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

Dessin de la ligne brisée, difficulté basse priorité 1

Génération de la ligne brisée, difficulté moyenne priorité 1

Défilement de la ligne brisée, difficulté moyenne priorité 1

(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 : 2h

Réorganisation du code en packages : 45 min

Création de la ligne: 30 min

Affichage de la ligne : 15 min

Déplacement de la ligne : 45 min

Ressources :

1 seule

Diagramme de Gant :

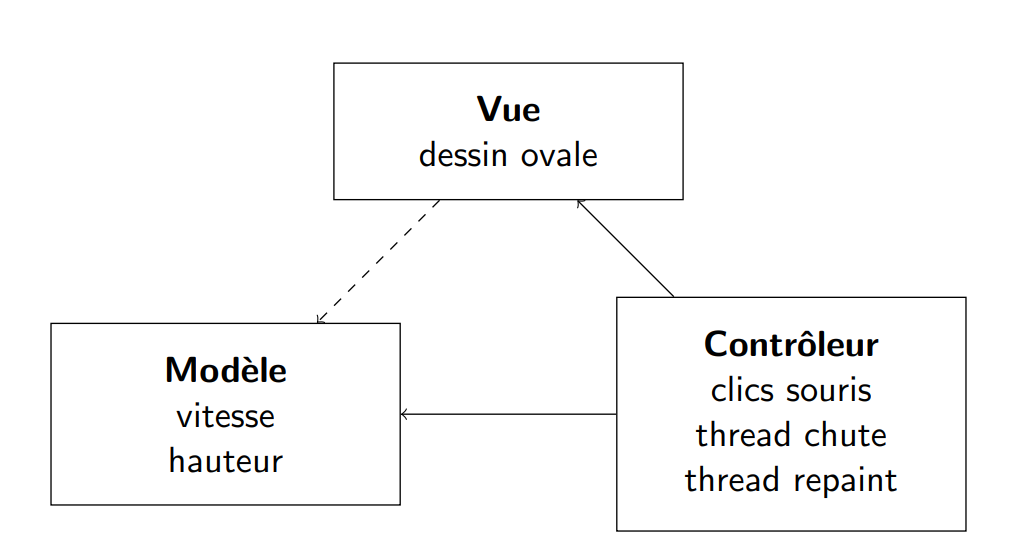




1. Conception :

Conception générale :

Patron MVC :



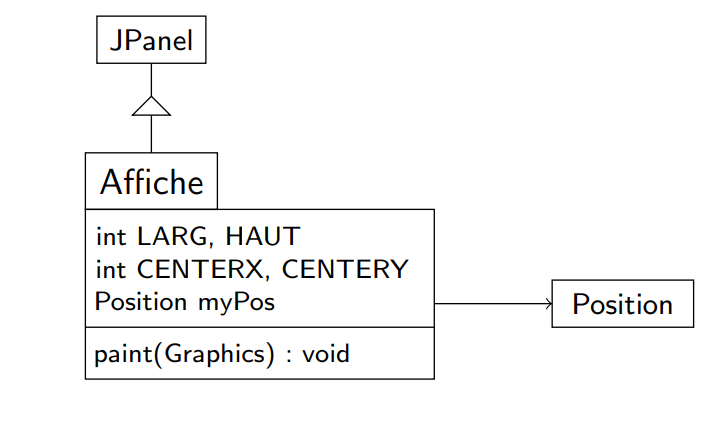
dessin ligne

Ligne

thread avance

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

get(): int

Thread

Redessine

int DELAY

run(): void

sleep(int): void

JPanel

Affichage

repaint(): void

revalidate(): 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

Génération ligne :

Algorithme :

Entrée : XMIN, XMAX, YMIN, YMAX, XDEPART, XFIN, YDEPART, LARG\_FENETRE

(YMIN >= bas de la fenêtre, YMAX <= haut de la fenêtre)

Sortie : Liste(Point)

L = [(0-XDEPART, YDEPART), (XDEPART, YDEPART)]

X = XDEPART + XMAX

Tant que X <= LARG\_FENETRE + XFIN

Last\_point = Get\_last\_point()

Y\_last\_point = Last\_point.y

Xr = Random(XMIN, XMAX)

Faire :

Ratio = Random (0, 10)

Yr = Y\_last\_point + (Ratio-5)\*10

Tant que Yr > YMAX ou Yr < YMIN

X = X + Xr

L += [(X, Yr)]

Défilement et affichage de la ligne :

* Position réelle des points avec le Array points
* Position affichée donnée par l’appel à la méthode get\_points()
* Position affichée = Position réelle – avancement (classe position)
* Mise à jour constante d’avancement avec le thread Avance
* Algorithme de génération et destruction des points le thread redessine appelle la méthode get\_points() il n’y a donc pas besoin de boucle while dans l’algorithme ci-dessous :

If points2[0].x < 0 – XDEPART then Remove(Points[0])

If points2[Size(points2) – 1].x < LARG\_FENETRE + XFIN then

Yr = 0

X = Get\_X(points, Size(points) – 1)

Last\_point = Get\_last\_point()

Y\_last\_point = Last\_point.y

Xr = Random(XMIN, XMAX)

Faire :

Ratio = Random (0, 10)

Yr = Y\_last\_point + (Ratio-5)\*10

Tant que Yr > YMAX ou Yr < YMIN

X = X + Xr

L += [(X, Yr)]

Parcours

ArrayList<Points> points

generate\_points(): void

get\_points(): ArrayList<Points>

Position

Int avancement

avance(): void

Thread

Redessine

int DELAY

run(): void

sleep(int): void

JPanel

Affichage

repaint(): void

revalidate(): void

Thread

Avance

int DELAY

run(): void

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)