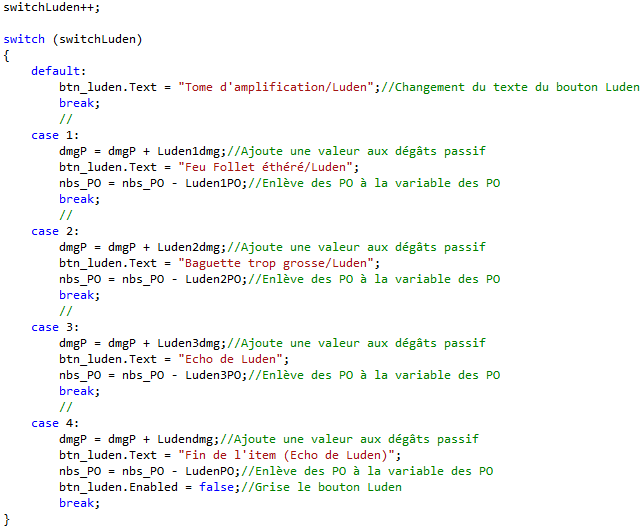


DmgA est égal au nombre de dégâts que l’arme par click infligera à la progressbar. Chaque click sur le bouton de dégâts par click augmente la variables switchInfini de 1. Chaque incrémentation de la variable, les dégâts de l’item augmente et le joueur doit payer l’item acheté. Cela change également le texte.



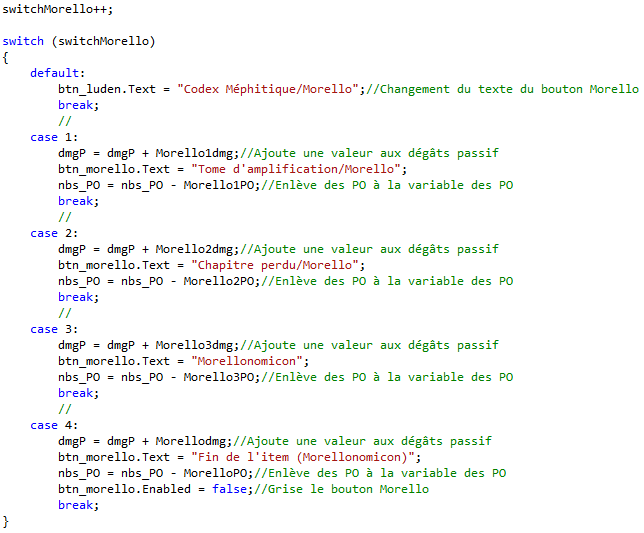
Ici, il a fallu faire la même chose que pour l’arme de dégâts par click mais avec les autres améliorations.

### Armes passive (Luden)



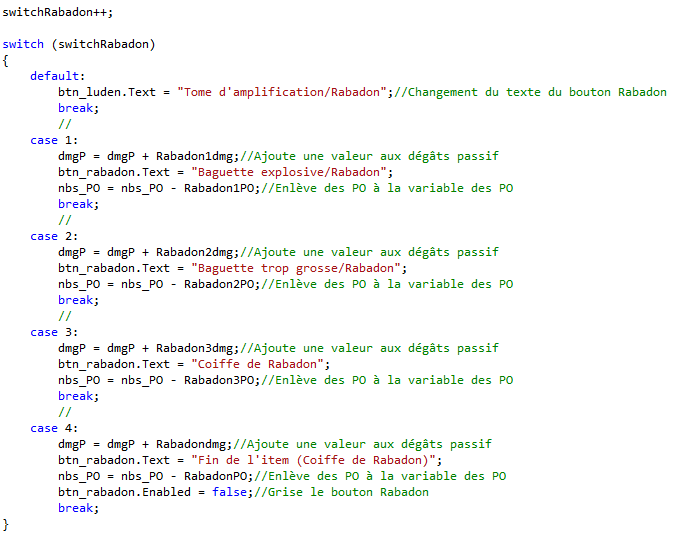
Ici, c’est pareil que pour l’arme aux dégâts par click, mais pour l’item Rabadon (dégâts passif). Cela fais les même procédé. Sauf que ici, la variables n’est pas la même que l’item Infini. Il a fallu créé un seconde variable de dégâts pour ne pas mélanger les 2 systèmes de dégâts différents.

### Armes passive (Morello)



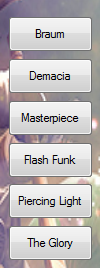
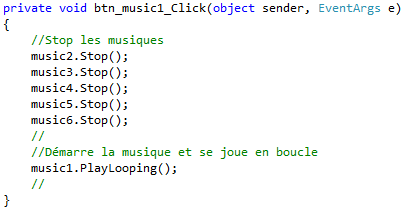
Exactement pareil que pour l’item Luden et il utilise la même variable que l’item Luden vu que c’est le même système de dégâts que celui-ci.

### Armes passive (Rabadon)



Et pareil pour celui-ci.

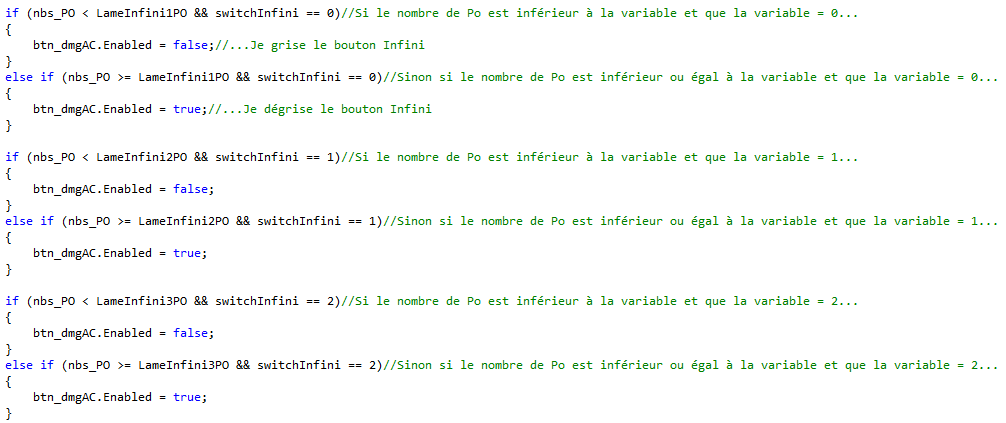
## Bouton Musique



Cette fonction permet de démarrer la musique que le joueur souhaite suivant les musique proposée. Il y a 5 autres événement comme celui-ci (btn\_music1\_Click/ btn\_music2\_Click… jusqu’à 6), c’est-à-dire que pour le 2, j’arrête la 1, 3, 4, 5 et 6 ème musique et lance la deuxième musique. Et ainsi de suite pour les autres boutons.

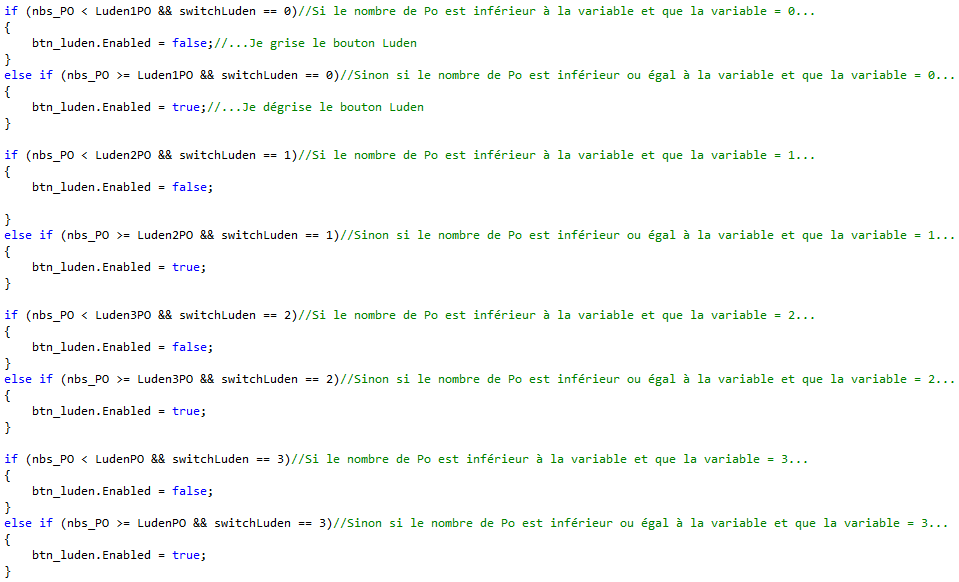
## Gestion du blocage des armes

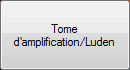
### Arme Active



Dans cette fonction, le programme regarde si le nombre de PO est suffisant pour acheter un item. Si ce n’est pas le cas, cela grise le bouton en question. Mais si le nombre est égal ou supérieur au montant demandé, cela dégrise le bouton Infini. Et cela fait la même chose juqu’au switchInfini = 8.

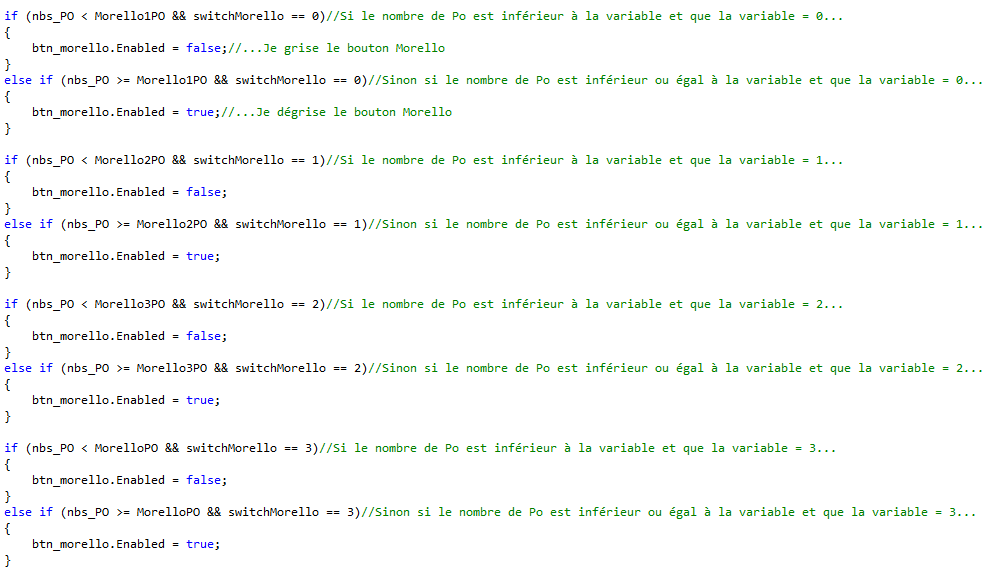
### Arme Passive (Luden)

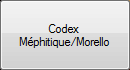




Dans cette fonction, le programme regarde si le nombre de PO est suffisant pour acheter un item. Si ce n’est pas le cas, cela grise le bouton en question. Mais si le nombre est égal ou supérieur au montant demandé, cela dégrise le bouton Luden.

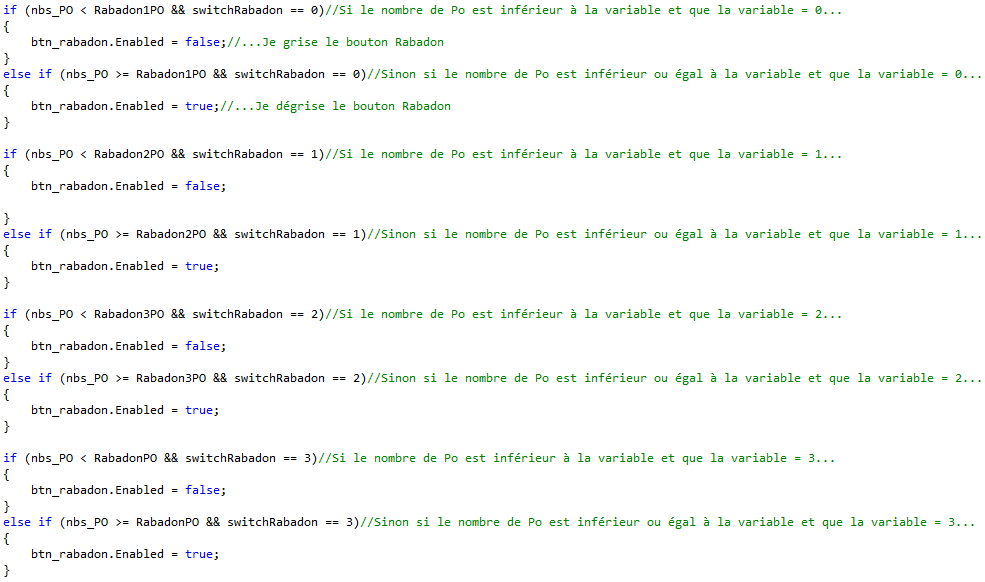
### Arme Passive (Morello)





Dans cette fonction, le programme regarde si le nombre de PO est suffisant pour acheter un item. Si ce n’est pas le cas, cela grise le bouton en question. Mais si le nombre est égal ou supérieur au montant demandé, cela dégrise le bouton Morello.

### Arme Passive (Rabadon)

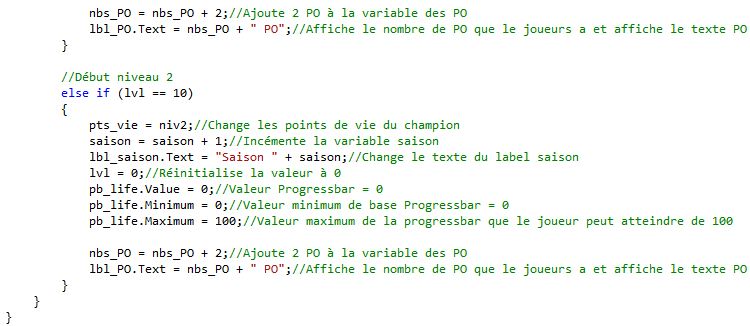




Dans cette fonction, le programme regarde si le nombre de PO est suffisant pour acheter un item. Si ce n’est pas le cas, cela grise le bouton en question. Mais si le nombre est égal ou supérieur au montant demandé, cela dégrise le bouton Rabadon.

## Gestion Niveau et Gain PO

### Niveau 1

Ici le premier if regarde si la variable est égal à 1, cela change l’image et la couleur des labels pour adapter les couleurs avec l’image. Puis cela va sur le second if :

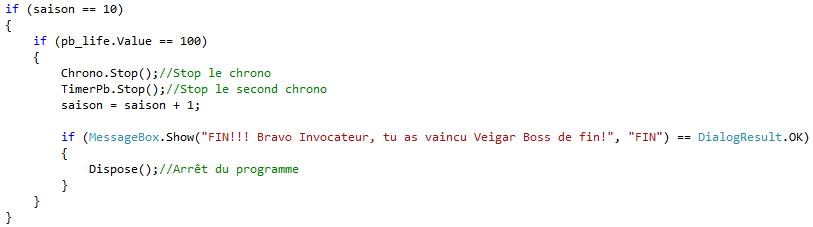
Le second if regarde si la valeur de la progressbar est égal à 100, si oui, le « lvl » s’incrémente de 1.

3ème if regarde si « lvl » est inférieur à, si oui, la valeur de la progressbar se « réinitialise », pareil pour la valeur minimum et la « vie » de la progressbar revient à 0. Ensuite, le joueur gagne 2 PO et cela s’affiche dans le label prévu.

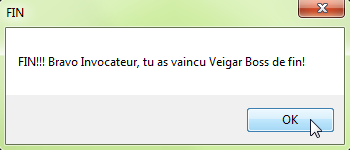
Sinon, si le « lvl » est égal à la valeur de la progressbar, si oui, revient à 0, la valeur minimum revient également à 0. La vie maximum est toujours de 100 mais la variable « pts\_vie » qui devient 250 et le joueur gagne 2 PO et l’affiche à nouveau.

Pour les autres niveaux c’est exactement pareil mais il y a juste les valeur qui change.

### Niveau Boss



Dès que le joueur a fini le niveau 10 (Boss), cela arrête le chrono et affiche une MessageBox qui indique au joueur qu’il a gagé. Il a fallu mettre l’augmentation de la variable « saison » pour ne pas afficher la MessageBox plusieurs fois.



## Fermeture programme

Le but est le même que pour le menu quitter mais pour la form directement (bouton X).

## Chrono Tick

### Chronomètre

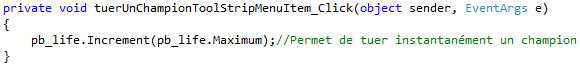
Ici, cela permet de récupérer l’heure actuel et de faire fonctionner le chronomètre.

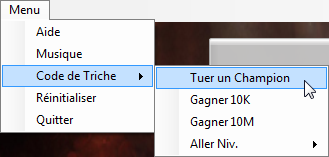
### Actualisation Label

Ici, il faut obligatoirement actualiser les label pour pouvoir afficher les bon résultats. Sinon, les label n’afficherons pas correctement les informations.

## Code de Triche

### Tuer un Champion

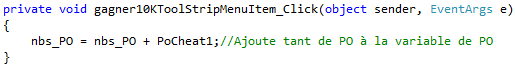


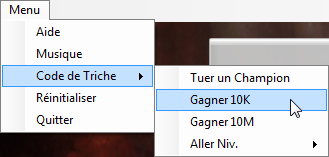




Cette commande permet de tuer n’importe quel champion, même le Boss. Vu que les dégâts se font en pourcentage et que la valeur de la progressbar est toujours égal à 100, la commande fait des dégâts direct qui sont égaux à 100.

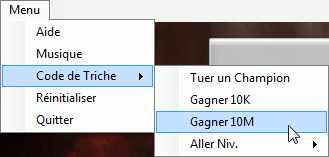
### Gain 10K PO





Cette commande permet au joueur de gagner directement dix mille PO et cela ajoute 10'000 à la variable de PO.

### Gain 10M PO

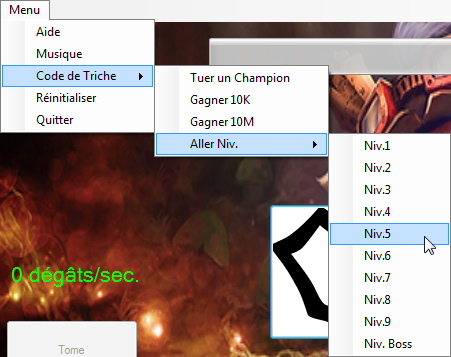


Cette commande permet elle aussi de gagner des PO (10'000'000) et cela ajoute 10'000'000 à la variable de PO.

### Aller Niveau 1

Ici, cela me permet de changer directement de niveau, par exemple, si le joueur est au niveau 1 et qu’il a envie d’aller directement plus loin, il peut aller niveau 5, il va dans le menu et demande d’aller au niveau 5.

Avant Click Après Click

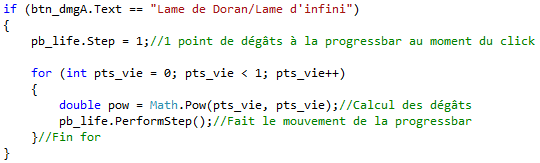
  

Cela l’emmènera instantanément au niveau souhaité.

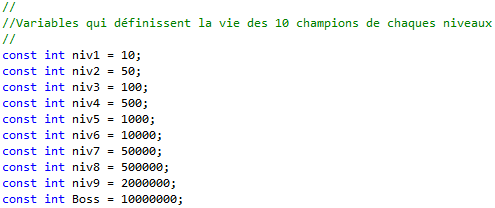
# Modifications – Problèmes rencontrés

## Dégâts par click

Le premier problème qui a été rencontrés affectait le reste du programme, c’était beaucoup moins fluide et pas adéquat. C’est-à-dire que le programme fonctionnait mais ce n’était pas fluide et la barre de progression saccadait avec cette méthode. Il a donc fallu changer la manière de faire des dégâts. La nouvelle méthode consiste à faire des dégâts en pourcentage. Une variable a été créé pour faire ces dégâts en pourcentage (dégâts par click \* 100 / les points de vie du champions). Le « pb\_life.Step » n’est donc la bonne commande (événement).



## Variables Vie Champion



Le second changement apporté au programme sont les variables qui définissaient la vie du champion.

Cette modification était requise. Elles ont dû être modifiées parce que si par exemple, le joueur était arrivé au niveau 2 sans avoir acheté aucun items, c’est-à-dire qu’il aurait toujours 1 point de dégâts par click, et que la vie était égale à 250, le calcul pour faire les dégâts en pourcentage donnait comme résultat un chiffre inférieurs à 1%, du coup, l’incrémentation ne se faisait pas, même en modifiant les variables nécessaire en float, l’événement Increment ne prenait pas en compte les chiffre à virgules. C’est donc pour ça que ce changement était nécessaire.

## Changement Switch

Ce troisième changement améliore la fluidité du jeu et m’a appris une nouvelle commande : le Switch.

Auparavant, le fonction « btn\_dmgAC\_Click / btn\_Luden\_Click / btn\_Morello\_Click / btn\_Rabadon\_Click / MenuMusique\_Click » comportaient des if et else. Mais cela ralentissait le programme, donc pour améliorer la fluidité du programme et l’optimiser, ils ont été remplacé par des Switch.

# Tests

## Dossier des tests

|  |  |  |
| --- | --- | --- |
| Objets | Description | Résultats |
| Arme Dégâts par click | Les dégâts se font correctement, dès que le joueur appuie sur le bouton correspondant, cela change bien les dégâts, le texte et l’enlèvement des PO | OK |
| Armes Dégâts par seconde | Les dégâts se font bien toutes les 50 millisecondes et le bouton qui correspond change bien le texte, les dégâts et enlève les PO | OK |
| Système de niveau | Chaque niveau fonctionnent. Le Boss fonctionne aussi mais il manque le chronomètre des 30 secondes pour tuer le Boss | Moyen |
| Barre de vie | Les dégâts à la « progressbar » se font correctement à l’exception des fameux 1 % parlé plus haut | Moyen |
| Menu | Chaque sous-menu fonctionnent | OK |
| Système de PO | Le système de PO est opérationnel, à l’achat d’un item, les PO s’enlève et bien et à chaque champions tués, gagne des PO | OK |
| Chronomètre | Le chrono démarre bien au moment du click du bouton « Start » | OK |

# Conclusion

## Bilan des fonctionnalités demandées

Tout le cahier des est fait à part le chronomètre ajouté pour le Boss qui permettait de n’avoir que 30 secondes pour tuer le Boss.

* L’arme des dégâts par click fonctionne. Elle fait les dégâts à la progressbar en pourcentage (100 point de vie : 10% de dégâts) et non brut (100 point de vie : 10 points de dégâts).
* Les armes aux dégâts toutes les 50 millisecondes fonctionnent également. Les dégâts se font aussi en pourcentage comme pour l’arme aux dégâts par click.
* Le système de niveau est opérationnel. Chaque fois que le joueur tue un champion, cela change instantanément de champion, de vie et d’image si besoin. Et les label qui s’actualisent correctement.
* Le menu et les sous-menu sont complet. Le joueur peut demander le sous-menu Aide, Musique, Code de Triche, Réinitialiser et Quitter. Ils sont tous opérationnels.
* Le chronomètre fonctionne correctement. Au moment du click sur le bouton « Start » le chrono démarre. Et au moment où le joueur décide de Réinitialiser le jeu, stop le chrono.
* Les 4 boutons de dégâts (passif et actif) font bien les changements de dégâts, les changements des textes.
* La barre de vie (progressbar) est opérationnel mais il y a un petit problème qui ne peut pas être résout. Les dégâts à la progressbar ne peuvent pas se faire en dessous de 1%. Même en changeant les variables nécessaire en type float, cela ne fait pas des dégâts avec des chiffres à virgules.
* Les différents labels sont tous fonctionnels. Les labels qui affichent le niveau où le joueur se trouve fonctionne correctement, même en utilisant les codes de triche. Le label de PO fonctionne également. Et le label qui affiche le nombre de dégâts par seconde que le joueur inflige fonctionne aussi.

## Bilan de la planification

Dans ma planification initiale, j’ai mis que je n’allais faire que 12 quart d’heure pour l’arme aux dégâts par click. Je pensais que ça allais être plus simple mais cela a été beaucoup plus difficile que prévu. Au début, je cherchais comment faire les dégâts et j’ai trouvé une méthode mais qui n’était pas la bonne j’ai donc dû refaire des recherches pour trouver un nouveau moyen de faire des dégâts.

Cette fois pour les armes aux dégâts passif, je pensais que ça allais me prendre plus que temps que ça mais finalement moins que prévu. Je pensais que vu que c’était 3 armes différentes, ça allais me prendre plus de temps que l’arme aux dégâts par click. Mais non, j’avais trouvé la bonne commande en quelques recherches. Il se trouve que ce fût la même commande pour l’autre arme.

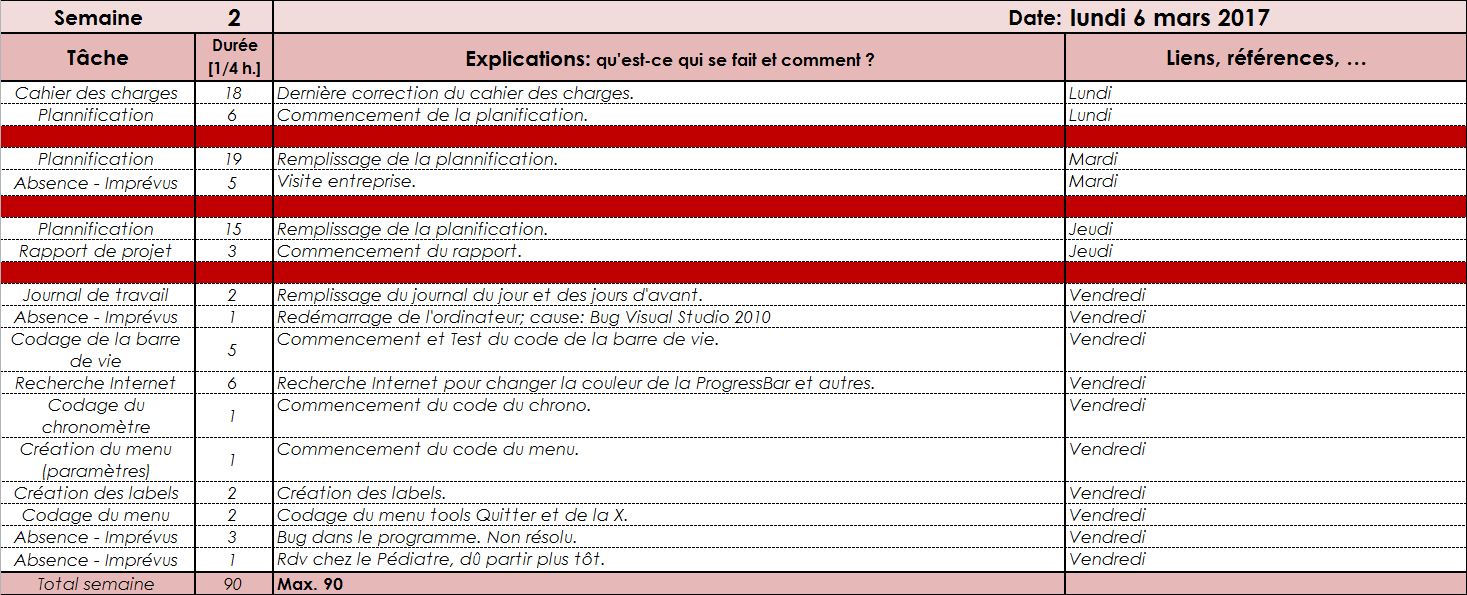
## Bilan personnel

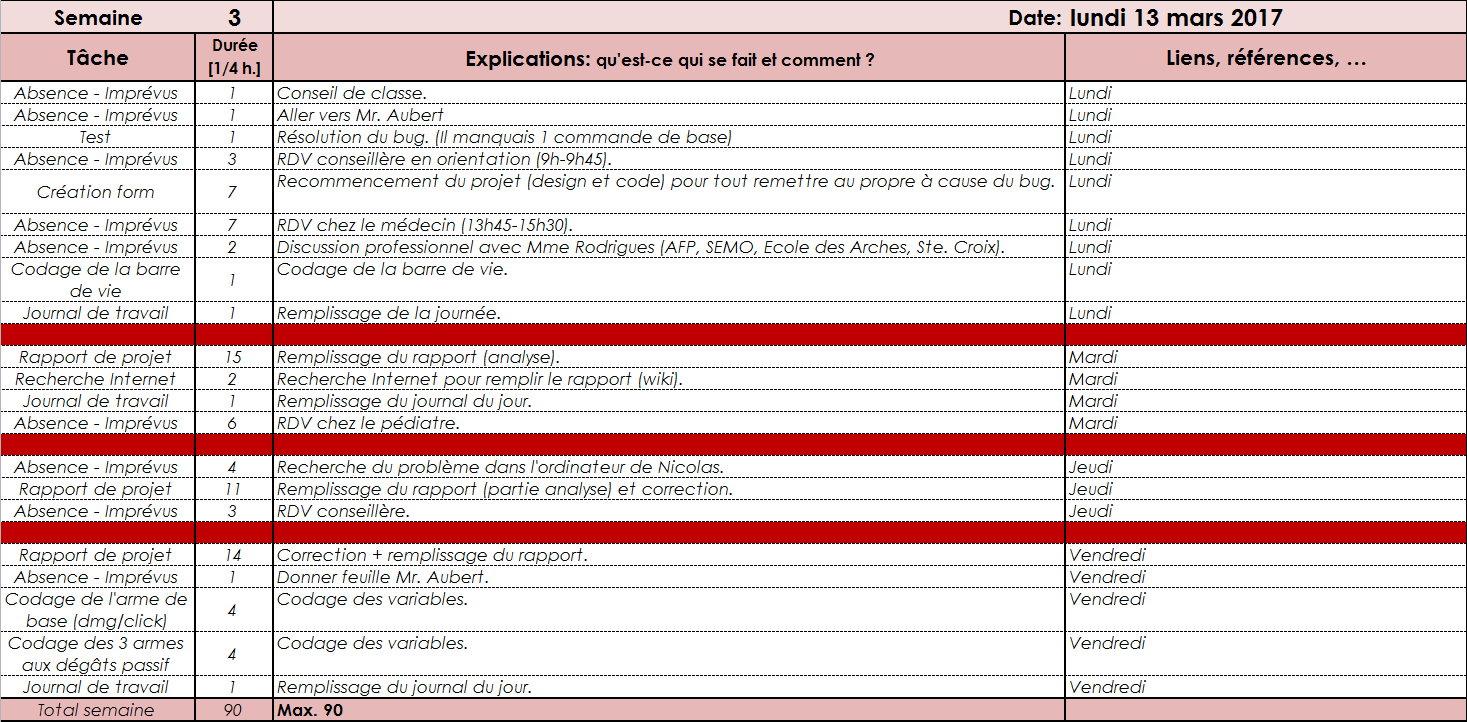
Dès le début du projet, je ne respectais pas le planning. Je décidais sur le moment de ce que je voulais faire et je ne regardais même pas ce que je devais faire durant la journée et c’était une erreur de ma part. J’ai également fais une erreur sur le fait de : Je commence une tâche et si je n’arrive pas à la finir, j’en fais une autres. Si je devais refaire ce projet, je suivrais à la lettre mon planning et finirais les tâches convenablement.

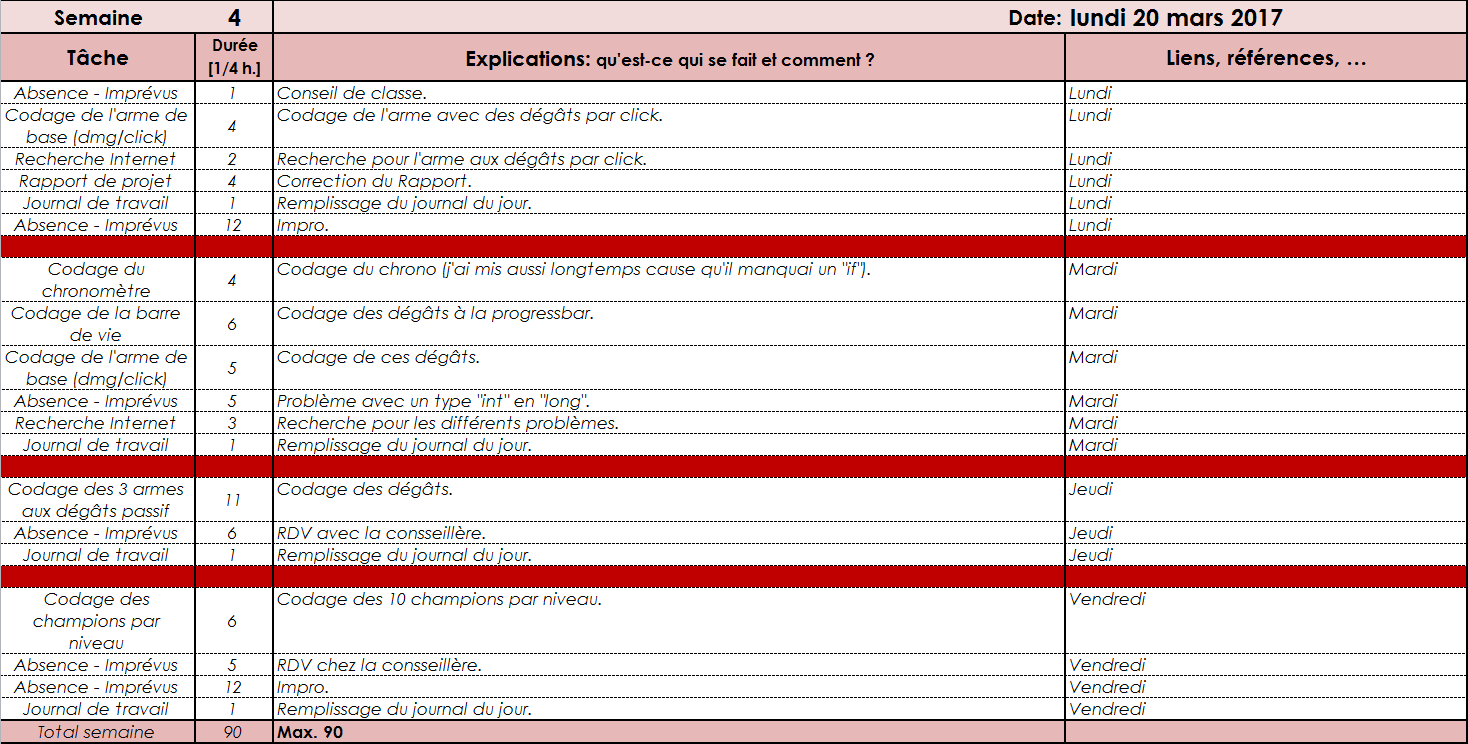
Ce projet m’a appris plusieurs choses, de une : Respecté son planning. De 2 : Finir ce qu’on a commencé. De 3 : L’utilisation des switch. Et de 4 : L’utilisation d’une progressbar.

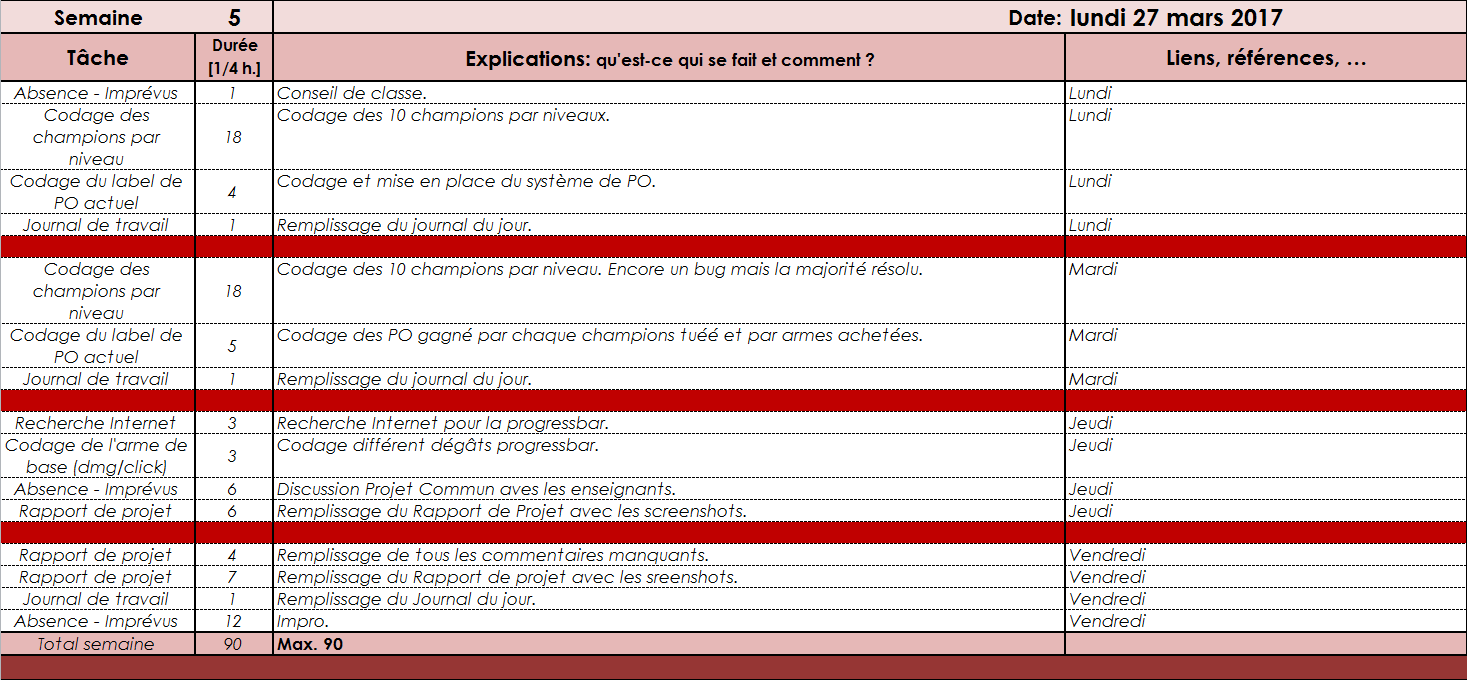
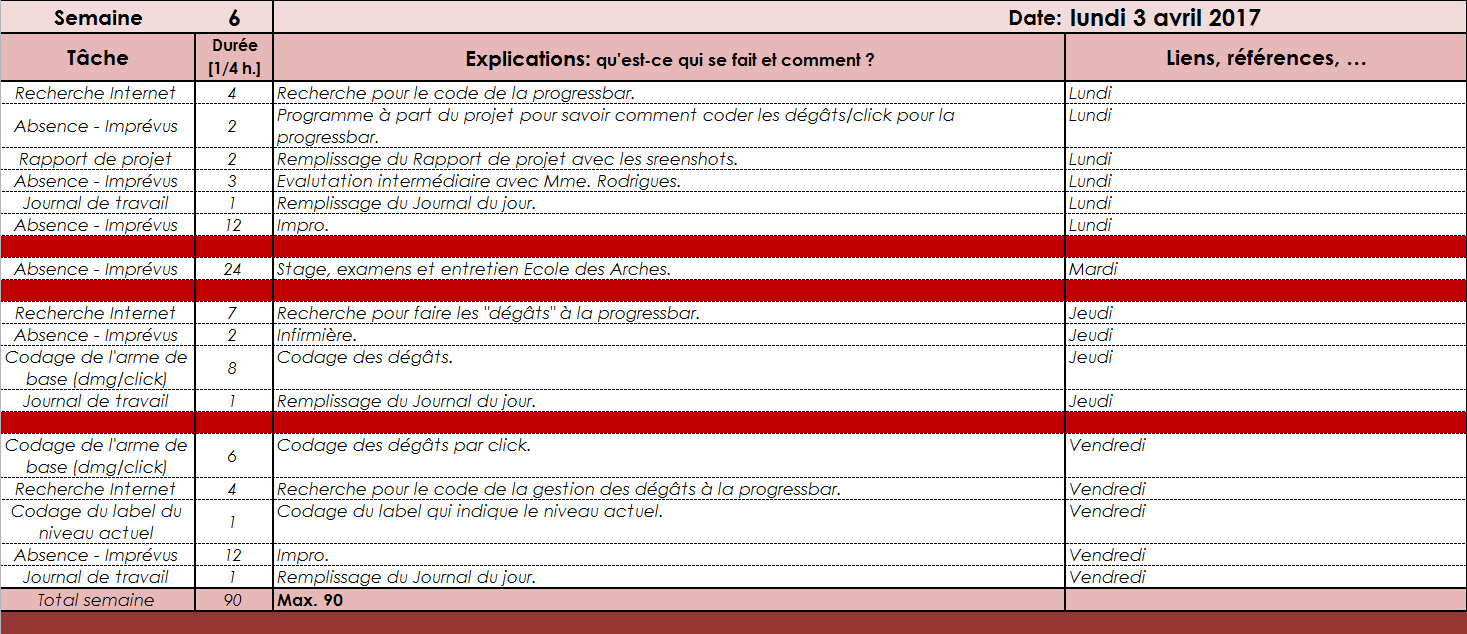
# Divers

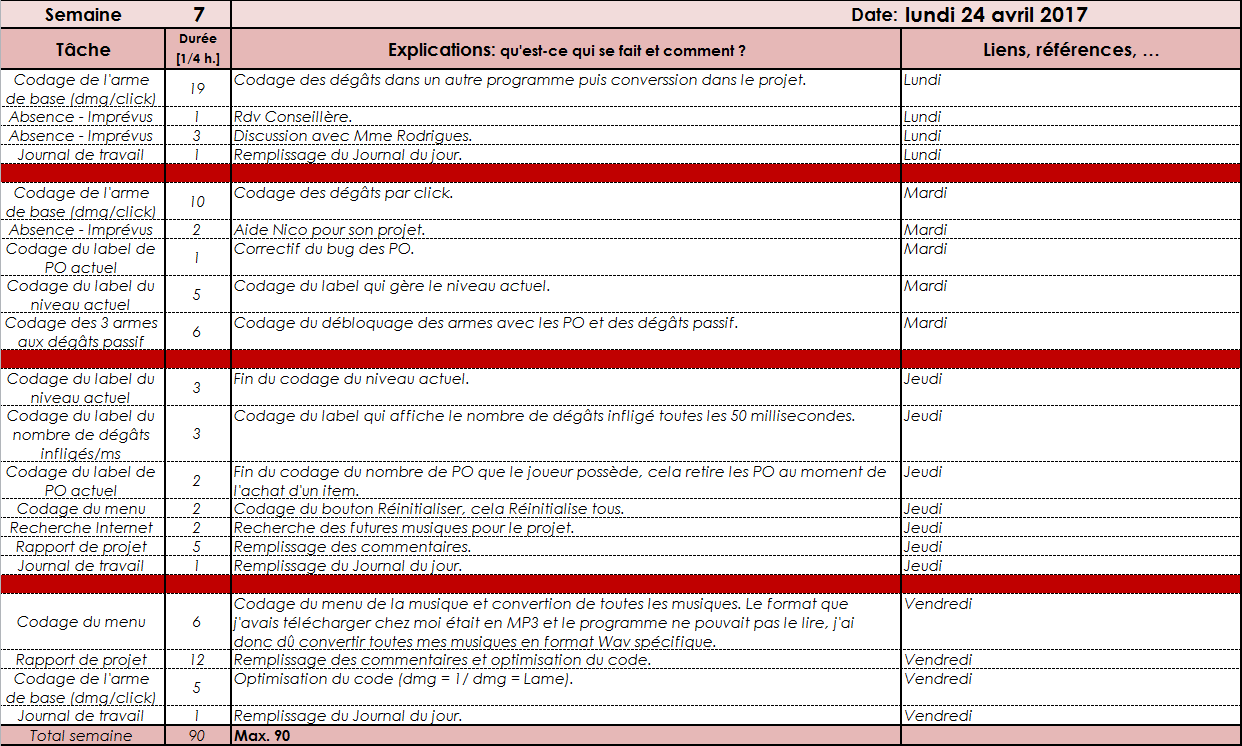
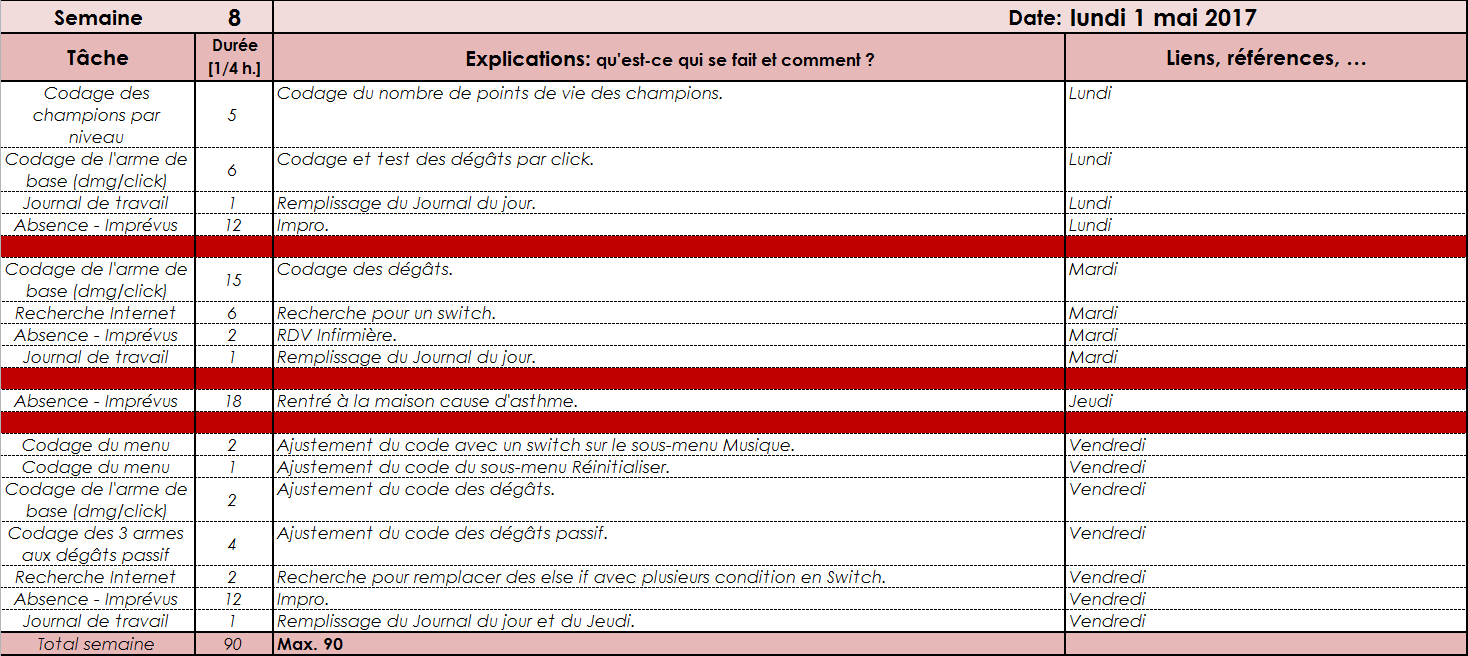
## Journal de travail

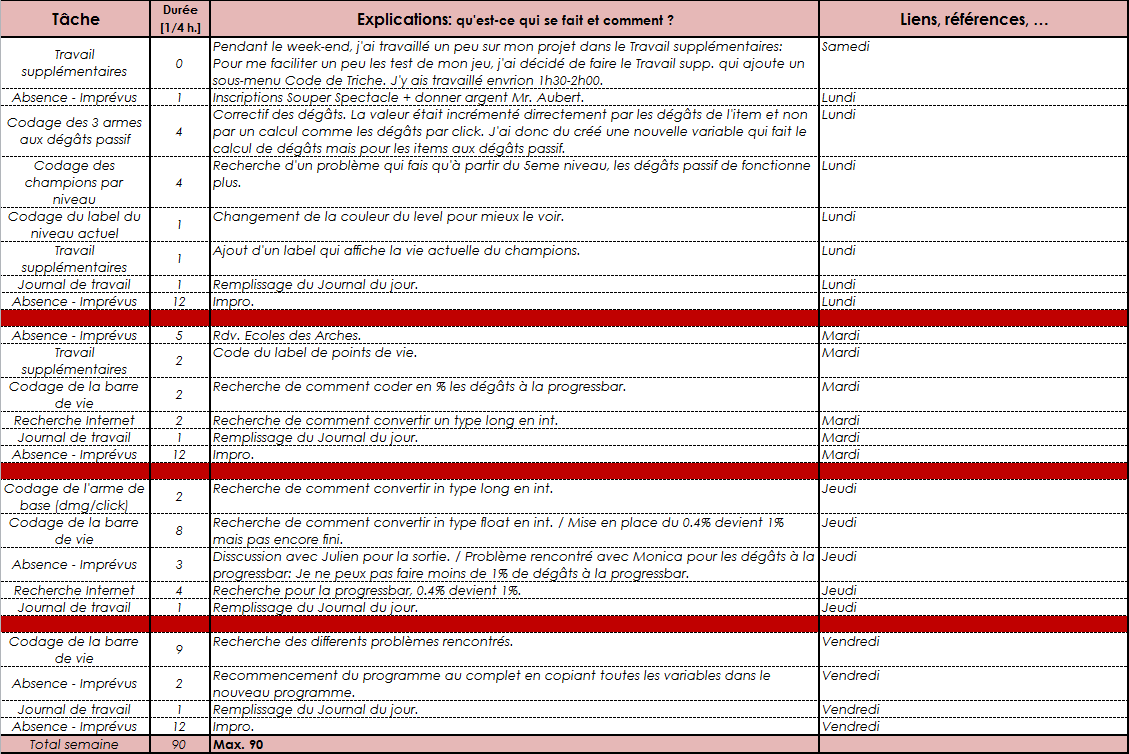
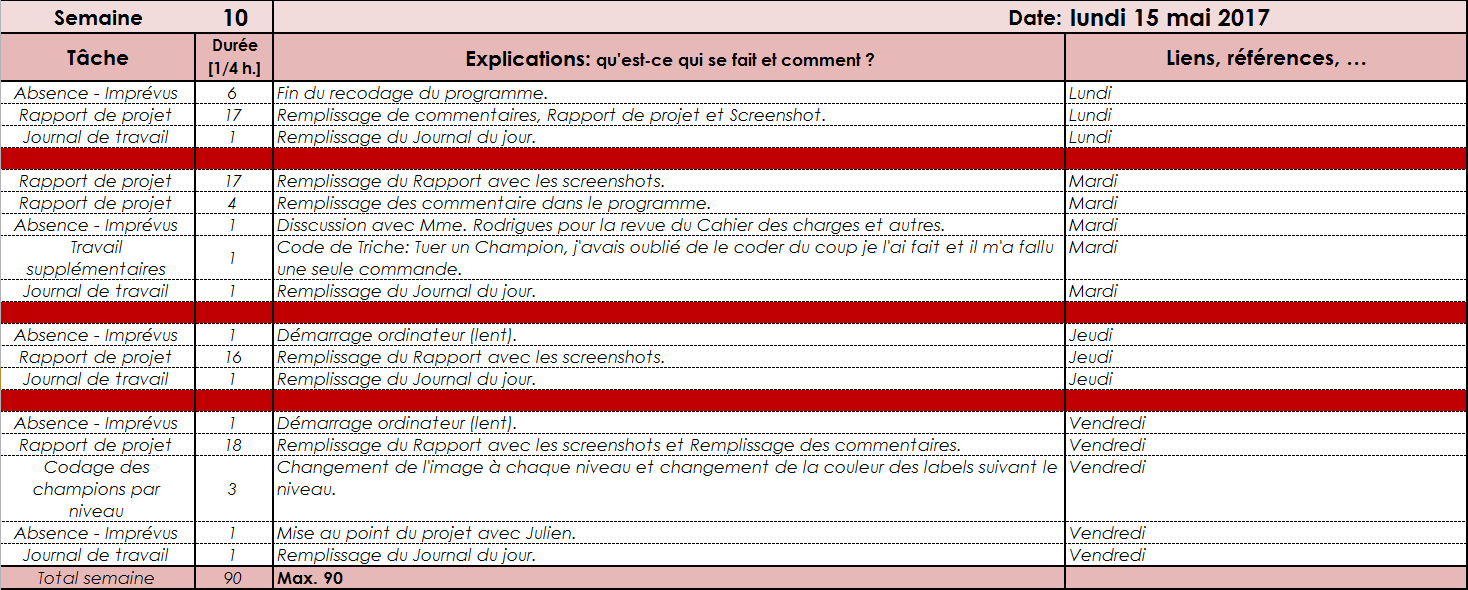
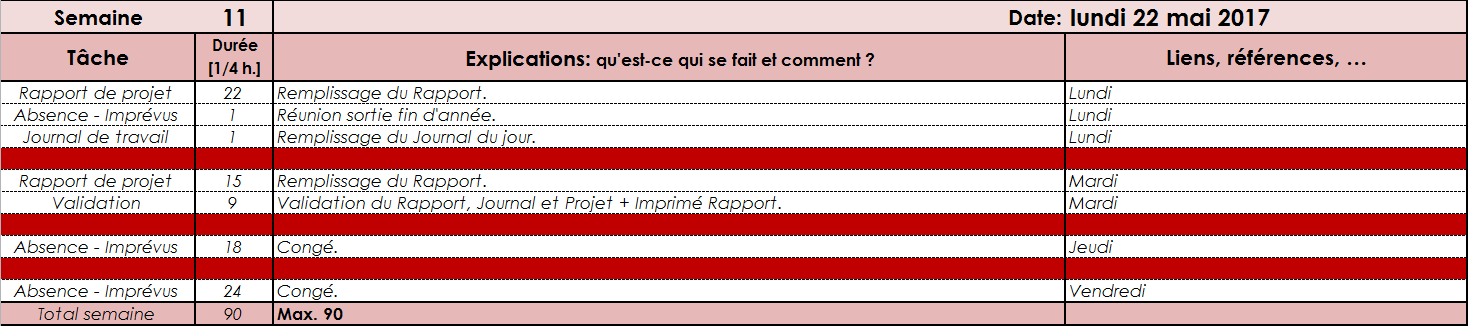








****

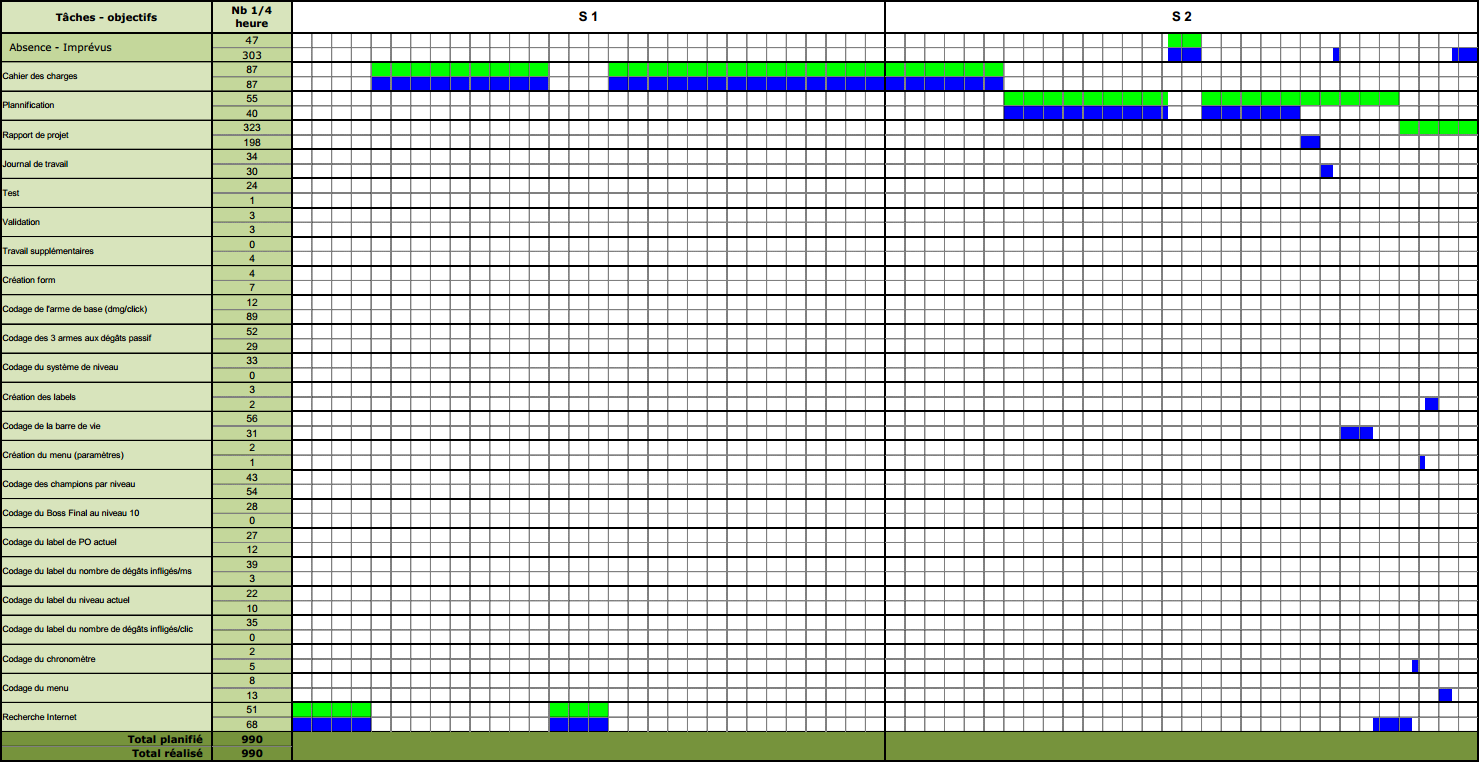
****

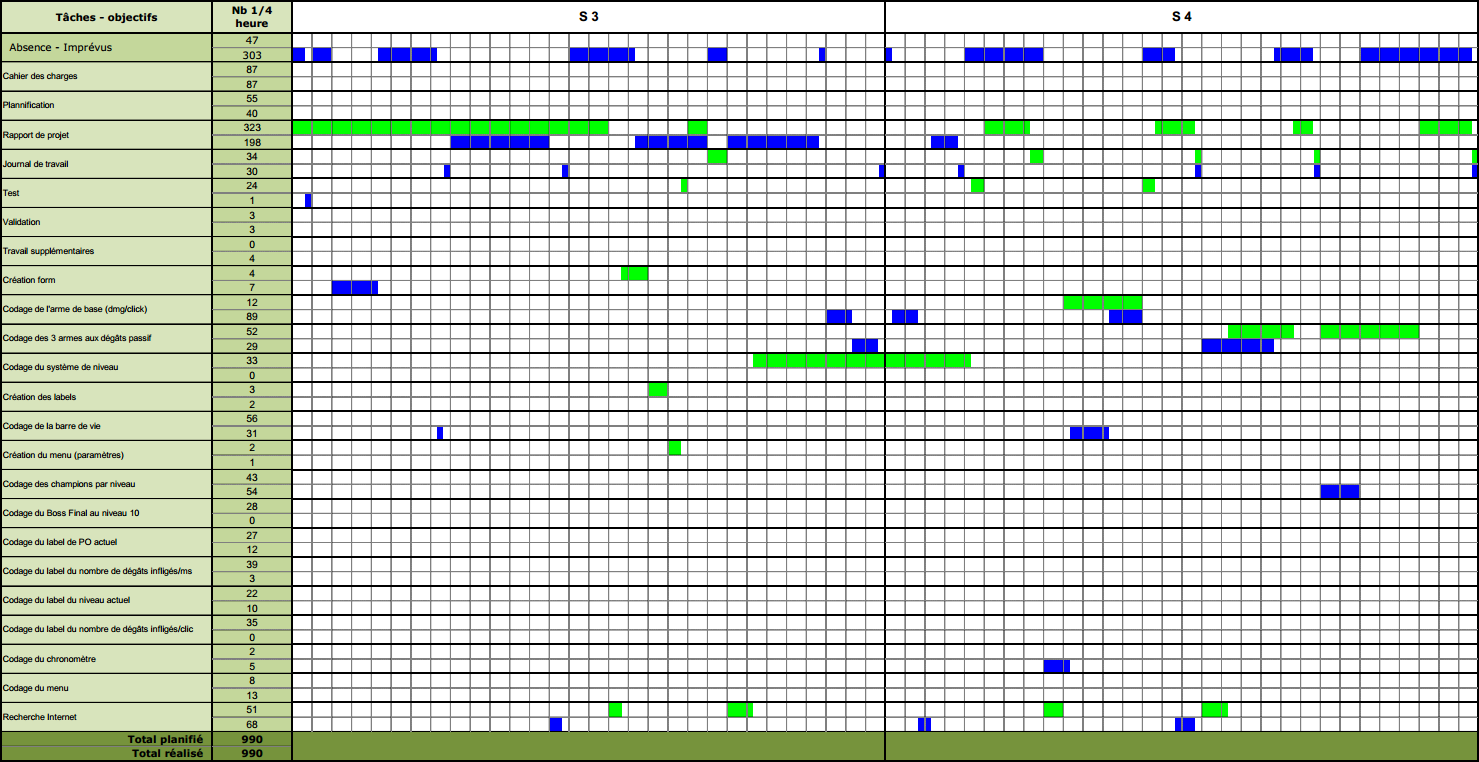
## Webographie

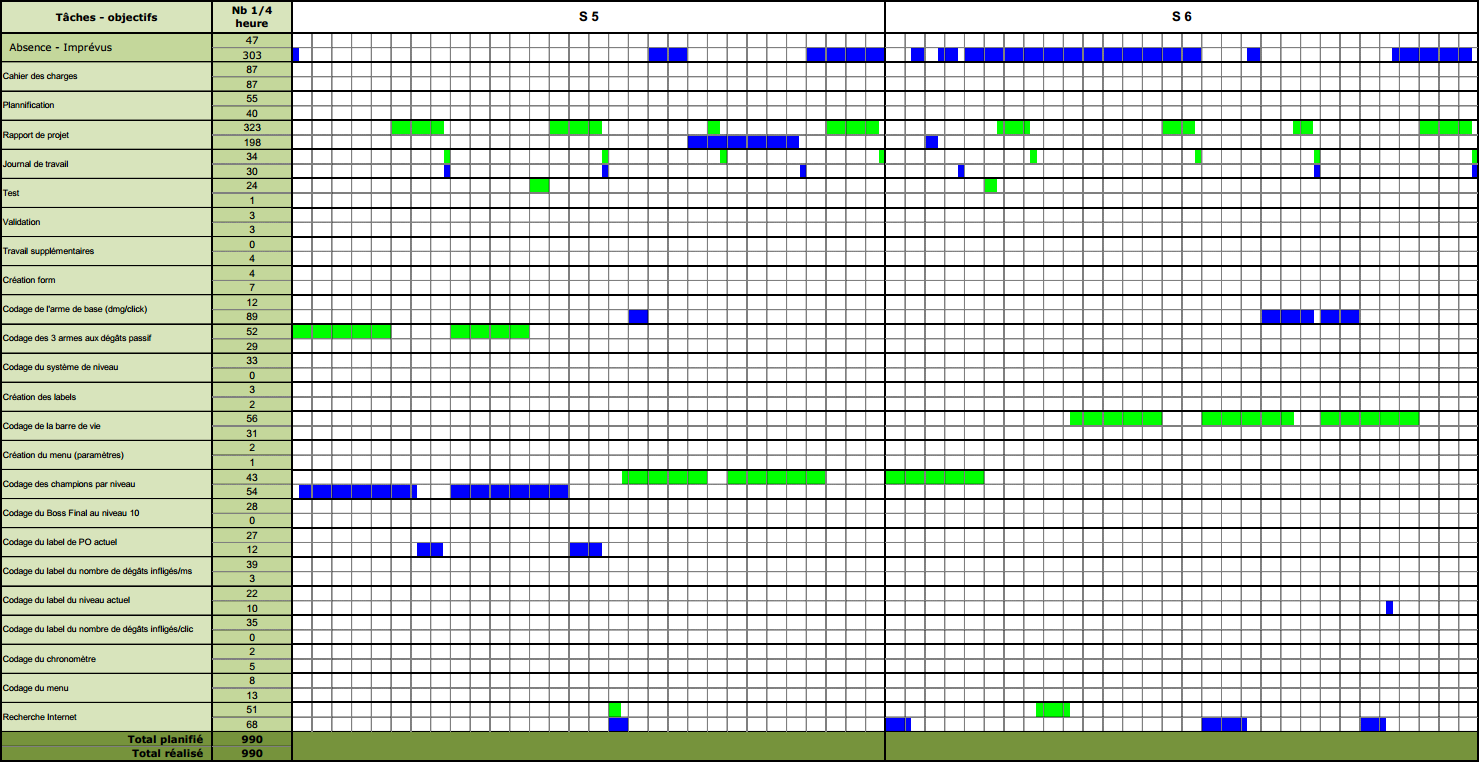
* <http://stackoverflow.com/questions/12126889/how-to-use-winforms-progress-bar>
* <https://social.msdn.microsoft.com/Forums/en-US/b898bce6-8b53-4916-ab48-95f3c514cba6/how-to-reset-progressbar?forum=vssmartdevicesvbcs>
* <https://msdn.microsoft.com/en-us/library/06tc147t.aspx>
* <http://www.convertdatatypes.com/Convert-long-to-int-in-CSharp.html>

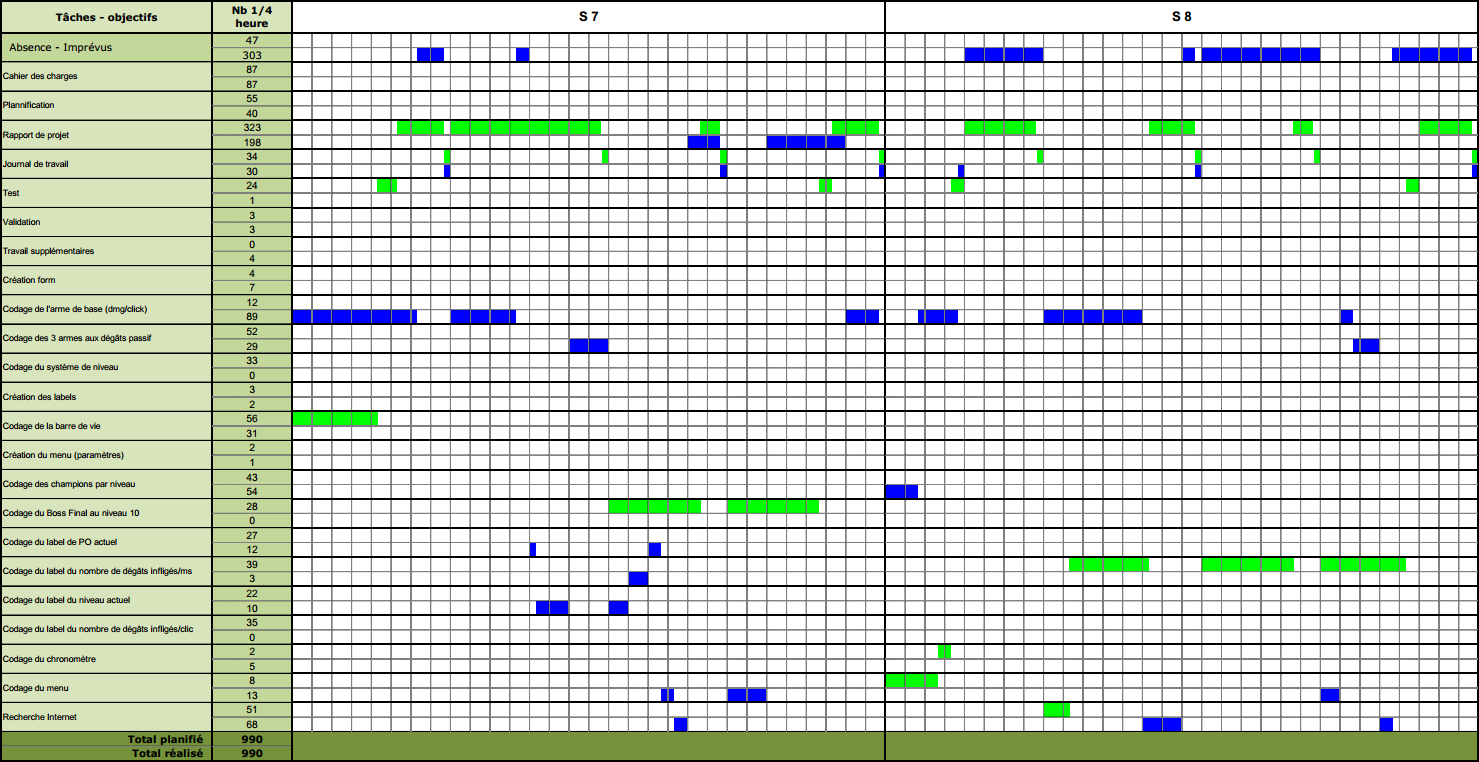
# Annexes

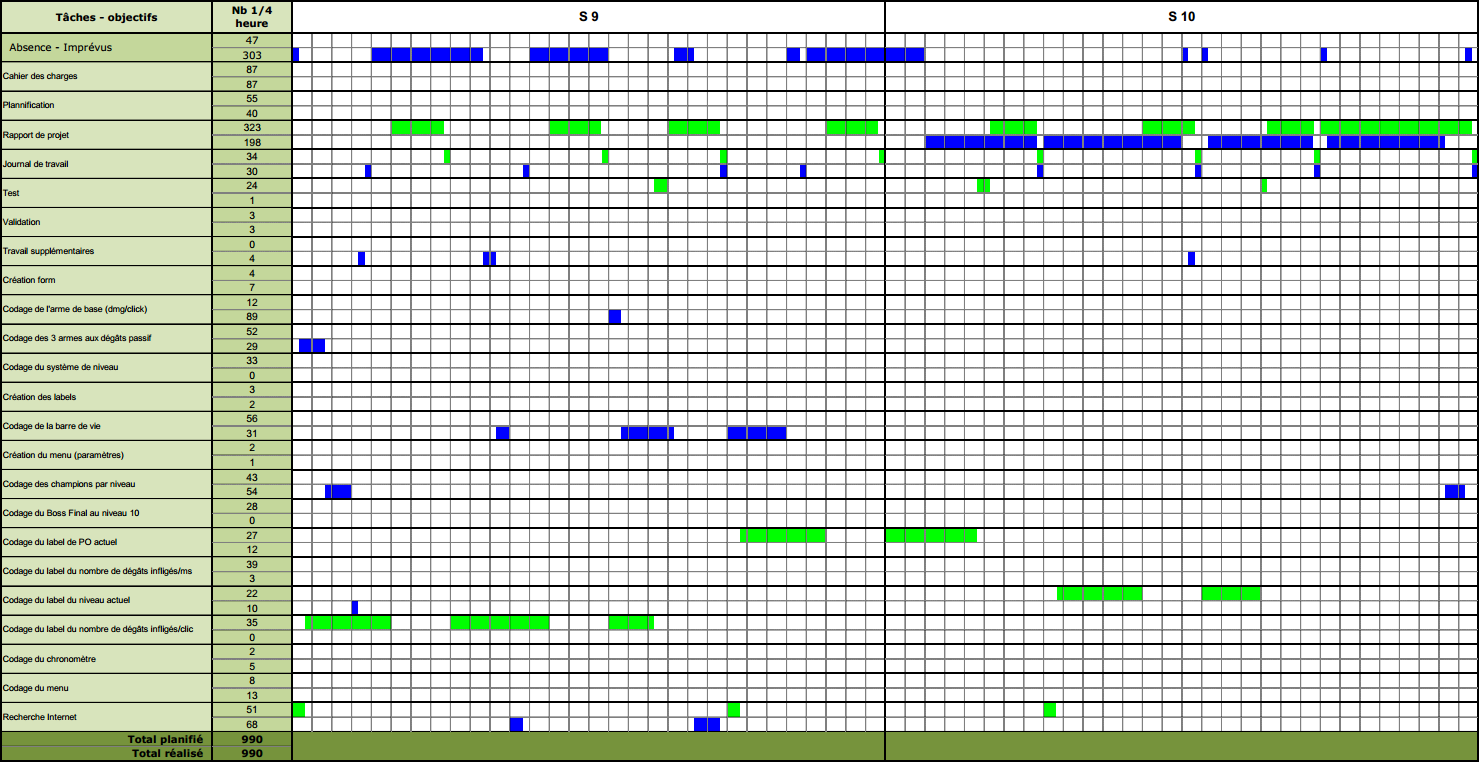
## Planification Détaillé

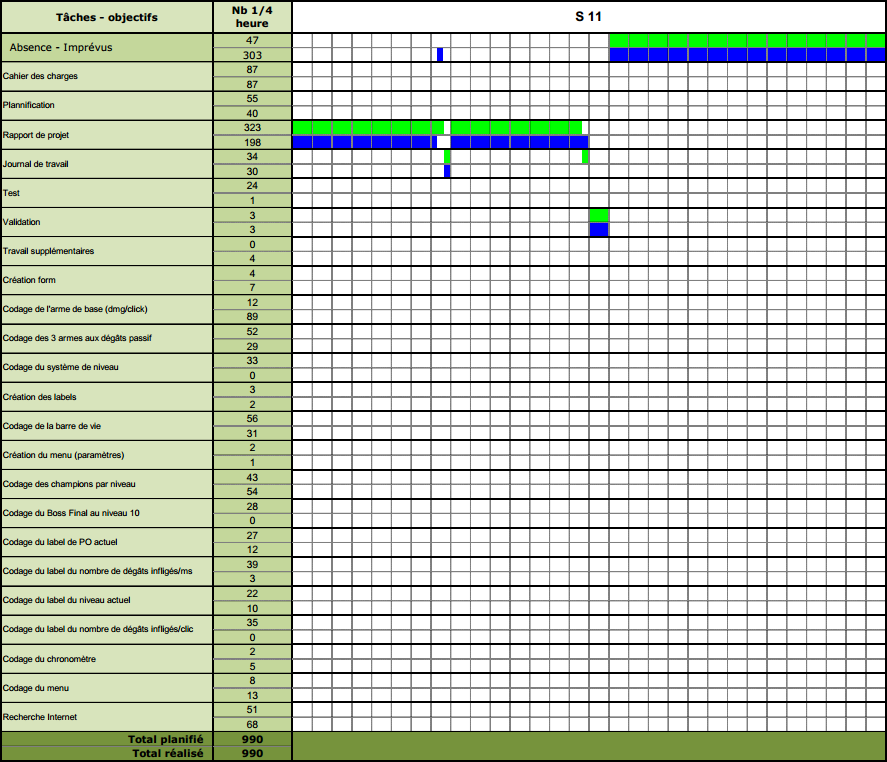


Le point : Codage du système de niveau n’a jamais été utilisé. Je pensais que ça allait être différents que du point : Codage des champions par niveau.

Durant la semaine 6, j’étais absent le mardi. J’avais un stage à l’Ecole des Arches.

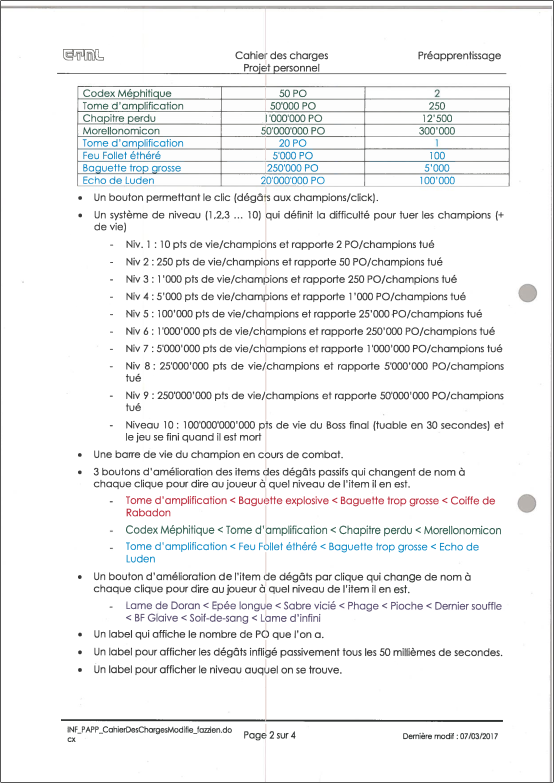
Pendant ces 2 semaines ci, je devais finir le codage de l’armes aux dégâts par click pour pouvoir faire le reste. Si non le reste du jeu ne fonctionnait pas correctement.

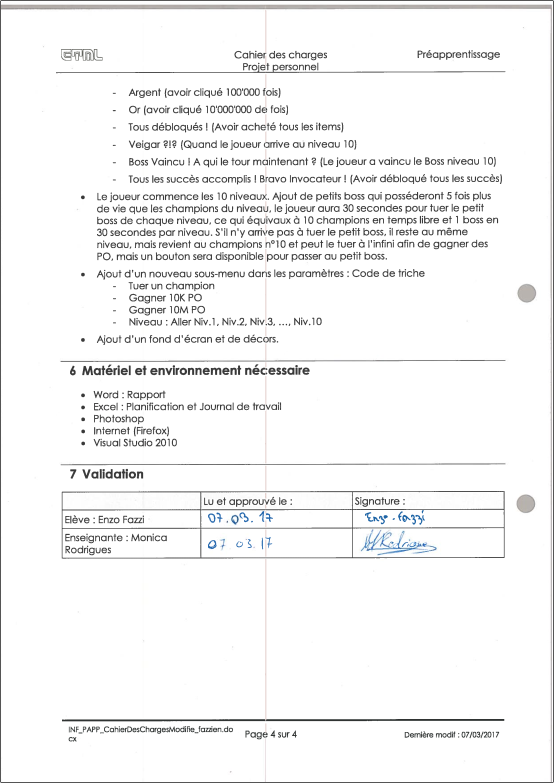
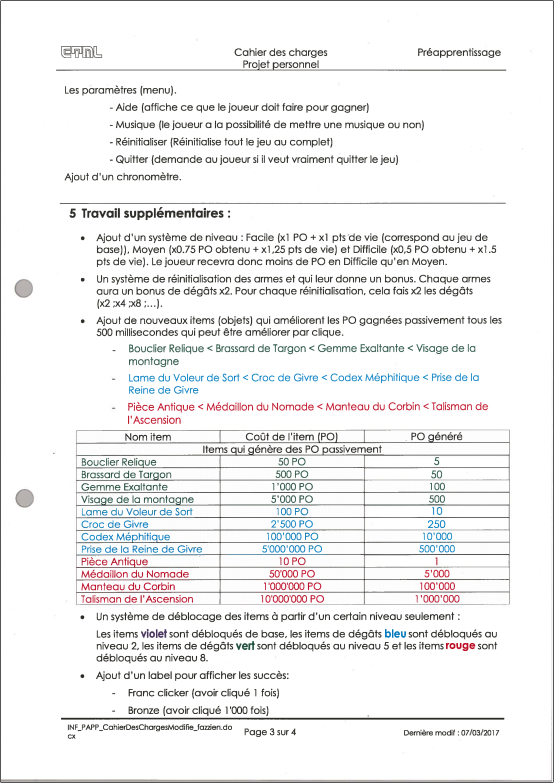
Dans la planification initiale, j’ai mis comme point : Codage du label du nombre de dégâts infligés/click, alors que je ne l’ai pas mis dans mon cahier des charges. Je n’ai donc jamais mis ce label.



## Cahier des charges

## 





## Documents Annexes

[Cahier des Charges](INF_PAPP_CahierDesCharges_fazzien.docx)

[Planification Initiale](INF_PAPP_PlanifInitiale_fazzien.xlsm)

[Planification Détaillée](INF_PAPP_PlanifDétail-JDT_fazzien.xlsm)

1. PO = Pièce d’Or [↑](#footnote-ref-1)