

« Chifoumi », le retour

1 Description du jeu dans sa version initiale

Le paragraphe qui suit décrit le jeu selon deux points de vues :

- Comportement du jeu
- Comportement des éléments graphiques de l'interface lors du redimensionnement de la fenêtre

1.1 Comportement du jeu (aspects fonctionnels et graphiques)

- L'utilisateur joue seul contre la machine
- Lorsque l'utilisateur lance le programme, le plateau se remplit avec les éléments du jeu et se met en état initial (cf. [Figure 1](#)) :
 - Partie haute : des zones de visualisation, où l'utilisateur peut consulter les informations qui le concernent (zone intitulée « **Vous** »), à savoir son score, le dernier coup qu'il a joué, ainsi que les informations concernant la machine, à savoir le score de la machine et le dernier coup qu'elle a joué.
 - Partie basse : des zones d'interaction, où l'utilisateur peut jouer un nouveau coup, ou bien commencer une nouvelle partie.
 - A l'état initial, le bouton « Nouvelle partie » est activé et détient le focus, les boutons contenant les figures (ciseau, papier, pierre) sont inactifs¹.
- Lorsque la partie en cours (cf. [Figure 2](#)), le déroulement d'une manche est le suivant :
 - Les boutons-figures sont actifs. Les nom et score du joueur sont de couleur bleue² : c'est donc son tour de jouer. Il joue en sélectionnant la figure qu'il souhaite tirer (pierre, papier, ciseau). La figure choisie s'affiche dans la zone d'affichage correspondante.
 - Les nom et score du joueur machine sont de couleur bleue : c'est donc son tour de jouer. Il choisit aussi une figure. Il s'agit d'un choix aléatoire. La figure est affichée dans la zone adéquate.
 - Les scores sont mis à jour : + 1 point en fonction du coup de chacun (pierre gagne à ciseau, ciseau gagne à papier, papier gagne à pierre). Les scores restent inchangés en cas de coup identique. Une fois recalculés, les scores de chaque joueur sont affichés dans les zones prévues.
 - A la fin de la manche, le joueur peut jouer une autre manche (en choisissant une figure), recommencer la partie (les scores sont remis à 0 : cf.), ou bien arrêter le jeu (fermeture de la fenêtre). Il n'y a pas de gagnant.

Lorsque la partie est en cours, le bouton « Nouvelle Partie » est toujours actif et est le bouton par défaut³. En le cliquant, les scores sont remis et réaffichés à 0, les zones d'affichage des derniers coups joués effacées, les figures sont actives (cf. [Figure 3](#) : L'utilisateur a demandé une Nouvelle Partie [Figure 3](#)).

Les Menus Fichier, Aide et le bouton nouvelle partie sont accessibles via des raccourcis clavier.

¹ Pourquoi ? : Citer la règle ergonomique qui a motivé ce choix.

² Pourquoi ? Citer la règle ergonomique qui a motivé ce choix.

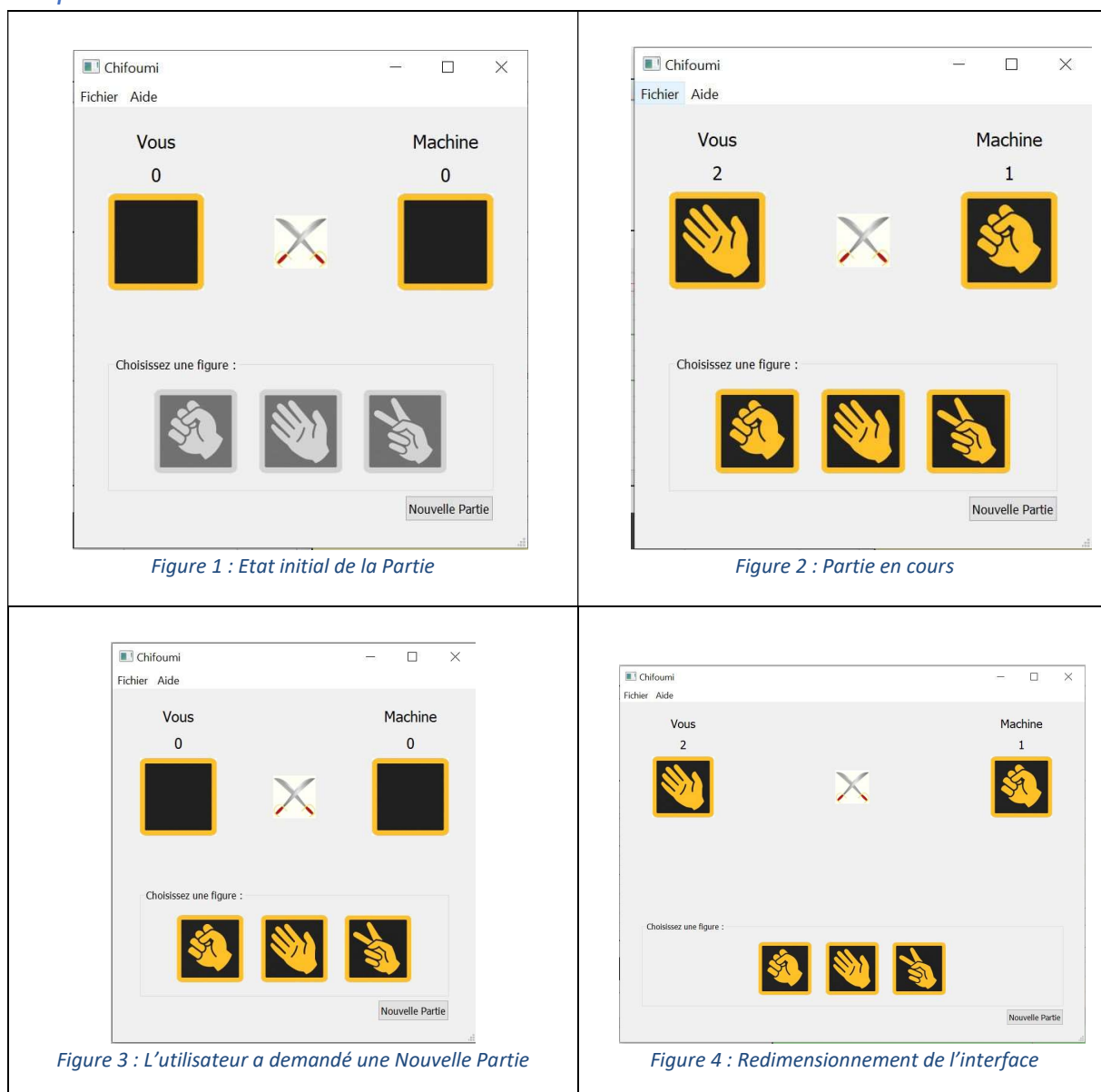
³ Bouton par défaut = c'est lui qui est activé lorsque l'utilisateur presse sur la touche 'Entrée'.

1.2 Comportement de l'interface lors du redimensionnement de la fenêtre

Le premier comportement souhaité est le suivant :

- Interface dans l'état initial de l'application : cf. [Figure 1](#)
- Lorsque l'utilisateur agrandit la fenêtre alors que la partie n'a pas encore commencé (cf. [Figure 4](#) *Erreur ! Source du renvoi introuvable.*) :
 - Les éléments composant le bloc 'Coup-Score' de chaque joueur restent groupés, les éléments du bloc 'Boutons' restent groupés ⁴, les blocs 'Coup-Score' du joueur et de la machine se séparent.
 - Le bouton NullePartie : ⁵ Reste proche des autres boutons d'action, mais pas sur la même ligne ⁶ Est aligné à droite avec les blocs 'Coup-Score' de la machine et 'Boutons' ⁶
 - Les espaces libres générés par le grandissement de la fenêtre se rajoutent de manière équitable à gauche et à droite des blocs 'Coups-Scores' et 'Boutons'.

1.3 Copies d'écran



⁴ Pourquoi ? Citer la règle ergonomique qui a motivé ce choix.

⁵ Pourquoi ? Citer la règle ergonomique qui a motivé ce choix.

⁶ Pourquoi ? Citer la règle ergonomique qui a motivé ce choix.

Lorsque ce comportement sera obtenu, il sera modifié / simplifié : la fenêtre restera bloquée sur sa taille initiale (cf. [Figure 1](#) et [Figure 2](#)).

2 Développement incrémental

L'application sera développée de manière incrémentale. Chaque version intégrera une amélioration, soit fonctionnelle (= 1 nouvelle fonctionnalité), soit liée à l'organisation du code produit.

Chaque version est décrite dans le tableau suivant (cf. § 2.1) et illustrations associées (cf. § 2.2). Un graphe de précedence entre les versions montre les différents chemins permettant d'atteindre la v10 (cf. § 2.3).

2.1 Table des versions

Version	Description fonctionnelle	Références figures
v0	Sources C++ de la classe Chifoumi dans un projet Non Qt	
v1	Version décrite dans le paragraphe §1 précédent. La barre de menus est inactive.	
v2	L'application offre les mêmes fonctionnalités que la version v1. Le code source est structuré selon le modèle MVC. → amélioration de l'organisation du code	
v3	Le menu Fichier >> Quitter permet d'arrêter l'application Le menu Aide >> A propos de... ouvre une Boîte de Message donnant des informations sur : <ul style="list-style-type: none"> la version de l'application la date de création les auteurs 	Figure 5 Figure 6, Figure 7
v4	Le jeu se joue en X points, 5 par défaut. Le premier joueur dont le score a atteint ce maximum a gagné.	Figure 8, Figure 9
v5	Le jeu se joue en temps limité, 30 secondes par défaut. A l'issue du temps imparti, il y a un joueur gagnant, ou bien il y a égalité entre les 2 joueurs.	De Figure 10 à Figure 14
v6	Le menu Fichier >> Paramétrer... ouvre une fenêtre de Dialogue dans laquelle l'utilisateur peut modifier différents paramètres avant de commencer la partie : <ul style="list-style-type: none"> son nom, le nombre max. de points le temps max. à jouer. Une fois la partie commencée, il n'est plus possible de modifier les paramètres.	A venir...
v7	v6 + Le nom du joueur est enregistré dans une base de données. v6 + ... Dès son lancement, le programme demande à l'utilisateur de s'authentifier (utilisateur + mot de passe). L'application vérifie en base de données que cet utilisateur existe, et si c'est le cas, l'utilisateur a alors accès à l'interface du jeu codée en v6. On ne vous demande pas dans cette version de pouvoir insérer à partir de l'application un nouveau joueur qui ne serait pas répertorié dans la base de données	A venir... Cf. présentation en cours
v8	v7 + A la fin de la partie (cf. temps max de jeu), le système enregistre dans la base de données le résultat de la partie en s'appuyant sur certains éléments recueillis, par exemple : Horodatage, nomJoueurHumain, scoreJoueurHumain, joueurMachine, scoreJoueurMachine	A venir... Cf. présentation en cours
v9	Le menu Fichier >> Réinitialiser... permet de vider la base de données. Une boîte de message indique que l'action s'est bien terminée.	A venir... Cf. présentation en cours

	Un nouveau menu est ajouté à l'interface Fichier >> Résultats... permettant à un utilisateur authentifié d'obtenir des informations sur les 10 meilleurs scores obtenus contre la machine par les joueurs qui ont fait des parties de chifoumi avec cette application particulière	
v10	<p>Ajout d'extensions jugées utiles par le groupe de projet</p> <p>A partir du menu Fichier >> Résultats..., l'application ouvre une fenêtre de Dialogue dans laquelle elle affiche tous les résultats enregistrés, classés par ordre alphabétique croissant sur le nom du joueur :</p> <p>nomJoueurHumain1 :</p> <ul style="list-style-type: none"> - [Horodatage optionnel], scoreJoueurHumain, joueurMachine, scoreJoueurMachine - [Horodatage optionnel], scoreJoueurHumain, joueurMachine, scoreJoueurMachine - ... <p>nomJoueurHumain2 :</p> <ul style="list-style-type: none"> - [Horodatage optionnel], scoreJoueurHumain, joueurMachine, scoreJoueurMachine - [Horodatage optionnel], scoreJoueurHumain, joueurMachine, scoreJoueurMachine - ... 	A venir...

2.2 Copies d'écran

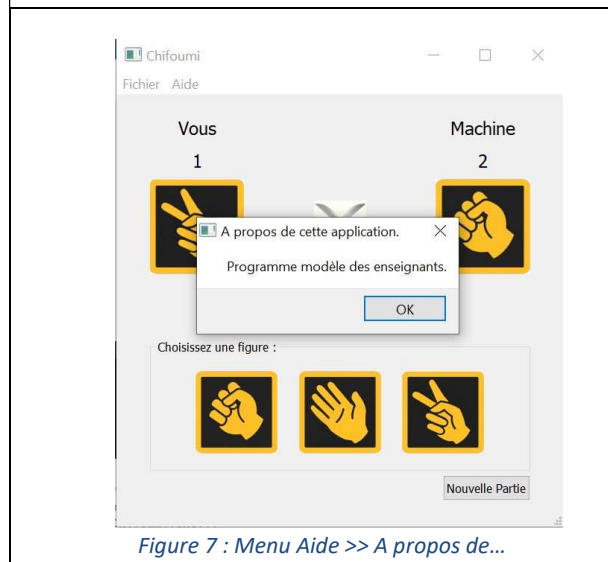
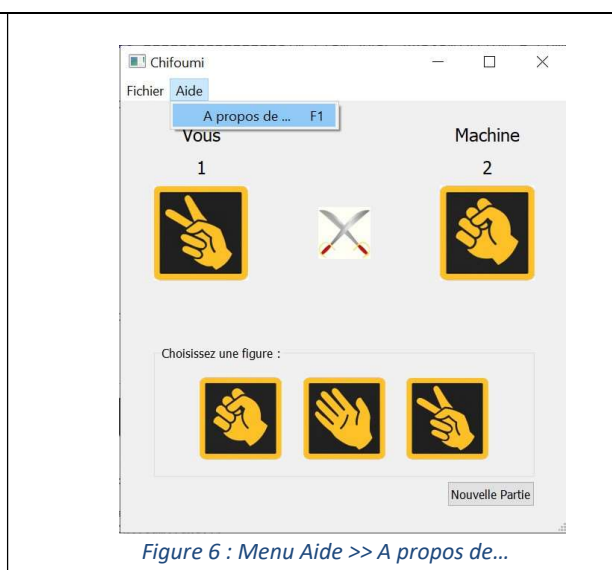
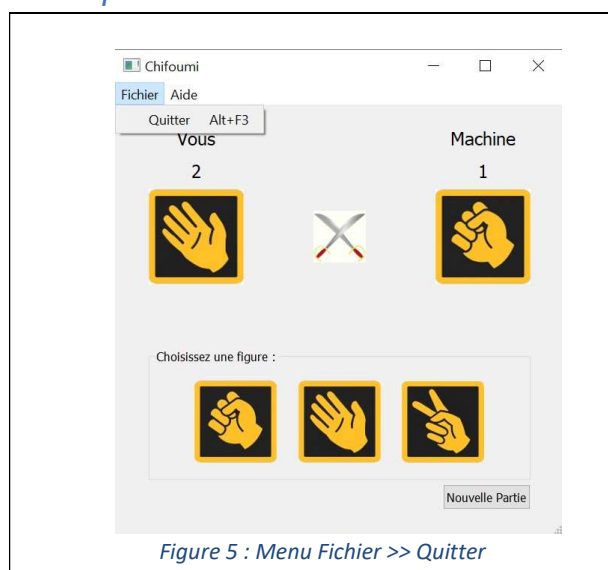




Figure 9 : Partie en X points, fin

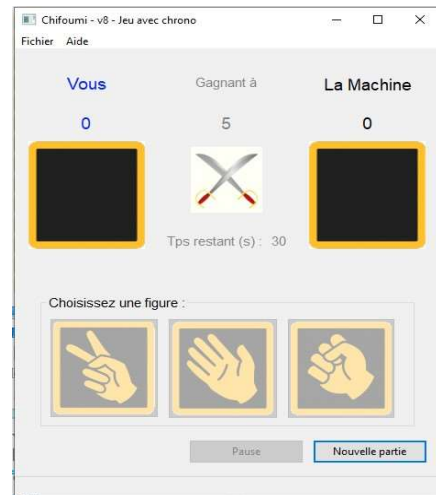


Figure 10 : Partie en temps limité, début



Figure 11 : Partie en temps limité, en cours

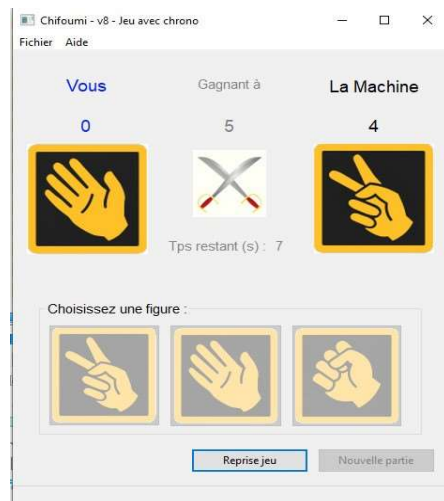


Figure 12 : Partie en temps limité, en pause

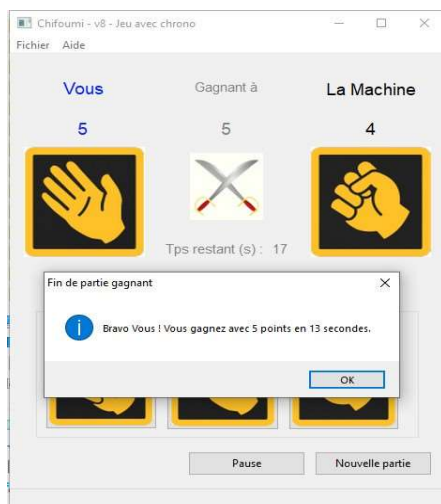


Figure 13 : Partie en temps limité, fin avec gagnant

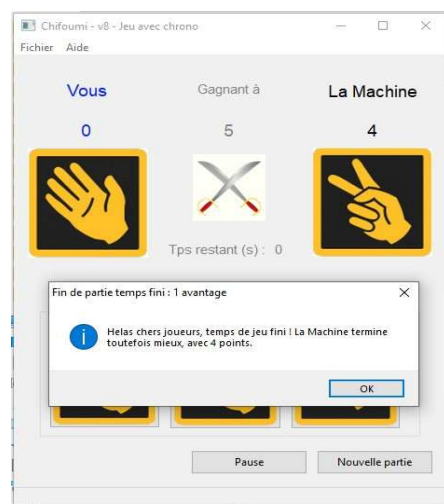


Figure 14 : Partie en temps limité, fin sans gagnant

2.3 Ordre de développement des versions

Vous avez une certaine liberté dans le choix de l'ordre de développement des versions. Le graphe de précedence (cf. Figure 15) montre les possibles chemins.

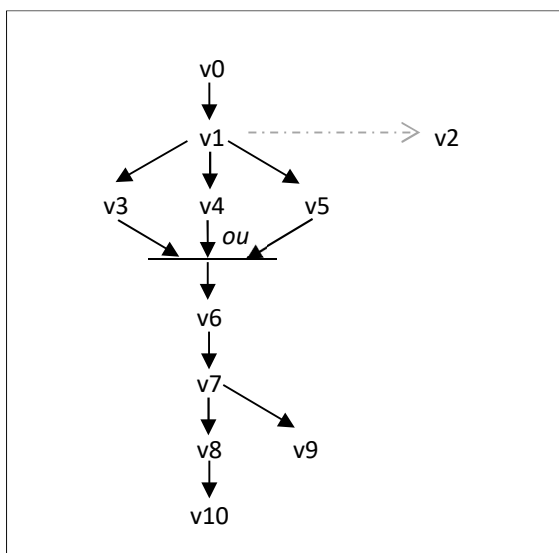


Figure 15 : Graphe de précédence des versions

La version v2 est particulière : elle peut être mise en œuvre à **n'importe quel moment**. Au début (après la v1), c'est mieux, car la suite du développement s'en trouve facilitée. Cependant, les équipes peuvent choisir le moment de la mettre en œuvre.

La raison de cette particularité : elle ne concerne pas une nouvelle fonctionnalité de l'application, mais s'attache à proposer la réorganisation du code de l'application dans un double objectif :

- Que la structure du code produit soit fidèle aux documents d'analyse. Il s'agit de faire de même qu'en programmation procédurale, où le code produit doit être fidèle au document de conception (= l'algorithme).
- Diminuer le couplage (l'interdépendance) entre le code traitant de la partie fonctionnelle de l'application et le code focalisé sur l'interface de l'application. Pour faciliter l'évolutivité et la maintenance de l'application.

3 Ressources à disposition

Intitulé	Description – Localisation
chifoumiAnalyseEtConception.doc	Premiers éléments du dossier d'analyse et de conception de l'application. Il contient : <ul style="list-style-type: none"> - Un diagramme des cas d'Utilisation - Un scénario - Une brève description de la classe Chifoumi Vous devrez compléter ce document au fur et à mesure de l'avancement de votre projet.
chifoumi.h chifoumi.cpp main.cpp	Classe Chifoumi, à compléter et tester avec le programme de test (main.cpp) fourni.
dossier images	Dossier contenant les images que nous avons utilisées pour réaliser l'application. Vous pouvez en choisir d'autres, en gardant à l'esprit qu'il y a sans doute autre chose à faire que de passer du temps à chercher des figures sur Internet....
fonction int randomMinMax()	Une fonction aléatoire, avec un exemple d'utilisation. Elle sera utile pour faire jouer la machine.

```

1  #include <iostream>
2  #include <cstdlib>      // pour le tirage aléatoire
3  #include <ctime>        // pour la date aléatoire
4
5  using namespace std;
6
7  /** sous-programmes utilisés **/
8  int randomMinMax(int min, int max);
9  /* pré-condition : min < max ;
10   Le nbre aléatoire généré est compris entre [min, max[ */
11
12
13  /** Programme principal
14  -----*/
15
16  int main()
17  {
18      // AVANT d'UTILISER la fonction : initialisation du générateur de nombres aléatoires rand
19      srand(time(NULL));
20
21      // UTILISATION de la fonction
22      int nbreAleatoire;
23      nbreAleatoire = randomMinMax(1,4); // Le nbre généré
24                                         // appartient à l'intervalle [1..4[
25
26      cout << nbreAleatoire;
27
28      return 0;
29
30  /** CORPS des sous-programmes utilisés
31  -----*/
32
33  int randomMinMax(int min, int max){
34      /* pré-condition : min < max ;
35       Le nbre aléatoire est compris entre [min, max[ */
36      return rand()%(max-min) + min;
37  }
38

```

Template pour .gitignore

Fichier template à utiliser pour versionner (Git et Github) les éléments pertinents du projet. A venir...

4 Equipes

Vous travaillerez par équipes de 2 ou 3 étudiants, à l'intérieur d'un même groupe de TP.

Groupe de TP	TP1	TP2	TP3	TP4	TP5
Effectif	12	10	12	10	15
21 équipes	4*3	2*3 + 2*2	4*3	2*3 + 2*2	5*3

Les équipes constituées s'inscriront sur eLearn selon la procédure habituelle (des saé 1.1 et 1.2), et fourniront l'url vers le dépôt Github qui sera utilisé dans la saé. Le dépôt sera public et aura pour nom « Chifoumi ».

5 Durée et planning des séances de la Saé

La saé dure 9 semaines.

Au total, elle prévoit 3 séances de TD et 13 de TP.

Attention, le planning est susceptible de varier un peu afin de dégager un peu plus de temps en dernière semaine pour la partie évaluation.

	Sem. Form.	Sem. Civile	du Lundi	au Samedi	séances par étudiant	
					TD 1.5h	TP 1.5h
	18	4	24/01/2022	29/01/2022	0	0
	19	5	31/01/2022	05/02/2022	0	0
	20	6	07/02/2022	12/02/2022	0	0
	21	7	14/02/2022	19/02/2022	0	0
		8	21/02/2022	26/02/2022		
	22	9	28/02/2022	05/03/2022	0	0
	23	10	07/03/2022	12/03/2022	0	0
	24	11	14/03/2022	19/03/2022	0	0
	25	12	21/03/2022	26/03/2022	0	0
	26	13	28/03/2022	02/04/2022	1	1
	27	14	04/04/2022	09/04/2022	0	1
	28	15	11/04/2022	16/04/2022	0	2
		16	18/04/2022	23/04/2022		
		17	25/04/2022	30/04/2022		
	29	18	02/05/2022	07/05/2022	1	1
	30	19	09/05/2022	14/05/2022	0	1
	31	20	16/05/2022	21/05/2022	0	1
	32	21	23/05/2022	28/05/2022	0	2
	33	22	30/05/2022	04/06/2022	0	2
	34	23	06/06/2022	11/06/2022	1	2
		24	13/06/2022	18/06/2022		

6 Modalités de remise du travail

Chaque version fera l'objet d'un commit sur le dépôt Github. Le commit concernera toujours :

- 1 fichier .pdf, nommé **acChifoumiNometudiant1Nometudiant2_TPx.pdf** contenant l'Analyse & Conception de votre application.
Exemple : acChifoumiCarpentierDagorret_TP6.pdf Il sera complété de dépôt en dépôt
- Le contenu du répertoire contenant les sources de l'application, à l'exception du fichier .pro.user Chaque commit devra obligatoirement contenir ces 2 livrables.

Chaque commit sera identifié en fonction de la version produite. Le détail est donné dans le § **Erreur ! Source du renvoi introuvable.**

6.1 Calendrier des remises

Elles correspondent aux semaines de saé avec présence des enseignants. Les équipes devront montrer qu'elles avancent d'une échéance à l'autre.

6.2 Le dossier d'analyse et de conception

La dernière version du dossier d'analyseEtConception de l'équipe sera déposée sur eLearn, dans la section associée à cette saé.

La page de garde contiendra : le titre de la saé, les auteurs, groupe de TP, année universitaire, année de formation, et **url sur Github** correspondant au lien vers votre dépôt public.

A l'intérieur, le dossier contiendra un sommaire, il y aura au moins autant de chapitres que de versions développées de l'application, et un dernier chapitre avec votre bilan :

- Bilan intermédiaire à compléter après chaque version : ce que vous avez appris, ce qui a été difficile, le TEMPS passé par chaque membre de l'équipe sur la version (analyseConception et code).
- Bilan final à compléter à la fin de la saé : ce que vous avez aimé / pas aimé, ce que vous auriez pu faire mieux (avec le recul), ce qui pourrait être amélioré dans la saé, ainsi que l

➔ Par conséquent, prévoyez d'ores et déjà un tableau dans l'espace partagé du groupe avec le temps de travail passé par version par chaque membre. Le but n'est pas de vous contrôler, mais d'avoir une vision réelle du temps de développement nécessaire pour que nous puissions le comparer au temps prévu par la saé.

6.3.1 Répertoire versionné

Votre répertoire de travail, contenant **des fichiers à versionner**, aura la structure suivante :

- Le répertoire de projet : **chifoumi**
Il contient les fichiers sources du projet, ainsi que le répertoire avec les images
- Le répertoire des fichiers compilés
Son contenu est généré automatiquement par Qt au et à mesure de vos compilations et éditions de liens
- Votre document d'analyse et conception **.pdf** Il évolue au fur et à mesure que votre projet avance
- Un répertoire avec les contenus intermédiaires servant à produire votre document d'analyse, par exemple :
 - Le fichier d'analyse et conception éditable ○ Les fichiers contenant les schémas UML à inclure dans le dossier ○ Des modèles de de fichiers de test ...

Les éléments à versionner sont :

- Le répertoire de projet : **chifoumi** **excepté** le/les fichier(s) **.pro.userXXX**
- Votre document d'analyse et conception **.pdf**

Afin de versionner et déposer sur Github uniquement les fichiers demandés, vous devrez ajouter dans votre répertoire de travail un **fichier .gitignore** qui explicite la liste des fichiers non souhaités.

Utilisez comme base le fichier fourni grâce au lien suivant :

<https://github.com/github/gitignore/blob/main/Qt.gitignore>

Vous devrez le compléter en fonction des documents intermédiaires que vous produisez mais qui ne feront pas l'objet d'un dépôt sur Github.

6.3.2 Dépôt sur Github

Il contiendra :

- Le répertoire de projet : **chifoumi** **excepté** le/les fichier(s) **.pro.userXXX**
- Votre document d'analyse et conception **.pdf**

Chaque version fera l'objet d'un **commit** accompagné du TAG et de la description textuelle fournis ci-dessous, en respectant scrupuleusement la formulation fournie.

Version	TAG	Description
v0	v0	Sources C++ de la classe Chifoumi dans un projet Non Qt
v1	v1	Version simple correspondant au sujet
v2	v2	Mise en œuvre de MVC sur la version vXX ⁷
v3	v3	Ajout Barre Menus Fichier et Aide
v4	v4	Jeu en X points
v5	v5	Jeu en temps limité
v6	v6	Ajout Fichier >> Paramétrage
v7	v7	Enregistrement joueur en BD
v8	v8	Enregistrement résultat partie en BD
v9	v9	Ajout Fichier >> Réinitialiser qui vide la BD Ajout Fichier >> Résultats
v10	v10	Ajout Fichier >> Résultats

⁷ Selon la version à partir de laquelle vous avez élaboré votre v2

7 Modalités d'évaluation

L'évaluation sera basée sur :

- le travail intermédiaire (analyseConception + code) montré par les étudiants aux enseignants durant les séances d'accompagnement prévues, et les réponses aux questions posées par l'enseignant sur ce travail
 - le dossier final rendu par les équipes, et les versions produites
- la correspondance code <-> dossier
- la qualité du code produit
- le respect des règles d'ergonomie identifiées durant le projet
- le mini-oral prévu en fin de projet où chaque équipe présentera son travail aux enseignants.