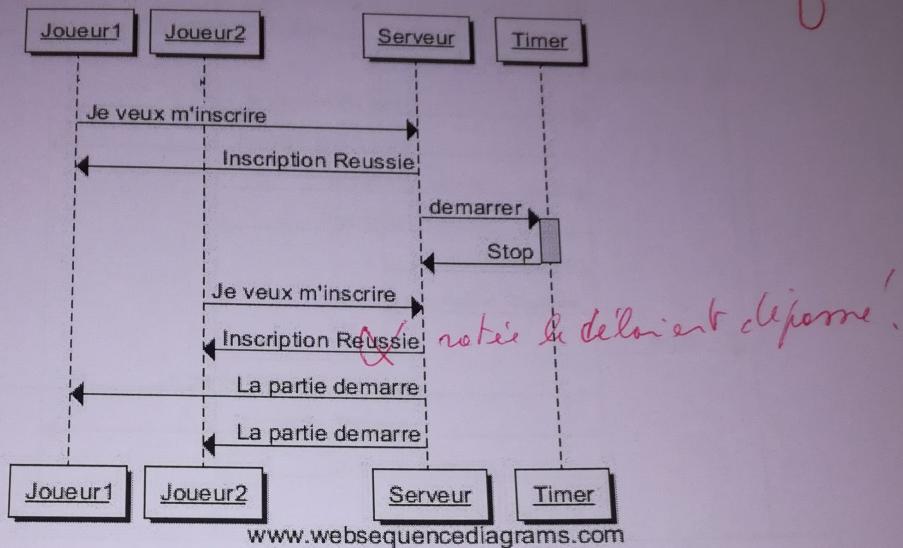


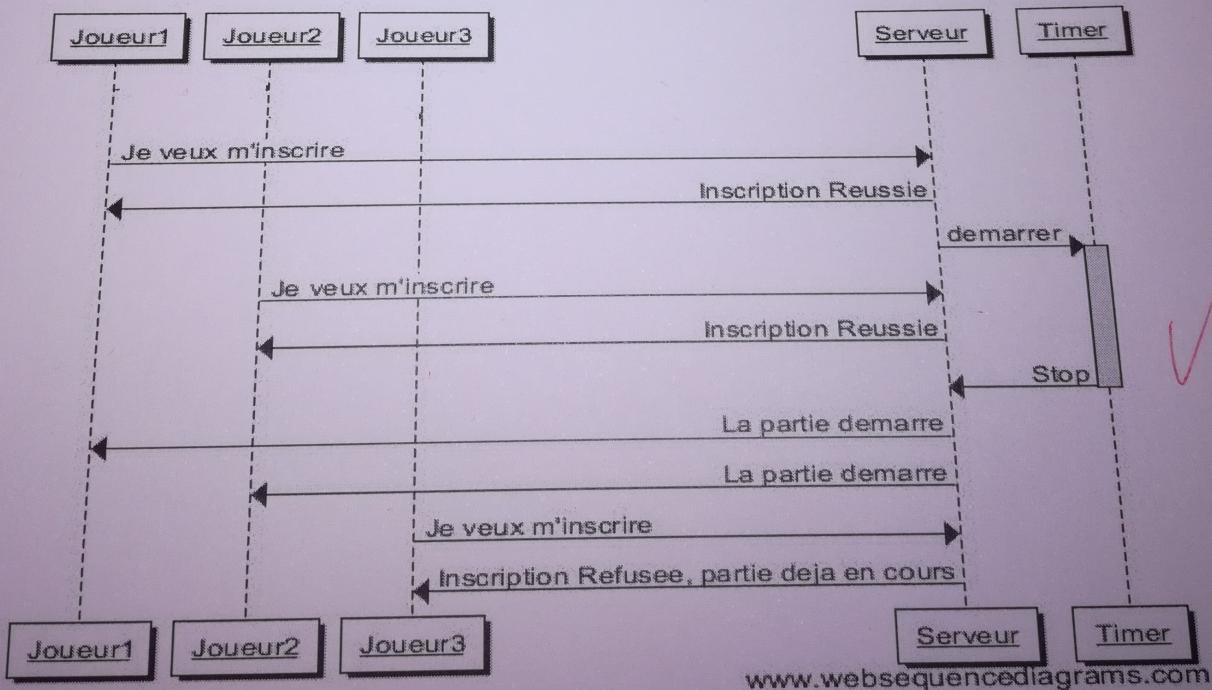
### Analyse préliminaire

#### 1) Scénarios :

##### Inscription - Un deuxième joueur veut s'inscrire après le timer :



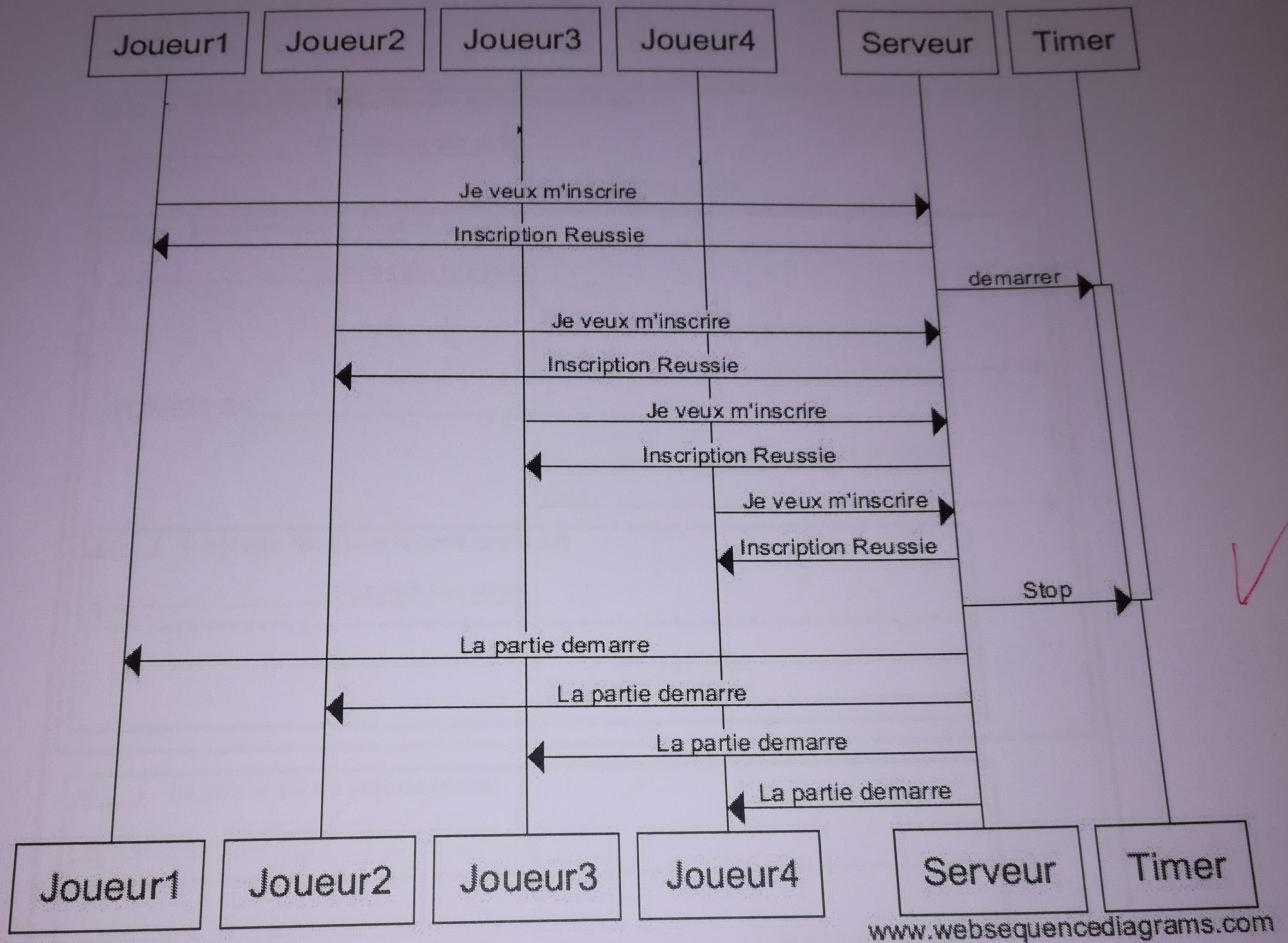
##### Inscription - Un deuxième joueur veut s'inscrire durant la durée du timer + Une inscription tardive après lancement de partie :



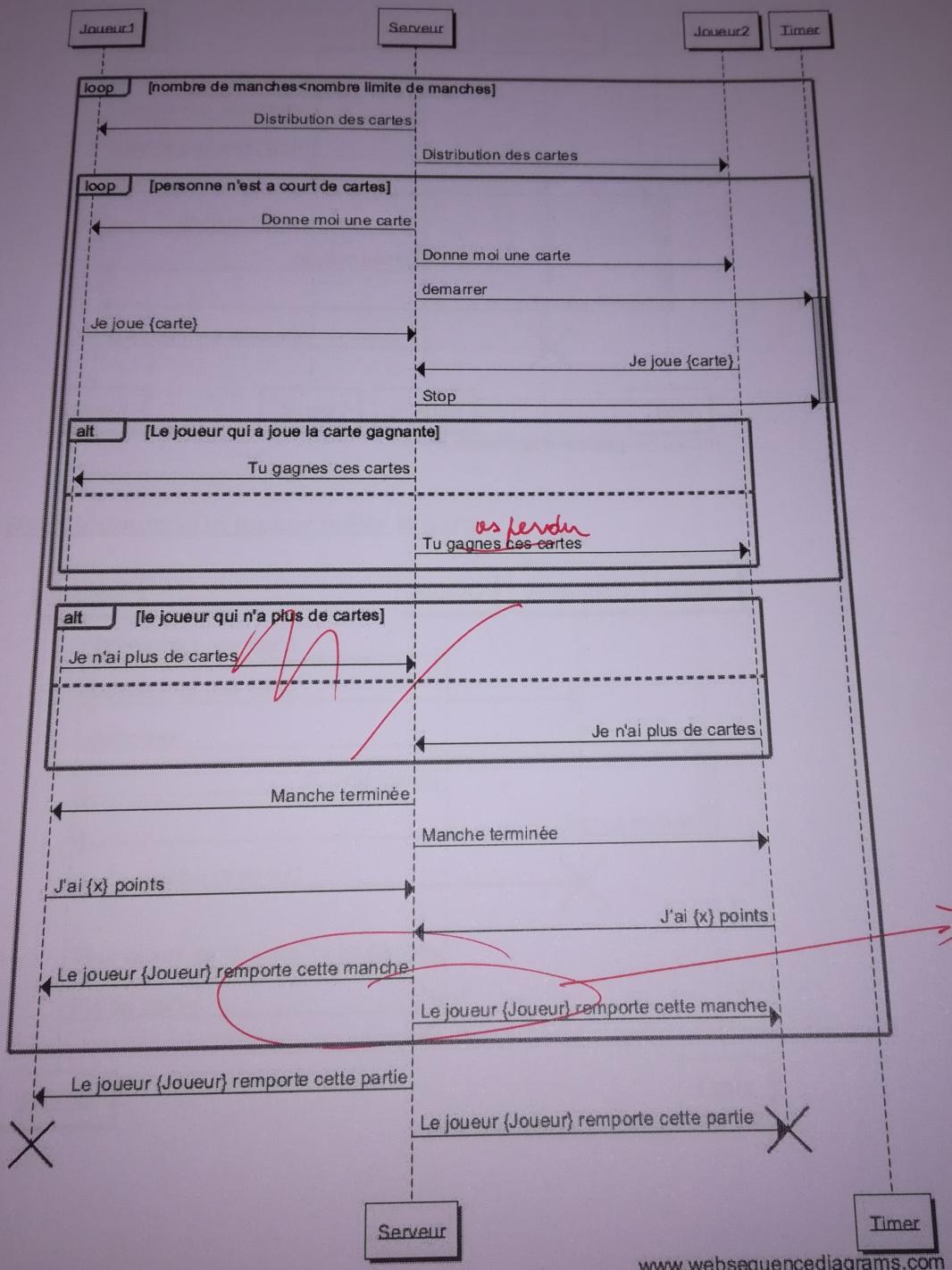
YAKOUB Jacques  
VERWILGHEN Maxime  
Série 1

27-04-2016

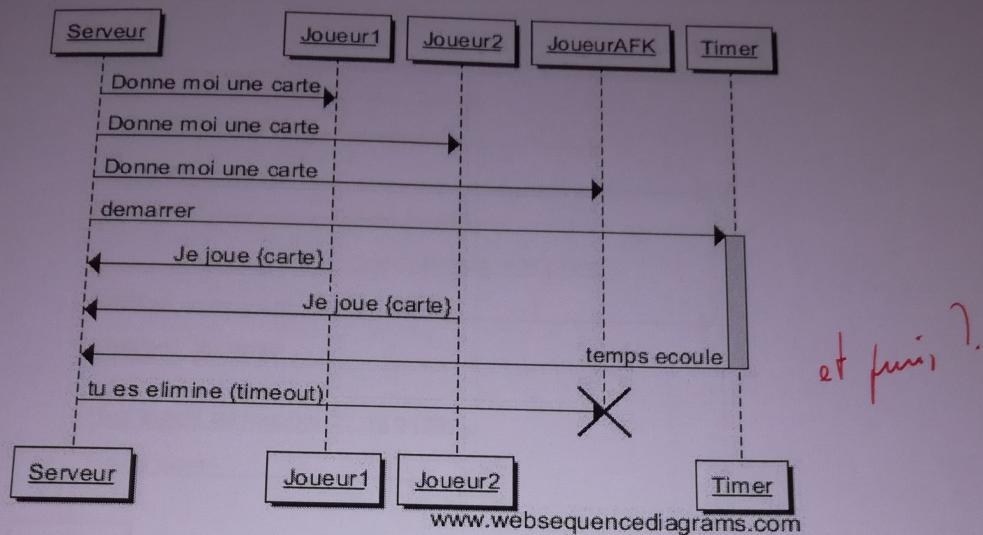
Inscription - 4 Joueurs s'inscrivent :



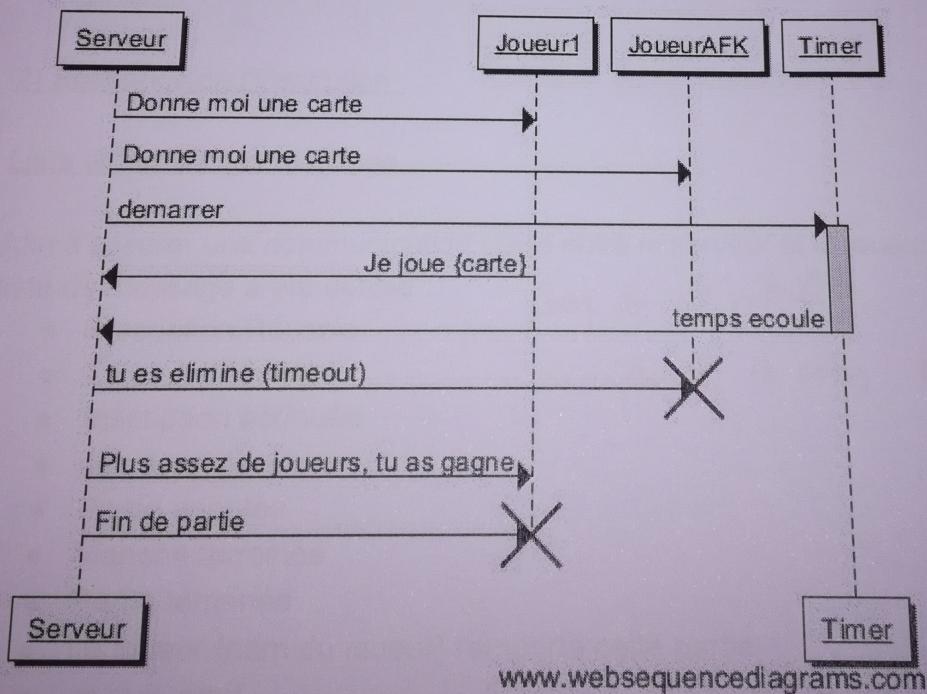
Déroulement d'une partie "normale" :



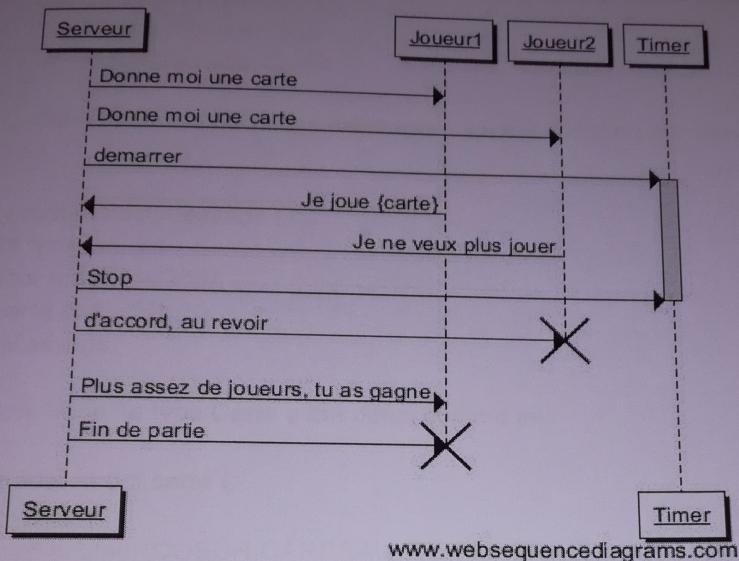
Trois joueurs - Un joueur quitte la partie :



Deux joueurs - Un joueur quitte la partie :



Deux joueurs - Un joueur souhaite arrêter gracieusement la partie :



## 2) Scénarios de l'inscription :

### Liste et format des messages :

Afin d'assurer une communication claire entre le serveur et le joueur, la présente liste de message a été définie : *avec la lisibilité.*

- Inscription Réussie
  - La partie démarre
  - Inscription échouée
  - Je veux m'inscrire
  - Partie annulée
  - Manche terminée
  - Partie terminée
  - Le joueur {nom du joueur} remporte cette partie
  - Je suis mort.
  - Le joueur {Joueur} remporte cette manche
  - Je joue {carte}
  - Distribution des cartes
  - J'ai {x} points.
  - Le serveur demande une carte

pour la lisibilité,  
indiquez le sens serveur → jeu  
on  
jeu → serveur

YAKOUB Jacques  
VERWILGHEN Maxime  
Série 1

27-04-2016

- Tu gagnes ces cartes
- Je veux m'inscrire
- Donne moi une carte
- Tu gagnes ces cartes

Concernant la structure des messages , il a été choisi de les représenter comme ceci :

```
typedef struct message {  
int type;  
char message[256];  
Carte cartes[4];  
} Message;
```

Nota bene, le type Carte a été défini comme tel :

```
typedef struct carte {  
int valeur;  
enum couleur{COEUR,CARREAU,TREFLE,PIQUE};  
} Carte;
```