Table des matières

[Intro Projet : (30s – 1mn) 1](#_Toc532390560)

[Explication du projet : 1](#_Toc532390561)

[Aperçu du programme en cours d’exécution (10sec) pour illustrer. 2](#_Toc532390562)

[Développement : 2](#_Toc532390563)

[Personnel : -> Phrase d’introduction + Description basée sur l’UML. 2](#_Toc532390564)

[Salle : 2](#_Toc532390565)

[Cuisine : 2](#_Toc532390566)

[Client : 2](#_Toc532390567)

[Console et paramétrage : 2](#_Toc532390568)

[Logs 2](#_Toc532390569)

[Temps 2](#_Toc532390570)

[Synchronisation : 2](#_Toc532390571)

[Communication entre les différents postes (synchro et échanges de données) : 2](#_Toc532390572)

[Utilisation des pools pour la gestion de certaines ressources : 2](#_Toc532390573)

[Synchronisation du service d’une table pour chaque plat 2](#_Toc532390574)

[BDD : 2](#_Toc532390575)

[La carte du restaurant a été complété et dispose de 10 recettes pour chaque partie du repas : 2](#_Toc532390576)

[Les scenarios sont enregistrés en BDD : 2](#_Toc532390577)

[Chaque tache de chaque processus ou thread doit être horodatée dans un log : 2](#_Toc532390578)

[Conclusion : 2](#_Toc532390579)

[Optimisation, pistes d’améliorations, et fonctionnalité(s) en cours de développement : 2](#_Toc532390580)

# Intro Projet : (30s – 1mn)

## Explication du projet :

Outil de simulation et de supervision du fonctionnement d’un restaurant permettant de mettre en évidence les lacunes des différents process de l’établissement, résultats qui seront ensuite utilisés afin d’en tirer des propositions d’amélioration.

Des propositions qui seront dans l’obligation de reposer sur des éléments mesurables pour mettre en évidence, sans ambiguïté, les gains que nos propositions suggèrent, et devant être exprimées en unité de mesure (temps, pourcentage, argent, etc…).

Aperçu du programme en cours d’exécution (10sec) pour illustrer.

Le programme consiste en un aperçu 2D du restaurant avec des personnages représentant clients et personnels, avec à chacun un comportement qui leur est propre, interagissant avec les éléments d’intérêt (lave-vaisselle, chambre froide, tiroir des ustensiles, etc ...) et donc modifiant notre base de données.

# Développement :

1. Démarrage de la vue / système d’affichage
2. Début du déroulement du lancement du programme : Instanciation des éléments du programme

## Personnel : -> Phrase d’introduction + Description basée sur l’UML.

### Salle :

#### Maitre d’hôtel :

Explication : le maître d’hôtel est responsable de l’accueil du client. Il représente la première image que le client aura de l’établissement. Pour le besoin de notre projet, il attribue une table au client et appelle le chef de rang pour le placement sur la table.

Le maître d’hôtel est un poste fixe. Il ne bouge pas dans la salle.

On considère un seul maître d’hôtel pour la salle.

Image :

#### Chef de rang :

Explication : Responsable d’un certain nombre de tables, qui composent un rang. Il est sous la responsabilité du maître d’hôtel et peut lui-même superviser et coordonner l’action de commis positionné sur son rang. Pour simplification pour notre projet, nous considérerons que nous avons un seul chef de rang par carré. C’est lui qui distribue les cartes et prend les commandes.

Le chef de rang se déplace dans la salle entre le poste du maitre d’hôtel et son carré.

S’il n’y a aucune tâche à faire, il peut soit donner un coup de main à l’autre chef de rang, soit attendre quelque part dans son carré ou à l’accueil du restaurant.

On considère un seul chef de rang par carré.

Image :

#### Serveurs:

Explication : Le serveur a pour fonction de servir les clients de l'établissement après la commande. Il est garant du déroulé du service pour les tables d’un carré dont il a la charge et donc de la satisfaction client. A la fin de chaque plat, il ramasse les assiettes et couverts des clients et les emmène à un endroit de stockage (ils seront pris en charge par la plonge en cuisine).

Le serveur se déplace seulement dans son carré. S’il n’y a pas de clients dans son carré, il peut donner un coup de main aux autres carrés.

On considère 2 serveurs par carré.

Image :

#### Commis :

Explication :

Image :

#### Clients :

Explication :

Image :

### Cuisine :

#### Chef de cuisine :

Explication : Il est le Chef d'Orchestre de la cuisine. Le chef de cuisine coordonne, dirige et supervise l'ensemble de l'activité de la cuisine. Il dispatche les tâches aux chefs de parties. De lui dépend le bon ordonnancement du service aux clients. Il y en a un seul dans la cuisine.

Image :

#### Chef de partie :

Explication : Les chefs de partie sont spécialisés dans leur « partie » et ont la responsabilité de leurs commis, apprentis et/ou stagiaires respectifs.

Image :

#### Commis de cuisine :

Explication : Ils sont là pour apprendre le métier grâce à l'aide de l'ensemble de la brigade. Notre restaurant compte 2 commis.

Pour simplifier le projet, nous allons leur assigner certaines tâches bien précises.

Image :

#### Plongeur :

Explication : Ils s'occupent de laver la batterie (matériel de cuisine) aussi bien que la vaisselle, nettoient parfois les locaux ou aident à l'épluchage et au taillage des légumes. Notre restaurant compte un plongeur en cuisine.

Image :

### Client :

#### Différents types :

Explication : 3 types différents en fonction de leur empressement

Image :

#### Départ du client à la fin du repas et remise au propre de la table

Explication :

Image :

## Console et paramétrage :

### Logs

Explication : Présentation du controller logs

Image : Aperçu console

Temps

Explication : Paramétrable avant/pendant, PAUSE

Image : Partie graphique pause/accélération

## Synchronisation :

### Communication entre les différents postes (synchro et échanges de données) :

Explication :

Image :

### Utilisation des pools pour la gestion de certaines ressources :

Explication :

Image :

### Synchronisation du service d’une table pour chaque plat

Explication :

Image :

## BDD :

### La carte du restaurant a été complété et dispose de 10 recettes pour chaque partie du repas :

Explication :

Image :

### Les scenarios sont enregistrés en BDD :

Explication :

Image :

### Chaque tache de chaque processus ou thread doit être horodatée dans un log :

Explication :

Image :

# Conclusion :

## Optimisation, pistes d’améliorations, et fonctionnalité(s) en cours de développement :