**Introduction**

Pour ce projet Simulation, nous avons décidé de simuler un hôpital et plus particulièrement, ces salles d’opérations. Le bût de la simulation étant donc de représenter l’adaptation du planning d’attribution des salles aux fur et à mesure de la journée en fonction de diverses règles de gestion appliquées. Ainsi, une fois la simulation déroulée, les temps d’attentes des patients et les temps où certaines salles sont libres sont analysable pour en déduire l’efficacité des règles de gestion appliquées. En plus de ces salles, nous avons aussi décidé de représenter les infirmiers et les chirurgiens, ainsi les salles suivront un cycle du type : une salle est affectée > un infirmier vient préparer la salle > un chirurgien vient opéré > un infirmier vient libérer la salle > la salle est disponible.

Pour se faire, nous avons identifié un seul type d’entité : les patients. Concernant les ressouces nous en avons identifié 3 types : les salles, les infirmiers et les chirurgiens. Nous avons ensuite séparés nos patients en 2 types : les patients rendez-vous, qui sont les patients que l’on connaît dès le début de la simulation et nos patients urgents, qui seront déclarés pendant la simulation. Nous avons également séparé nos salles en 3 types : les salles peu équipées, les salles semi équipées et les salles très équipées. Ces différentes salles correspondent aux différentes « gravité » caractérisés par nos patients. Ainsi, des patients de gravité « Peu équipée » pourront être affecté à nimporte quelle salle, tandis que des patients de gravité « Trés équipées » ne pouront être affecté qu’à des salles « très équiées ». Pour rajouter plus de bruit à la simulation et se rapprocher encore plus de la réalité, nous avons décidé que chaque patient aura un temps d’opération qui lui est propre et qui n’est pas connu à l’avance.

Une fois cela fait, nous avons refléchis aux différentes qui allaient être importantes. Dans un premier temps, nous allons avoir besoin de représenter le modele simulation, c’est-à-dire les entités, les ressources, les evenements et les règles de gestions mais aussi les listes d’attentes et le planning. Pour se faire, nous avons décidé de créer un objet simulation qui contiendrait l’ensemble de ces composantes. Ensuite, nous avons estimé utile de développer une fenetre de saisie sur laquelle l’ensemble des données caractérisant la Simulation (nombre d’infimieres, temps de préparation d’une salle, règle de gestion appliquée, …) peuvent être saisie, extraite ensuite dans un fichier texte, permettant ensuite de créer la Simulation, après lecture. Ensuite, nous décidé de développer un outil qui extrait les valeurs de temps après la simulation dans un fichier json, permettant d’observer le résultat. Enfin, nous avons pensé qu’il pouvait être utile de créer une fenetre qui lit ce fichier json pour en faire des graphiques.

**L’objet Simulation**

C’est logiquement cette partie qui nous a pris le plus de temps. Dans un premier temps nous avons du identifier ses composantes essentielles. Une fois fait nous avons obtenu le diagramme qui suit :

--

Suite, à cela, nous avons commencé à coder. Nous avons commencé par simplement créer l’ensemble de ces classes sans s’occuper de coder les methodes pour le moment. Nous avons ensuite coder les entités, les ressources et la classe Simulation. Une fois fait, nous nous sommes repartis le codage des evenements et des regles de gestions. Le codage du contenu des évènements a rapidement du être arrêté car les règles de gestions devaient être finis avant. Nous avons donc commencé à travailler sur l’initialisation de l’objet Simulation puis sur l’extraction du fichier json et enfin la fenêtre d’initialisation. Nous avons ensuite finalisé le codage des règles de gestion et pu s’attaquer aux événements.