TD n°4 : CISI

Exercice 1 :

Tracer des traits en pressant et en relachant la souris

Partie 1 : évènements

mousePressed(x,y), mouseMoved(x,y), mouseReleased(x,y)
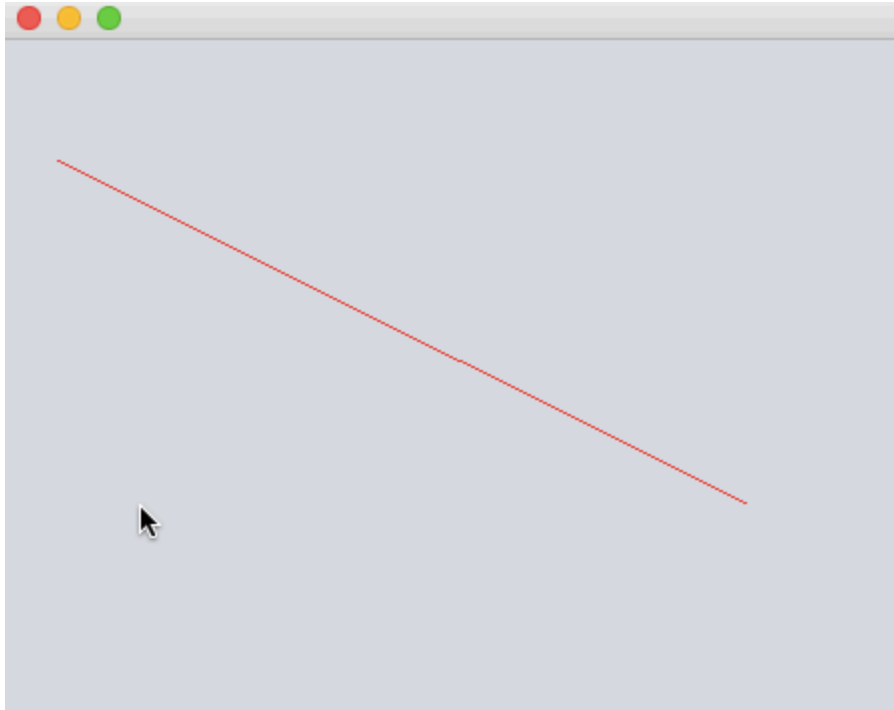-> mouseDrag en réalité (JAVA)

Partie 2 : actions

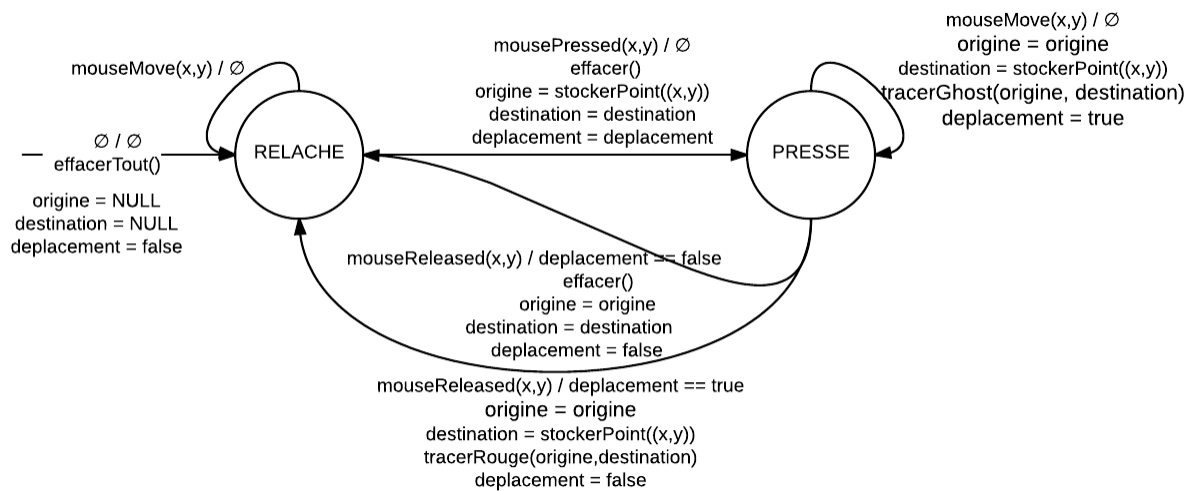tracerGhost(point1, point2)
effacer()
tracerRouge(point1, point2)
stockerPoint(x, y)

Partie 3 : automate
État: RELACHE, PRESSE
Variable coord

mousePressed(x,y) / ∅
effacer()
origine = stockerPoint((x,y))
destination = destination
deplacement = deplacement

mouseMove(x,y) / ∅

mouseMove(x,y) / ∅
origine = origine
destination = stockerPoint((x,y))
tracerGhost(origine, destination)
deplacement = true

∅ / ∅
effacerTout()

origine = NULL
destination = NULL
deplacement = false

RELACHE

PRESSE

mouseReleased(x,y) / deplacement == false
effacer()
origine = origine
destination = destination
deplacement = false

mouseReleased(x,y) / deplacement == true
origine = origine
destination = stockerPoint((x,y))
tracerRouge(origine,destination)
deplacement = false

Partie 4 : matrice état/évènements

	mouseMove(x, y)	**mousePressed(x,y)**	**mouseReleased(x, y)**

PRESSE	`s = PRESSE;`
	`origine = origine;`
	`destination = stockerPoint((x,y));`
	`tracerGhost(origine,destination);`
	`deplacement = true;`
		Impossible	`s = RELACHE;`
			`if (deplacement) {`
			`    origine = origine`
			`    destination = stockerPoint(x, y)`
			`    deplacement = false`
			`    tracerRouge(origine, destination)`
			`} else {`
			`    origine = origine`
			`    destination = destination`
			`    deplacement = false`
			`    effacer()`
			`}`

RELACHE	`s = RELACHE`
	`origine = origine`
	`destination = destination`
	`deplacement = deplacement`
		`s = PRESSE;`
		`effacer();`
		`origine =`
		`stockerPoint((x,y));`
		`destination =`
		`destination;`
		`deplacement =`
		`deplacement;`
			Impossible

```
s = RELACHE
origine = NULL;
destination = NULL;
deplacement = false;
```

Partie 5 : Pseudo code

Exercice 2 :

Clic : origine
Bouger la souris : traçage ghost
Autre clic : destination puis traçage
Touche Espace : ferme le polygone
Clic droit : supprime la dernière origine

Partie 1 : évènements

1(x, y), mouseRightClicked(x,y), keyPressed(space), mouseMove(x,y)

Partie 2 : actions

effacer();
dessinerGhost(Point destination);
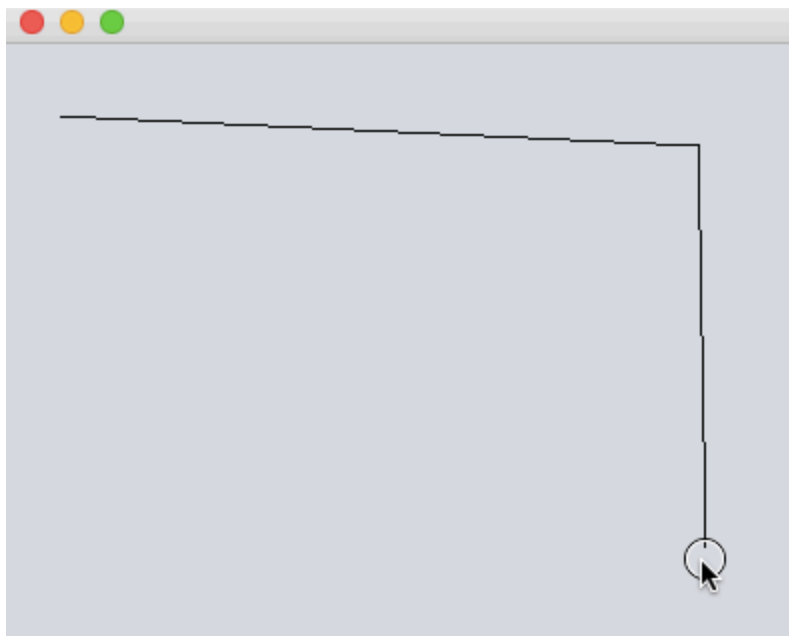dessinerListeRouge();
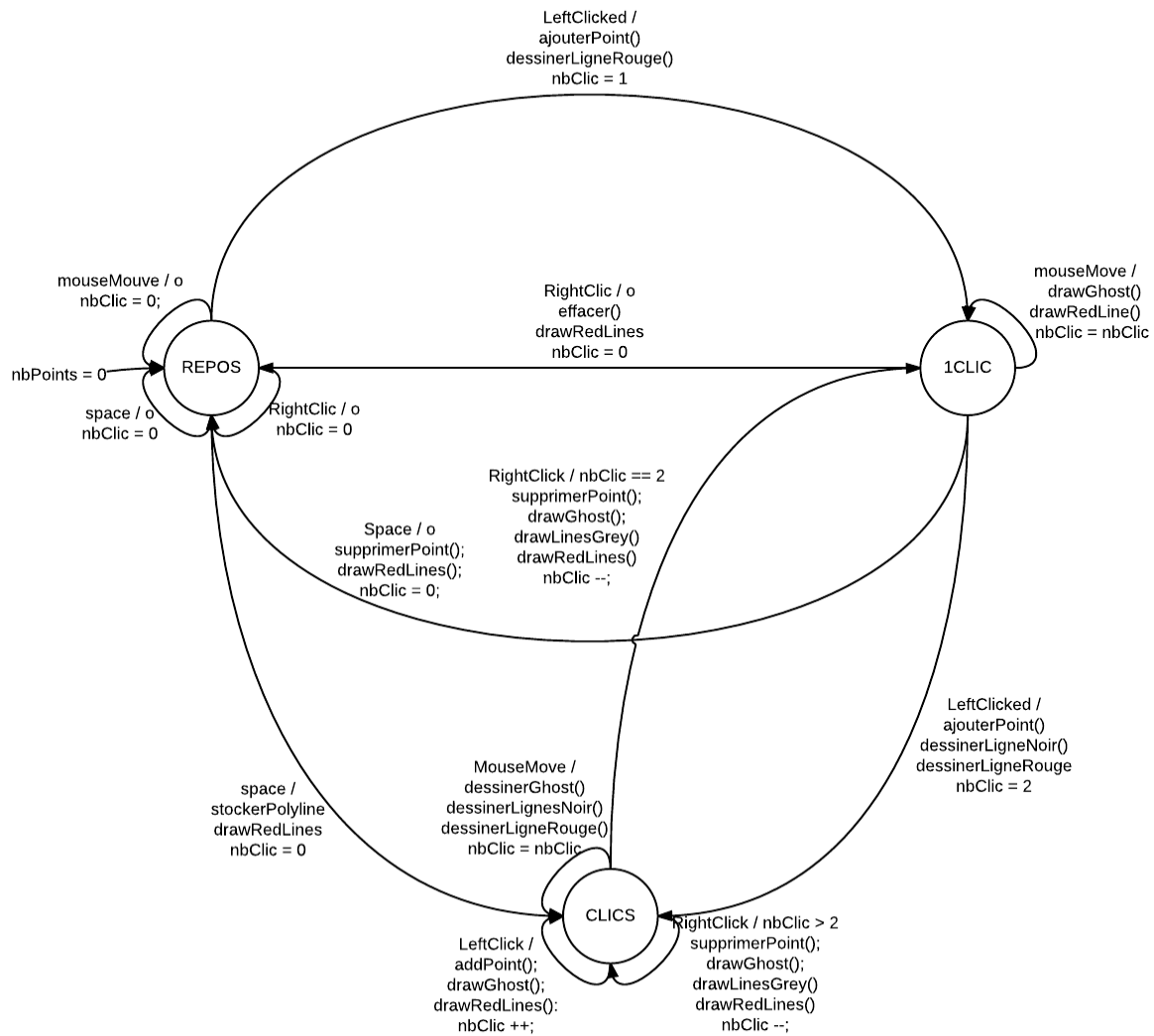dessinerListeNoir();
ajouterPoint(p)
retirerPoint()
stockerPolyline(point1);

Partie 3 : automate

https://www.lucidchart.com/invitations/accept/8c0987c3-83f1-4a1b-9c94-214f63c4a19b

LeftClicked /
ajouterPoint()
dessinerLigneRouge()
nbClic = 1

mouseMouve / o
nbClic = 0;

mouseMove /
drawGhost()
drawRedLine()
nbClic = nbClic

nbPoints = 0

REPOS

RightClic / o
effacer()
drawRedLines
nbClic = 0

1CLIC

space / o
nbClic = 0

RightClic / o
nbClic = 0

RightClick / nbClic == 2
supprimerPoint();
drawGhost();
drawLinesGrey()
drawRedLines()
nbClic --;

Space / o
supprimerPoint();
drawRedLines();
nbClic = 0;

LeftClicked /
ajouterPoint()
dessinerLigneNoir()
dessinerLigneRouge
nbClic = 2

space /
stockerPolyline
drawRedLines
nbClic = 0

MouseMove /
dessinerGhost()
dessinerLignesNoir()
dessinerLigneRouge()
nbClic = nbClic

CLICS

LeftClick /
addPoint();
drawGhost();
drawRedLines():
nbClic ++;

RightClick / nbClic > 2
supprimerPoint();
drawGhost();
drawLinesGrey()
drawRedLines()
nbClic --;

Partie 4 : matrice état/évènements

effacer();
dessinerGhost(point1,point2);
dessinerListeRouge(point1, point2);
dessinerListeNoir(point1, point2);
ajouterPoint(p)
retirerPoint()

stockerPolyline(point1);

	rightClic	LeftClic	MouseMove	SpacePressed

REPOS	s = REPOS;
	nbClic = 0;
		s = ONECLICK;
		nbClic = 1;
		ajouterPoint(p);
			s = REPOS;
			nbClic = 0;
				s = REPOS;
				nbClic = 0;

UNCLIC	s = REPOS
	nbClic=0
	retirerPoint()
	effacer()
	dessinerLignesRouges()
		s = CLICS
		nbClics++
		effacer()
		ajouterPoint(p)
		dessinerGhost()
		dessinerLignesNoires()
		dessinerLignesRouges();
			s = UNCLIC;
			nbClic = nbClic;
			effacer()
			dessinerGhost();
			dessinerLignesRouges();
				s = REPOS;
				nbClic = 0;
				effacer()
				supprimerPoint();
				dessinerLignesRouges();

CLICS	```if(nbClic > 2){```
	```s = CLICS```
	```nbClic --```
	```suppPoint()```
	```dessinerGhost()```
	```effacer()```
	```drwnRedLine()```
	```drawGhostLines()```
	```}```
	```else {```
	```s = 1CLIC```
	```nbClic --```
	```suppPoint()```
	```dessinerGhost```
	```drawRedLines```
	```drawGhostLine()```
		s = CLICS
		nbClic ++
		effacer()
		ajouterPoint(p)
		drawGhost()
		dessinerLignesNoires()
		dessinerLignesRouges()
			s = CLICS
			nbClic = nbClic;
			effacer();
			dessinerGhost();
			dessinerGreyLines();
			dessinerLignesRouges();
				S = REPOS;
				nbClic = 0;
				stockerPolyline();
				drawRedLines();

REPOS :
s = REPOS
nbClic = 0;