1. Document d’analyse:

Analyse globale:

Interface graphique: Ovale et ligne brisée

Génération et défilement continu de la ligne brisée

Mouvement contrôlé de l’ovale

Analyse détaillée :

Dessin de l’ovale, difficulté basse priorité 1

Montée de l’ovale lors d’un click, difficulté moyenne, priorité 1

Descente permanente de l’ovale, difficulté basse, priorité 1

Rajout de vitesse, difficulté moyenne, priorité 2

(à compléter au fur et à mesure)

1. Plan de développement (incomplet)

Tâches :

Apprentissage Swing : 1h

Apprentissage threads : 30 min

Dessin de l’ovale : 15 min

Évènement souris : 15min

Thread de mouvement : 15 min

Rédaction document analyse et conception : 1h

Ressources :

1 seule

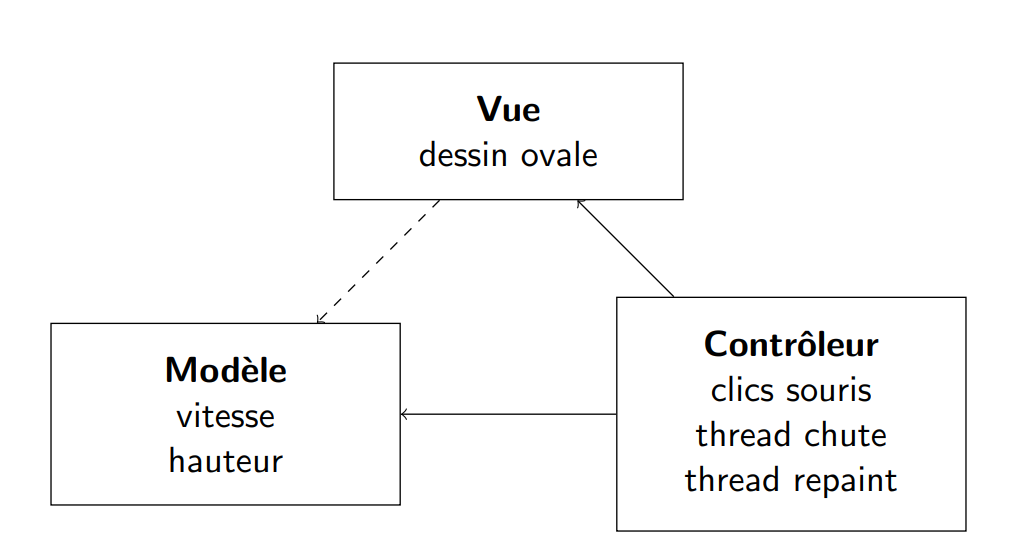
Diagramme de Gant :



1. Conception :

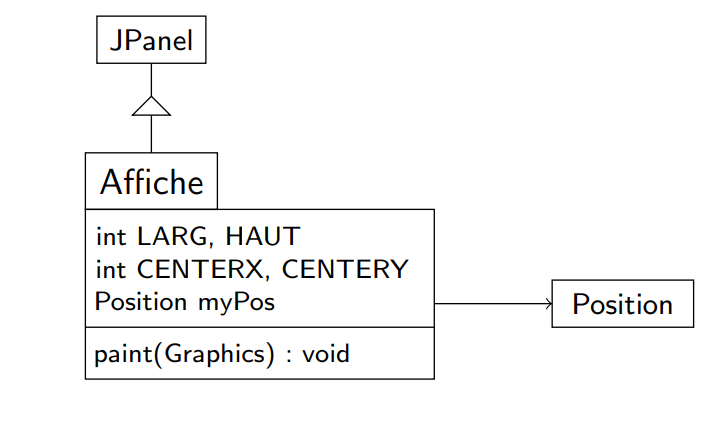
Conception générale :

Patron MVC :



Conception détaillée :

Dessin de l’ovale :



Clic souris :

* Programmation événementielle avec la classe MouseAdapter
* Vitesse
* Thread de mise à jour de l’affichage (délai de 50 ms dans une constante)

MouseAdapter

ReactionClic

mouseClicked(MouseEvent): void

Position

int vitesse

int JHEIGHT

jump(int): void

Thread

Redessine

int DELAY

run(): void

sleep(int): void

JPanel

Affichage

repaint(): void

Chute Ovale :

* Thread (et constante pour la vitesse)
* Hauteur

Thread

Descendre

int DELAY

run(): void

Position

int height, vitesse

int JHEIGHT

get(): int

move(): void

Thread

Redessine

sleep(int): void

1. Résultats:
2. Documentation utilisateur:

* Prérequis : Java avec un IDE
* Mode d’emploi : Importez le projet dans votre IDE, sélectionnez la classe Main à la racine du projet puis cliquez sur « Run ».
* Cliquez sur la fenêtre pour faire monter l’ovale

1. Documentation développeur

* Classe main
* Swing, awt, Thread

1. Conclusions et perspectives (à remplir)