

Projet Java IV 2023-2024

HELBThermo

Introduction :

La société HELBThermo cherche à se spécialiser dans la modélisation de systèmes thermodynamiques. Pour cela, elle vous demande de créer un logiciel permettant la simulation et mesure d'un système où des échanges de chaleur ont lieu.

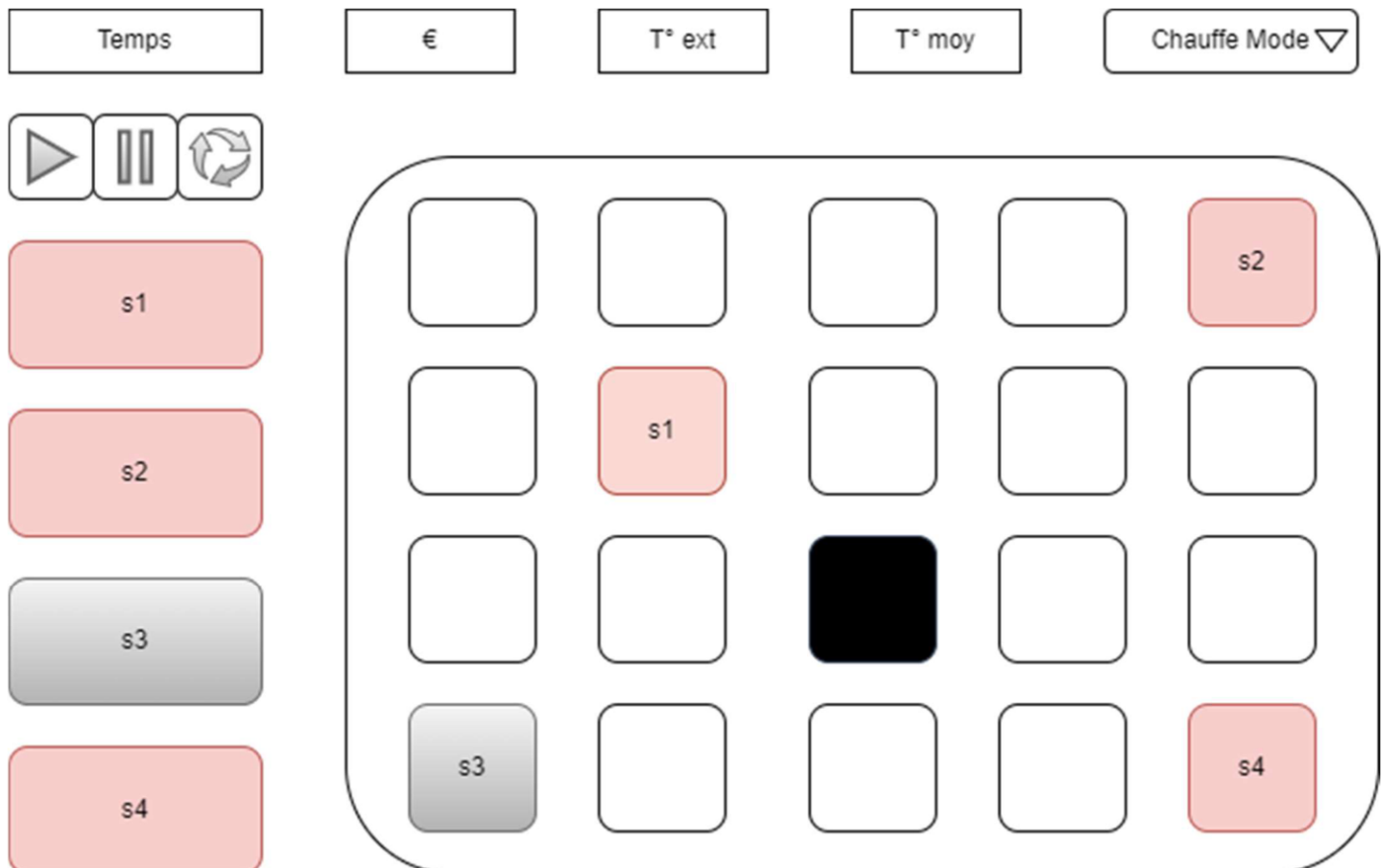
Description du problème :

Un système est un espace rectangulaire composé de cellules. Ces cellules sont influencées par la température extérieure. **Sans aucune intervention dans le système, toutes les cellules ont une même température qui est celle de la température extérieure.**

Certaines cellules du système sont des sources de chaleur qui peuvent influencer les autres cellules du système. **Quand une source de chaleur est activée, elle diffuse sa chaleur aux autres cellules du système.**

Interface de contrôle :

Afin d'interagir avec le système, on vous demande d'implémenter une interface de contrôle. Voici un schéma spécifiant la disposition des différents éléments de l'interface :



La partie centre droit de l'interface représente le système avec les différentes cellules. On peut y voir 4x5 cellules au total. Parmi ces cellules, les cellules dont le nom commence par « s » sont des sources de chaleur. Celles-ci sont activables/désactivables via le panneau de configuration situé à gauche. Une couleur grisée permet de montrer que la source de chaleur est désactivée. Quand une cellule est activée, sa couleur dépend de sa température.

Les éléments supérieurs de l'interface représentent différentes mesures et options possibles pour interagir avec le système.


Au lancement du programme, le système n'est pas activé, il est en pause. Il est possible de cliquer sur « Start » pour démarrer le système. Une fois le système démarré, on peut voir le nombre de secondes qui défilent dans l'emplacement dédié. On peut également voir le coût de chauffage du système, dont la valeur change avec le temps en fonction des sources de chaleurs activées. On peut également visualiser, la température extérieure au système ainsi que la température moyenne du système. Enfin, il est possible de définir le mode de planification de chauffe, qui permet une activation/désactivation automatique des différentes sources.

À tout moment, on peut mettre en pause/réactiver le système, ainsi que procéder au reset, qui permet de remettre le temps ainsi que le coût à zéro.

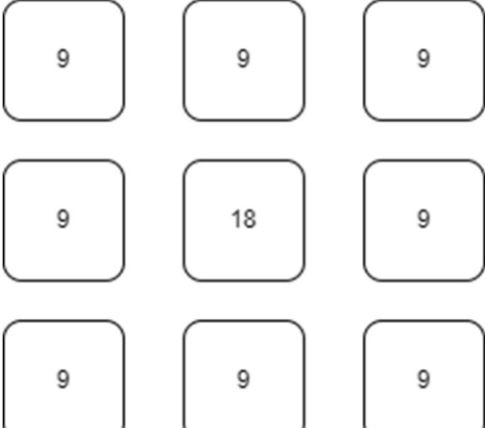
Diffusion de la chaleur et couleur des cellules :

La diffusion de chaleur se fait, pour chaque cellule, en prenant la moyenne de la chaleur de la cellule et des cellules adjacentes. Les cellules en bordure du système doivent également prendre en compte pour cellules adjacentes, la température extérieure.


Prenons un exemple pour illustrer. Supposons le système suivant, composé de 4 cellules, sans aucune source de chaleur, avec une température extérieure de 0° et avec les températures initiales suivantes :

		$T^{\circ}(n)$	$T^{\circ}(n+1)$
	0,0	9	$(0*5+9*4) / 9 = 4$
	0,1	9	$(0*5+9*4) / 9 = 4$
	1,0	9	$(0*5+9*4) / 9 = 4$
	1,1	9	$(0*5+9*4) / 9 = 4$

Un autre exemple un peu plus complexe, toujours avec une T° extérieure de 0° et sans source de chaleur :

		$T^\circ(n)$	$T^\circ(n+1)$
	0,0	9	$(0*5+9*3+18)/9=5$
	0,1	9	$(0*3+9*5+18)/9=7$
	0,2	9	$(0*5+9*3+18)/9=5$
	1,0	9	$(0*3+9*5+18)/9=7$
	1,1	18	$(9*8+18)/9=10$
	1,2	9	$(0*3+9*5+18)/9=7$
	2,0	9	$(0*5+9*3+18)/9=5$
	2,1	9	$(0*3+9*5+18)/9=7$
	2,2	9	$(0*5+9*3+18)/9=5$

Les sources de chaleurs, quand elles sont activées, gardent toujours une température constante, peu importe les cellules adjacentes où la température extérieure. Pour illustrer, le même exemple que l'exemple précédent avec la cellule centrale étant une source de chaleur active :

		$T^\circ(n)$	$T^\circ(n+1)$
	0,0	9	$(0*5+9*3+18)/9=5$
	0,1	9	$(0*3+9*5+18)/9=7$
	0,2	9	$(0*5+9*3+18)/9=5$
	1,0	9	$(0*3+9*5+18)/9=7$
	1,1	18	18
	1,2	9	$(0*3+9*5+18)/9=7$
	2,0	9	$(0*5+9*3+18)/9=5$
	2,1	9	$(0*3+9*5+18)/9=7$
	2,2	9	$(0*5+9*3+18)/9=5$

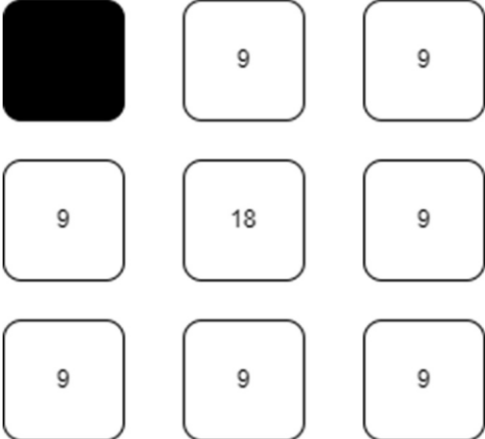
Quand les sources de chaleur sont désactivées, leur température est influencée par les cellules adjacentes et la température extérieure de la même manière que pour les autres cellules.

La couleur d'une cellule est représentative de sa température. Plus la température est élevée, plus la couleur de la cellule tend vers un rouge prononcé. Quand une cellule a une température de zéro, sa couleur est blanche. Quand sa température augmente, sa couleur devient un peu plus rouge. Si sa température augmente encore, sa couleur devient encore plus rouge. Pour illustrer, un exemple possible :

0	9
9	18

Cellules mortes :

Certaines cellules sont des cellules mortes. Quand une cellule est une cellule morte, sa couleur est noire. Une cellule morte n'a pas de température et ne diffuse pas la chaleur. Une cellule morte ne rentre pas en compte dans le calcul de la température moyenne du système. Si une cellule possède une cellule morte comme cellule adjacente, sa température n'est pas influencée par cette voisine morte. Un exemple pour illustrer, avec une cellule morte en (0,0), une température extérieure de 0° et aucune source de chaleur :

		$T^{\circ}(n)$	$T^{\circ}(n+1)$
	0,0		
	0,1	9	$(0*3+9*4+18)/8=6,75$
	0,2	9	$(0*5+9*3+18)/9=5$
	1,0	9	$(0*3+9*4+18)/8=6,75$
	1,1	18	$(9*7+18)/8=10,125$
	1,2	9	$(0*3+9*5+18)/9=7$
	2,0	9	$(0*5+9*3+18)/9=5$
	2,1	9	$(0*3+9*5+18)/9=7$
	2,2	9	$(0*5+9*3+18)/9=5$

Les cellules mortes sont générées aléatoirement dans le système au lancement du programme. Certaines caractéristiques augmentent les chances pour une cellule d'être une cellule morte au lancement du programme : Plus une cellule est proche du centre du système, plus elle a de chances d'être une cellule morte. Plus une cellule est éloignée des sources de chaleur, plus elle a de chances d'être une cellule morte.

Sources de chaleur et cout de chauffage :

Vous devez modéliser le cout de chauffage du système. Pour cela on vous demande de considérer que, une source de chaleur, allumée pendant n seconde à une température de d degrés a un cout de $n*(d^2)$ euros.

Par exemple, un système chauffé pendant 5 secondes par deux sources de chaleurs activées respectivement à 5° et 10° aura un cout de $5*(5^2)+5*(10^2) = 125+500=625$ euros à l'issue de ces 5 secondes.

Planification de chauffe :

Il existe différents modes permettant d'automatiser le système de chauffe.

- Le mode manuel : dans ce mode, l'utilisateur a le total contrôle des sources de chaleur. Tant qu'il ne les active pas, elles ne s'activent pas.
- Le mode successif : dans ce mode, chaque source de chaleur est allumée successivement pendant une seconde.
- Le mode target : dans ce mode, les sources de chaleur sont activées/désactivées de manières à ce que la température moyenne du

système se maintienne à 20°. Quand la température moyenne dépasse cette température, des sources sont désactivées. Quand la température moyenne est inférieure à cette température, des sources sont activées.

Température extérieure :

La température extérieure du système peut varier. Afin de simuler l'évolution de la température du système, un fichier .data vous sera fourni. Chaque ligne du fichier correspond à la température du système pendant une seconde. Exemple :

0
0
5
5
5
2

Au lancement du programme, la température est de 0° pendant 2 secondes. Elle passe ensuite à 5° pendant 3 secondes et enfin 2° pendant 1 seconde. Puisqu'il n'y a plus de données ensuite, la température extérieure reste constante à 2°.

Configuration du système :

Votre application doit pouvoir fonctionner pour plusieurs configurations de systèmes. Autant le nombre de lignes que de colonnes doit être configurable dans le code. La taille des cellules doit s'adapter à la configuration. Quand le système comporte beaucoup de lignes et de colonnes, il faut que les cellules soient automatiquement plus petites.

Au lancement du système, il n'y a pas de sources de chaleur. Il est toutefois possible à tout moment de cliquer sur une cellule pour la définir comme source de chaleur. Cela permet de faire apparaître un menu proposant différentes options. Voici une maquette permettant d'illustrer les différentes options de ce menu :

Position de la cellule

Définir comme cellule morte :	<input checked="" type="checkbox"/>
Définir comme source de chaleur :	<input checked="" type="checkbox"/>
T° de la source quand activée :	20
<input type="button" value="Valider"/>	

Il doit être possible de visualiser la position de la cellule dans le système, de définir la cellule comme étant une cellule morte ou non, source de chaleur ou non, et la température de la source si elle est activée. Il n'est pas possible pour une cellule d'être à la fois une source de chaleur et une cellule morte. Quand ce menu est ouvert, le système est mis en pause et il n'est plus possible d'interagir avec la fenêtre principale du système jusqu'à la fermeture de ce menu.

Log système :

Votre application doit permettre un log du système. Concrètement, quand votre application est fermée, un fichier dont le nom est « DDMMYY_hhmmss.log » est créé avec l'état du système à chaque seconde. Par exemple, si votre application est fermée le 23/02/2024 à 13h14m12s, le nom du fichier doit être « 230224_131412.log ».

Chaque ligne est composée de la façon suivante :

«nombre de secondes;cout;T°moyenne;T°externe»

Travail demandé :

Pour ce projet, vous allez devoir implémenter **une application JavaFX** en appliquant les concepts vus en cours tel que **la conception pilotée par les responsabilités** et **l'application adéquate de design patterns**.

Le projet devra être **développé individuellement**.

Contrainte de développement :

- Votre programme devra être compilable et exécutable dans un environnement Linux Ubuntu tel qu'une des machines virtuelles utilisées en cours et disponible sur le SharePoint d'eCampus. Un script bash nommé « run.sh » devra permettre la compilation et l'exécution de l'application en utilisant seulement la commande suivante «bash run.sh » en terminale. Si le fichier n'est pas fourni, ou si la compilation et l'exécution ne fonctionnent pas dans l'environnement spécifié, le projet ne sera pas corrigé et sanctionné d'un zéro. **Testez donc avant que tout fonctionne !**
- Votre code java devra être **compatible avec la version de l'openjdk** disponible sur cette même machine virtuelle.
- Votre programme devra prendre en compte **le fichier de simulation** fourni par l'enseignant.
- A l'exception des commentaires, l'entièreté de votre programme **devra être codé en anglais** (nom de variables/fonction/classes/etc...).

- Votre code devra respecter les principes de designs orientés objet comme vu au cours. Pensez donc à faire des choix logiques de design de classes afin de produire un **code propre et maintenable**.
- Votre code devra présenter **une structure correcte et maintenable**.
Notamment :
 - Evitez la duplication de code.
 - Evitez les constantes magiques.
 - Evitez le code mort.
 - Commentez intelligemment et suffisamment votre code
 - Nommez correctement vos variables et méthodes et organisez correctement votre code.

Il est fortement conseillé de consulter la partie du cours sur les bonnes pratiques de développement. Un code dont la qualité sera jugée insuffisante **sera sanctionné d'un zéro**.

- Votre programme devra être développé en utilisant **exclusivement la librairie graphique JavaFX**.

De manière générale, l'application devra faire un usage suffisant et correcte des concepts vus au cours.

Les contraintes et fonctionnalités du projet sont susceptibles d'évoluer au cours du temps. Pensez donc à adapter une stratégie de développement agile.

Note : certains points de la description ne sont pas précisés ou sont laissés volontairement vagues. Il revient à vous de faire certains choix d'interprétations. Veillez toutefois à ce que votre approche soit logique et justifiée.

Rapport :

Il vous est demandé de rédiger un rapport décrivant votre projet. Votre rapport devra contenir au minimum les sections suivantes :

Introduction : Cette section devra introduire votre projet. Décrire ce qui a été réalisé et présenter brièvement la structure de votre rapport. (max 1 page)

Fonctionnalités de base : Cette section devra expliquer les fonctionnalités offertes par votre application d'un point de vue à la fois fonctionnel et technique. (max 6 pages)

Analyse et Application des Design Patterns : Cette section devra expliquer la structure de votre implémentation en utilisant les outils d'analyses déjà vus durant votre parcours. Attention tous les diagrammes

doivent être commentés. Vous devrez également dans cette section décrire les patterns utilisés et expliquer comment et pourquoi vous les avez utilisés. Les diagrammes d'analyse pourront vous aider à illustrer vos propos. (max 6 pages)

Limitations : Les limites de votre application, par exemple : dans quels cas d'utilisation votre application pourrait ne pas fonctionner comme prévu ? Y a-t-il des aspects techniques qui n'ont pas été traités ? Si vous aviez plus de temps pour le projet, qu'auriez-vous amélioré ? Plusieurs points de vue sont possibles, il revient à l'étudiant de choisir les points qu'il considère les plus pertinents pour réaliser son autocritique. (max 2 pages)

Conclusion : Votre conclusion sur le projet. Ce que vous avez réussi à faire ou non durant le projet et les apprentissages que vous en tirez. (max 1 page)

Il vous est demandé de respecter le nombre maximum de pages par sections. La taille de la police d'écriture devra obligatoirement être supérieure ou égale à 12 et inférieure ou égale à 14 pour le contenu et les titres. Le rapport sera notamment évalué sur la qualité écrite et l'effort de présentation ainsi que la pertinence et la complétude des points abordés. **Un rapport dont la qualité écrite est jugée très insuffisante sera sanctionnée d'un zéro.**

Maitrise des productions :

L'évaluation du projet vise à attester de la bonne maîtrise, des concepts liés au cours et au développement du projet, par l'étudiant. Toute réalisation pour laquelle l'étudiant ne peut pas démontrer une maîtrise suffisante lors de l'évaluation orale ne sera pas prise en considération.

Deadline et remise :

La date limite pour la remise du projet est **encore à définir**. Le projet devra être déposé sur eCampus à l'intérieur d'un fichier .zip contenant toutes les sources de votre projet ainsi que le rapport au format DOCX.

Développement et Triche :

- Tout acte de triche sera sanctionné par **une note de fraude au bulletin et sera notifié à la direction qui pourra possiblement décider de sanctions supplémentaires**. Des parties de code réutilisés d'un projet existant (d'un autre étudiant ou disponible sur le net) sans références dans votre rapport et sans mention de l'utilité du code utilisé est considéré comme une fraude.
- Pour ce projet, **vous ne pouvez pas reprendre des parties du code d'un autre étudiant, de cette année, ou d'une année précédente**.

- Pour ce projet **vous ne pouvez pas vous inspirer/servir d'un jeu/code disponible sur internet.**
- Pour ce projet **vous ne pouvez pas vous inspirer/servir d'un code généré par des outils de génération de code tels que, par exemple, ChatGPT.**
- Si vous avez un doute, contactez l'enseignant le plus tôt possible afin d'éviter du refactoring inutile, ou pire, **une note de zéro/fraude.**

Conseil pratique :

Voici quelques conseils qui j'espère pourront vous aider.

- Veillez à ce que votre code ne contienne pas de constantes magique et/ou de duplication qui serait facilement évitable avec l'utilisation de méthodes.
- Veillez à effectivement implémenter les designs patterns quand cela est pertinent.
- Réfléchissez en terme d'« attribution de responsabilités » en accord avec les principes vu au cours (segmentation logique en classes).
- Ne négligez pas la théorie du cours (vous serez interrogés dessus).
- Ne négligez pas votre rapport. Tachez d'y expliquer/justifier explicitement vos choix d'implémentation. (Exemple : pourquoi un héritage ici ? pourquoi avoir choisi (ou non) d'implémenter ce design pattern ? quel avantage en termes de structure ? ...)
- Prenez le temps de bien comprendre tout l'énoncé avant de vous lancer (et lisez la FAQ).
- **Vérifiez que votre code compile et run effectivement via le script bash prévu sur la machine Ubuntu présente dans le SharePoint du cours.**

FAQ :

- **Puis je ajouter d'autres sections ou sous-sections dans le rapport ?**

Oui. La partie rapport de ce document donne seulement la structure minimum.

- **Puis je coder ou rendre mon rapport en anglais ?**

Oui. Pour ce qui est du code vous devez obligatoirement coder en anglais.

- **Puis je programmer sur Windows avec Eclipse ?**

Oui vous pouvez programmer comme vous le désirez mais vous devez respecter les contraintes de ce document, notamment : votre code doit être exécutable sur un environnement Linux Ubuntu via un script run.sh que vous devez fournir en

même temps que vos sources (voir les contraintes de développement). Une machine préconfigurée sera disponible sur le SharePoint, celle-ci pourrait être utilisée pour la démo de votre application lors de la défense du projet.

- **Le rapport est-il important ?**

Oui. Le rapport est une **pièce centrale de votre projet** et c'est le premier outil de communication qui me servira à juger de la bonne réalisation du projet, pas seulement du point de vue du code mais également de la méthodologie utilisée. Il convient pour le rapport d'être pertinent et complet quant aux informations contenues.

- **Que voulez-vous dire par « tous les diagrammes doivent être commentés ».**

Les diagrammes doivent servir à illustrer et appuyer vos explications sur la structure de votre implémentation. Ils ne remplacent aucunement un texte explicatif revenant sur les points d'attention.

- **Je n'ai pas réussi à tout réaliser. Est-ce que ça vaut la peine de vous rendre le projet ?**

Oui, veuillez toutefois à être claire sur les parties non implémentées. Il est très déconseiller de dissimuler ou d'« oublier » de mentionner qu'une partie n'a pas été réalisée. Veuillez toutefois à bien respecter les consignes. Par exemple, votre code doit pouvoir compiler avec le script bash demandé, la qualité du code doit être suffisante, etc...

- **Puis je réaliser le projet en groupe ?**

Non, le projet doit être réalisé individuellement.

- **Que voulez-vous dire par « Votre code devra respecter les principes de design orienté objet comme vu au cours. »**

L'orienté objet fait intervenir certains principes comme l'héritage ou les méthodes statiques. Il revient à vous de décider quand les mettre en œuvre ou non. Votre approche devra toutefois être logique et justifiée. Cela implique notamment, de faire apparaître de l'héritage quand cela a du sens, de rendre une classe abstraite quand cela a du sens, d'utiliser intelligemment l'encapsulation etc...

- **Dois-je vraiment faire de l'orienté objet ? Mon programme peut fonctionner sans.**

Vous devez absolument mettre en œuvre l'orienté objet pour ce projet. Cela fait partie des contraintes du projet. Un non-respect de ces contraintes sera pénalisé. L'utilisation de l'orienté objet n'est pas une contrainte faible. Veuillez donc à la respecter.

- **Doit-on faire de l'encapsulation ou de l'héritage et du polymorphisme ?**

Vous devez toujours en faire usage quand cela a du sens et ne pas le faire est généralement un signe de mauvaise conception.

- **Dois-je vraiment utiliser des design patterns ? Mon programme peut fonctionner sans.**

Vous devez absolument mettre en œuvre les design patterns pour ce projet. Cela fait partie des contraintes du projet. Un non-respect de ces contraintes sera pénalisé. L'utilisation des design patterns n'est pas une contrainte faible. Veillez donc à la respecter. Il ne s'agit toutefois pas d'utiliser les patterns sans réflexion quant à leur utilité dans le projet. Leur utilisation et choix ou non choix doit être pertinente et justifiée.

- **L'aspect graphique et l'ergonomie sont-ils des critères importants ?**

Ces critères seront susceptibles d'être pris en compte dans l'évaluation mais sont nettement moins importants que l'implémentation des fonctionnalités et le respect des contraintes.

Dis grossièrement : Mieux vaut un programme moche mais avec toutes les fonctionnalités implémentées qu'un programme magnifique mais avec des fonctionnalités manquantes et un design logique (architecture du code) désastreuse.

- **Puis je utiliser un outil comme Scene Builder pour la création d'interfaces graphique ?**

Oui tant que le code généré fasse partie de la bibliothèque d'interfaces utilisateur JavaFX.

- **Puis je utiliser des designs patterns qui n'ont pas été vu au cours ?**

Oui mais vous devez les définir, justifier leur utilisation et fournir des sources. Vous devez dans tous les cas justifier leur utilisation.

- **L'interface graphique doit-elle être exactement celle présentée dans l'énoncé ?**

Les informations et fonctionnalités présentes dans les maquettes doivent apparaître dans l'application. Vous ne devez pas ajouter de fonctionnalités mais vous êtes libres de parfaire l'esthétique et l'ergonomie.