

VirtuCarrière

Livrable 3

Présenté à Thierry Eude

Par Équipe 09

Matricule	Nom	Courriel
111 254 476	Maxime Miville Deschênes	maxime.miville-deschenes.1@ulaval.ca
111 158 759	Vincent Masse	vincent.masse.4@ulaval.ca
111 238 923	Francis Boulianne	francis.boulianne.2@ulaval.ca
111 240 093	Philippe Vincent	philippe.vincent.3@ulaval.ca

Modèle du domaine

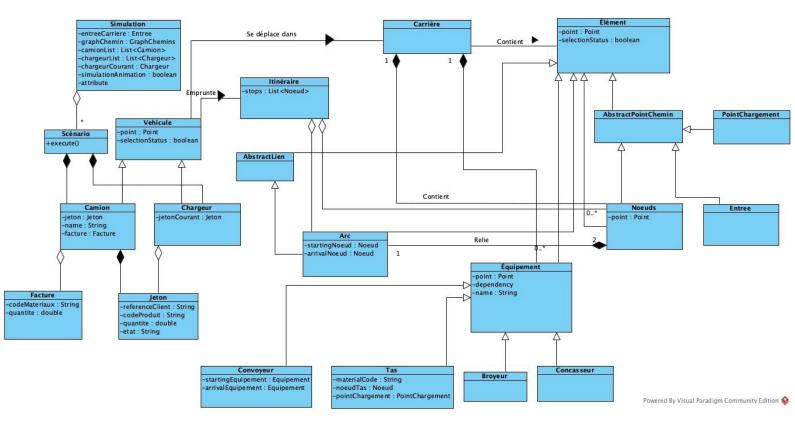
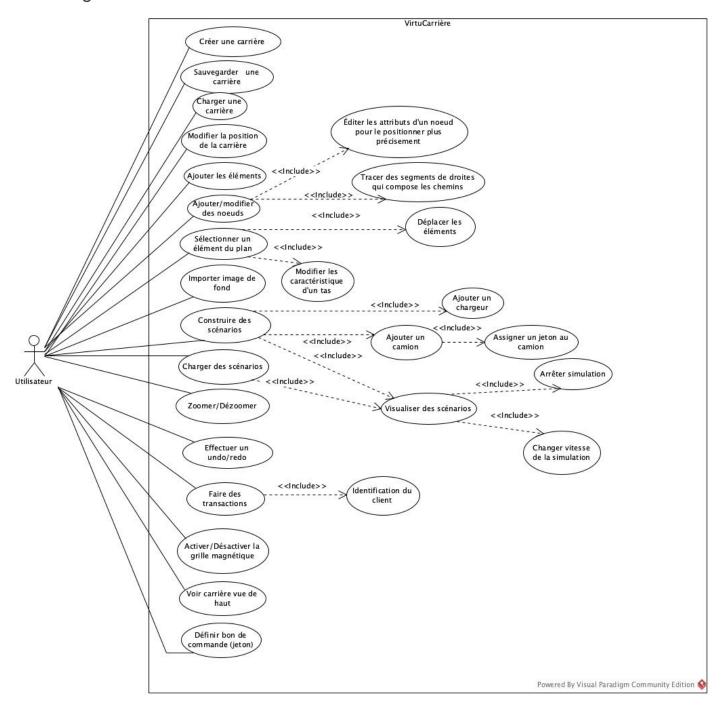


Diagramme des cas d'utilisation



Modèle de conception mis à jour

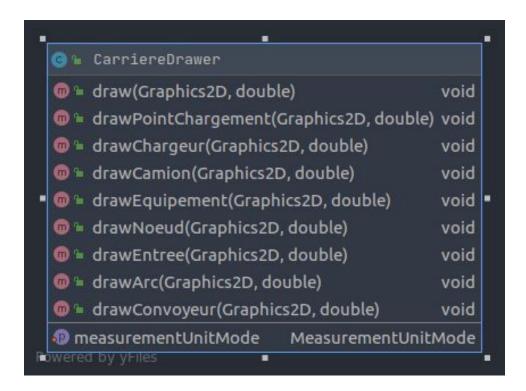
Pour le modèle de conception, nous avons eu l'autorisation de Thierry Eude pour générer notre diagramme de classe grâce à IntelliJ. Nous avons choisi de présenter le modèle de conception par package pour faciliter le visionnement et nous avons inclus le modèle de package à la suite pour permettre de voir la composition de notre application. De plus, nous allons joindre les images ci-dessous en .png pour faciliter votre visionnement, si désiré.

Controller

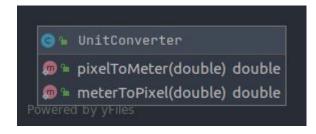
3 ⊾ ElementContainer	
switchSelectionStatus(double, double, boolean)	voic
□ vpdateSelectedItemsPosition(double, double)	voic
□ * trouverChargeurCorrespondant(Tas)	Chargeu
🗑 🎉 Chargeur Chemin To Path (Chargeur, Tas, List < Noe	ud>) Vector <abstractpointchemin< td=""></abstractpointchemin<>
🌚 🛰 noeudSelection(double, double)	void
	Ta
🗓 🦫 cheminDuCamion(Tas)	Vector <abstractpointchemin< td=""></abstractpointchemin<>
isNoeudPresent(Noeud)	void
🍩 🕨 isEquipementPresent(Equipement)	void
🚳 🕒 addArc(Arc)	void
□ ► clearEquipementConv()	vol
addElement(Point, EquipementModes)	volc
	volc
addTas(Point, String)	voic
⊕ removeCamion(Camion)	voic
addCamion(Point, String, String, double)	volc
	volc
removeArc(Arc)	voic
□ = removeConvoyeur(Convoyeur)	voic
addChargeur(Point)	volc
■ EditCamion(Camion, String, String, double)	voic
⊕ generateFacture(Camion)	voic
□ = verificationJeton(Camion, Chargeur)	boolear
h cheminDuCamionRetour(Tas)	Vector <abstractpointchemin< td=""></abstractpointchemin<>
⊕ removeChargeur(Chargeur)	voic
n genererFacture(Camion)	Facture
□ vreateToken(String, String, double)	voic
□ removeEquipement(Equipement)	volc
addEquipement(Equipement)	voic
□ * removeNoeud(AbstractPointChemin)	voic
snapSelectedElementToGrid(double)	voic
startSimulation()	voic
closeSimulation()	voic
Closesimulation() * getElement(Point)	Elemen
validateDependencies()	boolear
D entree	Entre
D convoyeurList	ArrayList <list<convoyeur></list<convoyeur>
🛮 simulationSpeed 🗈 arcList	double
	ArrayList <list<arc></list<arc>
D file	File
netreSimulation	Entre
D graphChemin	GraphChemin
simulationAnimation	boolear
D allNoeuds	List <noeud< td=""></noeud<>
D camionList	List <camion:< td=""></camion:<>
equipementList	List <equipement< td=""></equipement<>
graphCheminSimulation	GraphChemin
nchargeurList	List <chargeur< td=""></chargeur<>
noeudForArcList	List <noeud:< td=""></noeud:<>
D pointsForArcList	List <abstractpointchemin:< td=""></abstractpointchemin:<>
D noeudList	List <abstractpointchemin< td=""></abstractpointchemin<>

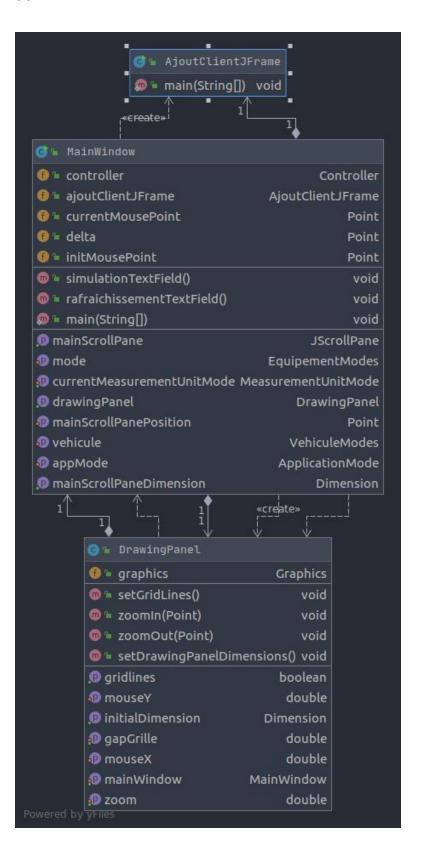
□ s addConvoyeur(Point, EquipementModes) void □ addCornoveur(Point, EquipementModes) void □ addCornoveur(Point, EquipementModes) void □ addCornoveur(Point, EquipementModes) void □ addCornoveur(Point, EquipementModes) void □ TrouverTasCorrespondant(List <tas>, String) Tas □ EditCamion(Camion, String, String, double) void □ cheminDuCamion(Tas) Vector<abstractpointchemin> □ addChargeur(Point) void □ addCamion(Point, String, String, double) void □ cheminDuCamion(Tas) Vector<abstractpointchemin> □ genererFacture(Camion) Facture □ createrOsen(String, String, double) void □ removeChargeur(Camion) Facture □ createrOsen(String, String, double) void □ removeChargeur(Chargeur) void □ cleatEquipementConv() void □ double() void □ addCar(Deint, EquipementModes, Point) void □ addTas(Point, String) void □ addTas(Point, EquipementModes) void □ addTas(Point, EquipementModes) void □ addArc(Point, EquipementModes) void □ addArc(Point, EquipementModes) void □ removeCanvoyeur(Convoyeur) void □ removeArc(Arc) void □ removeArc(Arc) void □ void void void void void void void void</abstractpointchemin></abstractpointchemin></tas>		•
■ addCcrible(Point, EquipementModes) void ■ addBConcasseur(Point, EquipementModes) void ■ addBConcasseur(Point, EquipementModes) void ■ TrouverTasCorrespondant(List-Tas>, String) Tas ■ EditCamion(Camion, String, double) void ■ cheminDuCamion(Tas) Vector <abstractpointchemin> ■ addChargeur(Point) void ■ addChargeur(Point) void ■ addChargeur(Point) void ■ cheminDuCamionRetour(Tas) Vector<abstractpointchemin> ■ addChargeur(Camion) Forture ■ createToken(String, String, double) void ■ createToken(String, String, double) void ■ removeChargeur(Chargeur) void ■ removeChargeur(Chargeur) void ■ clearEquipementConv() void ■ clearEquipementConv() void ■ clearEquipementConv() void ■ choisirChargeurCorrespondant(Tas) Vector<abstractpointchemin> ■ verificationJeton(Camion, Chargeur) void ■ addTas(pulpementMondes, Point) void ■ addTas(pulpementMondes, Point) void ■ addTas(Point, EquipementModes, Point) void ■ addTas(Point, EquipementModes, Point) void ■ changementSelectionStatus(double, double) void ■ removeArc(Arc) void ■ addErtere(Point, EquipementModes) void ■ removeArc(arc) void ■ removeArc(arc)</abstractpointchemin></abstractpointchemin></abstractpointchemin>	O w Controller	
addConcasseur(Point, EquipementModes) addBroyeur(Point, EquipementModes) addBroyeur(Point, EquipementModes) browerTasCorrespondant(List-Tas, String) cheminDuCamion(Tas) cheminDuCamion(Tas) cheminDuCamion(Tas) cheminDuCamion(Tas) cheminDuCamion(Tas) cheminDuCamion(Tas) cheminDuCamion(Retour(Tas) cheminDuCamionRetour(Tas) cheminDuCamionRetour(Tas) cheminDuCamionRetour(Tas) cheminDuCamionRetour(Tas) cheminDuCamionRetour(Tas) cheminDuCamionRetour(Tas) createToken(String, String, double) cholsinChargeur(Camion) cholsinChargeur(Tas) chargeurCheminToPath(Chargeur, Tas, List-Noeud-) cholsinChargeurCheminToPath(Chargeur, Tas, List-Noeud-) cholsinChargeurCheminConvolum (Tas) chargeurCheminToPath(Chargeur, Tas, List-Noeud-) cholsinChargeurCheminChemin chemincolum (Tas) chargeurCheminToPath(Chargeur, Tas, List-Noeud-) cholsinChargeurCheminToPath(Chargeur, Tas, List-Noeud-) cholsinChargeurCheminToPath(Chargeur, Tas, List-Noeud-) cholsinChargeurCheminToPath(Chargeur, Tas, List-Noeud-) cholsinChargeurCheminDouloud chargeurLiguipementModes, Point) chargeurLiguipementModes, Point) chargeurLiguipementModes, Point, AbstractPointChemin) chargeurLiguipementModes, Point, AbstractPointChemin) chargeurLiguipementModes, Point, AbstractPointChemin, Void chargeurLiguipementModes, Point, AbstractPoin	■ addConvoyeur(Point, Equipement)	Aodes) void
■ * addBroyeur(Point, EquipementModes) void ● * TrouverFasCorrespondant(List-fas>, String) Tas ● * EditCamion(Camion, String, String, double) void ● * addCamion(Point, String, String, double) void ● * addCamion(Point, String, String, double) void ● * addCamion(Point, String, String, double) void ● * deminDuCamionRetour(Tas) Vector <abstractpointchemin> ● * cereateToken(String, String, double) void ● * removeFacture(Chargeur) void ● * removeCamion(Camion) void ● * removeCamion(Camion) void ● * removeCamion(Camion) void ● * clearEquipementConvel void ● * clearEquipementConvel void ● * chargeurchemin ToPath(Chargeur) void • chargeurchemin ToPath(Chargeur) void • chargeurchemin ToPath(Chargeur) void • chargeurchemin ToPath(Chargeur) void • addTas(Point, String) void • addTas(Point, String) void • addTas(Point, String) void • addTas(Point, String) void <</abstractpointchemin>	addCrible(Point, EquipementMode)	s) void
■ * TrouverTasCorrespondant(List <tas>, String) Tas □ * EditCamion(Camion, String, String, double) voctor<abstractpointchemin> □ * AddChargeur(Point) void □ * addChargeur(Point) void □ * addChargeur(Point) void □ * demino(Point, String, String, double) void □ * createToken(String, String, double) void □ * clearEquipementConv() void □ * clearEquipementConv() void □ * choistirChargeurCorrespondant(Tas) Chargeur □ * verification.eton(camion, Chargeur) boolean □ * verification.eton(camion, Chargeur) void □ * verification.eton(camion, Chargeur) void □ * addTacyBeint, Estring void □ * addTacyBeint, Estring void □ * addTacyBeint, Estring void □ * camouveEntre(Entre)</abstractpointchemin></tas>	addConcasseur(Point, Equipement)	Modes) void
■ * EditCamion(Camion, String, String, double) void	🐵 > addBroyeur(Point, EquipementMod	des) void
deminDuCamion(Tas) Vector <abstractpointchemino addchargeur(point)="" cheminducamionretour(tas)="" clearequipement(equipement,="" clearequipementconv()="" createtoken(string,="" de="" double)="" facture="" genererfacture(camion)="" list<noeud="" removechargeur(chargeur)="" string,="" tas,="" vector<abstractpointchemino="" void="">) Vector<abstractpointchemin> de clearEquipement(Equipement) void de clearEquipement(Equipement) void de void void void de addEquipement(EquipementModes, Point) void de addRac(Point, EquipementModes, Point) void de addArc(Point, EquipementModes) void de addArc(Point, AbstractPointChemin, AbstractPointChemin) void de removeNoeud(AbstractPointChemin) void de removeRort(Arc) void de removeRort(Arc) void de removeRort(Arc) void de switchSelectionStatus(double, double, boolean) void de void void void void de void void void void de redo() void de</abstractpointchemin></abstractpointchemino>	⊕ TrouverTasCorrespondant(List <tas< p=""></tas<>	>, String) Tas
addCarrgeur(Point, String, String, double) addCamion(Point, String, String, double) chemin DuCamionRetour(Tas) genererFacture(Camion) createrToken(String, String, double) createrToken(String) clearEquipementConv() clearEquipementConv() clearEquipementConv() clearEquipementConv() clearEquipementConv() clearEquipementConv() clearEquipement(Equipement) clearCanlon, clearEquipement clearEquipement(Equipement) clearCanlon, clearEquipement clearEquipement(Equipement) clearEquipement(Equipement) clearEquipement(Equipement) clearEquipement(Equipement) clearEquipement(Equipement) clearEquipement(Equip		double) void
o → addCamion(Point, String, String, double)		
e > cheminDuCamionRetour(Tas)		
□ = genererFacture(Camion) Facture □ = createToken(String, String, double) void □ = removeCamion(Camion) void □ = choseCamion(Camion) void □ = chargeurentCorrespondant(Tas) Chargeur □ = chargeurChemin ToPath(chargeur, Tas, List<-Noeud>) Vector<-AbstractPointChemin> □ = chargeurChemin ToPath(chargeur, Tas, List<-Noeud>) Vector<-AbstractPointChemin> □ = chargeurChemin ToPath(Chargeur) boolean □ = removeEquipement(Equipement) void □ = addEquipement(Equipement) void □ = addEquipement(EquipementModes) void □ = addRay(Point, String) void □ = addRay(Point, EquipementModes) void □ changementSelectionStatus(double, double) void □ = removeNeueud(AbstractPointChemin) void □ = removeArc(Arc) void </td <td></td> <td></td>		
createToken(String, String, double) removeChargeur(Chargeur) removeChargeur(Chargeur) removeCamion(Camion) clearEquipementConv() choisirChargeurCorrespondant(Tas) choisirCharge		
o removeChargeur(Chargeur) void o removeCamion(camion) void o clearEquipementConv() void o clearEquipementConv() void o choisirChargeurCorrespondant(Tas) Chargeur o chargeurCheminToPath(Chargeur, Tas, List≺Noeud>) Vector≺AbstractPointChemin> o verification.Jeton(Camion, chargeur) boolean o removeEquipement(Equipement) void o addTas(Point, String) void o addTas(Point, String) void o addTas(Point, String) void o addTas(Point, String) void o addArc(Point, EquipementModes) void o addArc(Point, EquipementModes) void o addArc(Point, EquipementModes) void o removeNoeud(AbstractPointChemin, AbstractPointChemin) void o removeConvoyeur(Convoyeur) void o removeConvoyeur(Convoyeur) void o removeEntree(Entree) void o switchSelectionStatus(double, double, boolean) void o noeudSelection(double, double, boolean) void o noeudSelection(double, double, double) void o poenFile() void o saveAs() void o saveAs() void o asaveAs() vo		
clearEquipementConv() void clearEquipementConv()		
o clearEquipementConv()		
Chargeur Chargeur Correspondant (Tas) Chargeur Chargeur Chargeur Chargeur Chargeur Chargeur Chargeur (Chargeur) boolean chargeur chargeur) boolean void verification Jeton (Camion, Chargeur) boolean void addTas(Point, String) void addArc(Point, EquipementModes) void addArc(Point, AbstractPointChemin, AbstractPointChemin) void removeNoeud(AbstractPointChemin, AbstractPointChemin) void removeNoeud(AbstractPointChemin) void addEntree(Point, EquipementModes) void addEntree(Point, EquipementModes) void sermoveConvoyeur(Convoyeur) void addEntree(Point, EquipementModes) void void visuaddEntree(Point, EquipementModes) void void visuaddEntree(PointModes) void void void void void visuaddEntree(PointModes) void void void void void void void void		
ChargeurCheminToPath(Chargeur, Tas, List <noeud>) Vector<abstractpointchemins abstractpointchemin)="" abstractpointchemin,="" addarc(point,="" addentree(point,="" addnoeud(point,="" addtas(point,="" boolean="" boolean)="" changementselectionstatus(double,="" chargeur)="" chasesimulation()="" double)="" double,="" equipementmodes)="" parties(point,="" redo()="" redo(<="" removeconvoyeur(convoyeur)="" removeentree(entree)="" removeequipement(equipement)="" removenoeud(abstractpointchemin,="" string)="" switchselectionstatus(double,="" td="" verificationjeton(camion,="" vindateselection(double,="" void=""><td></td><td></td></abstractpointchemins></noeud>		
□ = verificationJeton(Camion, Chargeur) boolean □ = removeEquipement(Equipement) void □ = addEquipement(EquipementModes, Point) void □ = addTas(Point, String) void □ = addTas(Point, EquipementModes) void □ = changementSelectionStatus(double, double) void □ = removeNoeud(AbstractPointChemin, AbstractPointChemin) void □ = removeArc(Arc) void □ = removeConvoyeur(Convoyeur) void □ = removeEntree(Point, EquipementModes) void □ = removeEntree(Entree) void □ = switchSelectionStatus(double, double, boolean) void □ = void selection(double, double, double, boolean) void □ = void saveAs() void □ = vo		
o = removeEquipement(Equipement) void o = addEquipement(EquipementModes, Point) void o = addTas(Point, String) void o = addTas(Point, String) void o = addNoeud(Point, EquipementModes) void o = addNoeud(Point, EquipementModes) void o = changementSelectionStatus(double, double) void o = addArc(Point, AbstractPointChemin, AbstractPointChemin) void o = removeNoeud(AbstractPointChemin) void o = removeRoeud(AbstractPointChemin) void o = removeConvoyeur(Convoyeur) void o = removeEntree(Entree) void o = removeEntree(Entree) void o = removeEntree(Entree) void o = switchSelectionStatus(double, double, boolean) void o = switchSelectionStatus(double, double, boolean) void o = void void void o = void void o = void void o = void void void void o = void void void o = void = vo		
addTas(Point, String) void addTas(Point, String) void addTas(Point, EquipementModes) void addNoeud(Point, EquipementModes) void addNoeud(Point, EquipementModes) void addNoeud(Point, AbstractPointChemin, AbstractPointChemin) void addArc(Point, AbstractPointChemin, AbstractPointChemin) void removeNoeud(AbstractPointChemin) void removeArc(Arc) void aremoveConvoyeur(Convoyeur) void wremoveEntree(Point, EquipementModes) void wremoveEntree(Entree) void wremoveEntree(Entree) void wremoveEntree(Entree) void wroudateSelectionStatus(double, double, boolean) void wroudateSelection(double, double) void wroudateSelecteditemsPositions(double, double, doub		
o = addTas(Point, String) void o = addNoeud(Point, EquipementModes) void o = changementSelectionStatus(double, double) void o = changementSelectionStatus(double, double) void o = removeNoeud(AbstractPointChemin) void o = removeNoeud(AbstractPointChemin) void o = removeNoeud(AbstractPointChemin) void o = removeConvoyeur(Convoyeur) void o = addEntree(Point, EquipementModes) void o = addEntree(Point, EquipementModes) void o = switchSelectionStatus(double, double, boolean) void o = switchSelectionStatus(double, double, boolean) void o = openFile() void o = openFile() void o = saveAs()		
o = addNoeud(Point, EquipementModes)		
changementSelectionStatus(double, double) addArc(Point, AbstractPointChemin, AbstractPointChemin) removeNoeud(AbstractPointChemin) removeArc(Arc) removeConvoyeur(Convoyeur) addEntree(Point, EquipementModes) removeEntree(Entree) void removeEntree(Entree) void removeSelectionStatus(double, double, boolean) roud r		
addArc(Point, AbstractPointChemin, AbstractPointChemin) removeNoeud(AbstractPointChemin) removeArc(Arc) removeConvoyeur(Convoyeur) removeConvoyeur(Convoyeur) removeEntree(Entree) redid redid removeEntree(Entree) redid redid removeEntree(Entree) redid redid removeEntree(Entree) redid r		
□ = removeNoeud(AbstractPointChemin) void □ = removeArc(Arc) void □ = removeConvoyeur(Convoyeur) void □ = addEntree(Point, EquipementModes) void □ = removeEntree(Entree) void □ = switchSelectionStatus(double, double, boolean) void □ = noeudSelection(double, double) void □ = noeudSelecteditemsPositions(double, double) void □ = void void □ = void void □ = saveAs() void □ =		
□ = removeArc(Arc) void □ = removeConvoyeur(Convoyeur) void □ = addEntree(Point, EquipementModes) void □ = removeEntree(Entree) void □ = switchSelectionStatus(double, double, boolean) void □ = noeudSelection(double, double) void □ = noeudSelecteditemsPositions(double, double) void □ = void = void void □ = void = void void □ = saveAs() void □ = void = void void <td></td> <td></td>		
∅ * removeConvoyeur(Convoyeur) void ∅ * addEntree(Point, EquipementModes) void ∅ * removeEntree(Entree) void ∅ * switchSelectionStatus(double, double, boolean) void ∅ * updateSelecteditemsPositions(double, double) void ∅ * updateSelecteditemsPositions(double, double) void ∅ * saveAs() void ∅ * saveAs() void ∅ * saveSol void ∅ * newProject() void ∅ * newProject() void ∅ * validateOependencies() boolean ∅ * validateOependencies() boolean ∅ * void void ∅ * validateOependencies() void ∅ * redo() void ∅ * redo() void ∅ * redo() void ∅ * redo() void ∅ * retoe() void Ø * retoe() void </td <td></td> <td></td>		
addEntree(Point, EquipementModes) removeEntree(Entree) void switchSelectionStatus(double, double, boolean) switchSelectionStatus(double, double) noeudSelectCon(double, double) void supdateSelectedItemsPositions(double, double) saveAs() saveAs() saveAs() saveAs() saveAs() saveAs() saveAs() saveAs() saveO saveAs() sudd saveAs() sudd saveAs() sudd saveAs() sudd sudd sudd sudd sudd sudd sudd sud		
o removeEntree(Entree) vold o switchSelectionStatus(double, double, boolean) vold o noeudSelection(double, double) vold o updateSelectedItemsPositions(double, double) vold o vopenFile() vold o vopenFile() vold o saveAs() vold o saveAs() vold o saveAs() vold o vold o newProject() vold o vold o validateOependencies() vold o		es) void
□ ** noeudSelection(double, double) void □ ** updateSelectedItemsPositions(double, double) void □ ** openFile() void □ ** saveAs() void □ ** saveQ) void □ ** newProject() void □ ** validation() void □ ** validateDependencies() boolean □ ** validateDependencies() void		
□ vupdateSelecteditemsPositions(double, double) void □ vopenFile() void □ save() void □ newProject() void □ startSimulation() void □ validateDependencies() boolean □ validateOependencies() boolean □ validateOependencies() void □ entree Entree □ convoyeurList ArrayList <list<<convoyeur> double □ arrayList ArrayList<<list<<noeudence< td=""> □ graphChemin GraphChemins □ convoyeurList List<<noeudence< td=""> □ allNoeuds List<<convoyeur< td=""> □ allNoeuds List<<noeudence< td=""> □ convoyeurList</noeudence<></convoyeur<></noeudence<></list<<noeudence<></list<<convoyeur>		ble, boolean) void
⊕ vopenFile() void ⊕ vaveAs() void ⊕ vave() void ⊕ validester() void ⊕ validateDependencies() boolean ⊕ validateDependencies() boolean ⊕ validateDependencies() void ⊕ validateDependencies() boolean ⊕ validateDependencies() void ⊕ validateDependencies() void ⊕ validateDependencies() boolean ⊕ validateDependencies() void ⊕ validateDependencies() ArrayList <list<onvoyeurs< t<="" td=""><td>□ noeudSelection(double, double)</td><td>void</td></list<onvoyeurs<>	□ noeudSelection(double, double)	void
saveAs() void save() void newProject() void startSimulation() void void void void void void void void	□ updateSelectedItemsPositions(dou	ble, double) void
⊚ * save() void ⊚ * newProject() void ⊚ * startSimulation() void ⊚ * closeSimulation() void ⊚ * validateDependencies() boolean ⊚ * undo() void ⊚ * redo() void Ø entree Entree Ø convoyeurList ArrayList <list<convoyeur>> Ø element ElementContainer Ø simulationSpeed double Ø arcList ArrayList<list<arc>> Ø entreSimulation Entree Ø graphChemin GraphChemins Ø simulationAnimation boolean Ø simulationAnimation boolean Ø camionList List<camion> Ø equipementList List<noeud> Ø graphCheminSimulation GraphChemins Ø chargeurList List<chargeur> Ø noeudForArcList List<noeud> Ø pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></camion></list<arc></list<convoyeur>	@ ≥ openFile()	void
Image: startSimulation () void Image: startSimulation () simulationTotalist Image: startSimulation () Image: startSimulation () Image: start () Image: startSimulation () Image: startSimulation () Image: startSimulation () Image: start () Image: startSimulation () Image: start () Image: startSimulation () Image: start () Image: start () <	@ ≥ saveAs()	void
⊕ * startSimulation() void ⊕ * closeSimulation() void ⊕ * validateDependencies() boolean ⊕ * undo() void ⊕ * redo() void Ø entree Entree Ø convoyeurList ArrayList <list<convoyeur>> Ø element ElementContainer Ø simulationSpeed double Ø arcList ArrayList<list<arc>> Ø entreSimulation Entree Ø graphChemin GraphChemins Ø simulationAnimation boolean Ø camionList List<camion> Ø equipementList List<equipement> Ø allNoeuds List<noeud> Ø graphCheminSimulation GraphChemins Ø chargeurList List<chargeur> Ø noeudForArcList List<noeud> Ø pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion></list<arc></list<convoyeur>		void
© * closeSimulation() void © * validateDependencies() boolean © * undo() void © * redo() void © entree Entree © convoyeurList ArrayList <list<convoyeur>> © element ElementContainer © simulationSpeed double © arcList ArrayList<list<arc>> © entreSimulation Entree © graphChemin GraphChemins © simulationAnimation boolean © camionList List<camion> © equipementList List<equipement> © allNoeuds List<noeud> © graphCheminSimulation GraphChemins © chargeurList List<chargeur> © noeudForArcList List<noeud> © noeudForArcList List<noeud> © pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></noeud></chargeur></noeud></equipement></camion></list<arc></list<convoyeur>	⊕ newProject()	void
⊕ * validateDependencies() boolean ⊕ * undo() void ⊕ * redo() void ⊕ entree Entree ⊕ convoyeurList ArrayList <list<convoyeur>> ⊕ element ElementContainer ⊕ simulationSpeed double ⊕ arcList ArrayList<list<arc>> ⊕ entreSimulation Entree ⊕ graphChemin GraphChemins ⊕ simulationAnimation boolean ⊕ camionList List<camion> ⊕ equipementList List<equipement> ⊕ allNoeuds List<noeud> ⊕ graphCheminSimulation GraphChemins ⊕ chargeurList List<chargeur> ⊕ noeudForArcList List<noeud> ⊕ pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion></list<arc></list<convoyeur>		void
⊕ * undo() void ⊕ * redo() void ⊕ entree Entree ⊕ convoyeurList ArrayList <list<convoyeur>> ⊕ element ElementContainer ⊕ simulationSpeed double ⊕ arcList ArrayList<list<arc>> ⊕ entreSimulation Entree ⊕ graphChemin GraphChemins ⊕ simulationAnimation boolean ⊕ camionList List<camion> ⊕ equipementList List<equipement> ⊕ allNoeuds List<noeud> ⊕ graphCheminSimulation GraphChemins ⊕ chargeurList List<chargeur> ⊕ noeudForArcList List<noeud> ⊕ pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion></list<arc></list<convoyeur>		void
© > redo() void ① entree Entree ② convoyeurList ArrayList <list<convoyeur>> ③ element ElementContainer ⑤ simulationSpeed double ⑤ arcList ArrayList<list<arc>> ⑤ entreSimulation Entree ⑤ graphChemin GraphChemins ⑥ simulationAnimation boolean ⑥ camionList List<camion> ⑥ equipementList List<equipement> ⑤ allNoeuds List<noeud> ⑥ graphChemins GraphChemins ⑥ chargeurList List<chargeur> ⑥ noeudForArcList List<abstractpointchemin></abstractpointchemin></chargeur></noeud></equipement></camion></list<arc></list<convoyeur>	□ validateDependencies()	boolean
## Dentree ## Entree ## ConvoyeurList ## ArrayList <list<convoyeur>> ## Delement ## Delemen</list<convoyeur>	🙃 🖘 undo()	void
⑤ convoyeurList ArrayList <list<convoyeur>> ⑤ element ElementContainer ⑥ simulationSpeed double ⑥ arcList ArrayList<list<arc>> ⑥ entreSimulation Entree ⑤ graphChemin GraphChemins ⑥ simulationAnimation boolean ⑥ camionList List<camion> ⑨ equipementList List<equipement> ⑨ allNoeuds List<noeud> ⑨ graphCheminSimulation GraphChemins ⑨ chargeurList List<chargeur> ⑨ noeudForArcList List<noeud> ⑨ pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion></list<arc></list<convoyeur>	® ≒ redo()	void
① element ElementContainer ① simulationSpeed double ② arcList ArrayList <list<arc>> ② entreSimulation Entree ② graphChemin GraphChemins ③ simulationAnimation boolean ② camionList List<camion> ② equipementList List<equipement> ③ allNoeuds List<noeud> ④ graphCheminSimulation GraphChemins ④ chargeurList List<chargeur> ⑤ noeudForArcList List<noeud> ⑤ pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion></list<arc>	.0 entree	Entree
(1) simulationSpeed double (2) arcList ArrayList <list<arc> (3) entreSimulation Entree (3) graphChemin GraphChemins (3) simulationAnimation boolean (3) camionList List<camion> (4) equipementList List<equipement> (5) allNoeuds List<noeud> (6) graphCheminSimulation GraphChemins (7) chargeurList List<chargeur> List<noeud> (8) pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion></list<arc>	@ convoyeurList	ArrayList <list<convoyeur>></list<convoyeur>
① arcList ArrayList <list<arc> ② entreSimulation Entree ② graphChemin GraphChemins ③ simulationAnimation boolean ② camionList List<camion> ⑨ equipementList List<equipement> ⑨ allNoeuds List<noeud> ⑨ graphCheminSimulation GraphChemins ⑨ chargeurList List<chargeur> ⑨ noeudForArcList List<abstractpointchemin> ⑨ pointsForArcList List<abstractpointchemin></abstractpointchemin></abstractpointchemin></chargeur></noeud></equipement></camion></list<arc>		
10 entreSimulation Entree 10 graphChemin GraphChemins 10 simulationAnimation boolean 10 camionList List <camion> 10 equipementList List<equipement> 10 allNoeuds List<noeud> 10 graphCheminSimulation GraphChemins 10 chargeurList List<chargeur> 10 noeudForArcList List<noeud> 10 pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion>		
graphChemin GraphChemins simulationAnimation boolean camionList List <camion> equipementList List<equipement> aliNoeuds List<noeud> graphCheminSimulation GraphChemins chargeurList List<chargeur> noeudForArcList List<abstractpointchemin></abstractpointchemin></chargeur></noeud></equipement></camion>		
(1) simulationAnimation boolean (2) camionList List <camion> (3) equipementList List<equipement> (3) allNoeuds List<noeud> (4) graphCheminSimulation GraphChemins (5) chargeurList List<chargeur> (6) noeudForArcList List<noeud> (7) pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud></equipement></camion>		
© camionList List <camion> © equipementList List<equipement> © allNoeuds List<noeud> © graphCheminSimulation GraphChemins © chargeurList List<chargeur> © noeudForArcList List<abstractpointchemin></abstractpointchemin></chargeur></noeud></equipement></camion>		
① equipementList List <equipement> ② allNoeuds List<noeud> ③ graphCheminSimulation GraphChemins ② chargeurList List<chargeur> ② noeudForArcList List<abstractpointchemin></abstractpointchemin></chargeur></noeud></equipement>		
● allNoeuds List <noeud> ● graphCheminSimulation GraphChemins ● chargeurList List<chargeur> ● noeudForArcList List<noeud> ● pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur></noeud>		
graphCheminSimulation GraphChemins List <chargeur> noeudForArcList List<noeud> pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur>		
ChargeurList List <chargeur> noeudForArcList List<noeud> pointsForArcList List<abstractpointchemin></abstractpointchemin></noeud></chargeur>		
pointsForArcList List <abstractpointchemin></abstractpointchemin>		
LIST SADST ACT POINT C DEMIN		
werea by yours	owered by yours	List <abstractpointcnemin></abstractpointcnemin>

Drawing

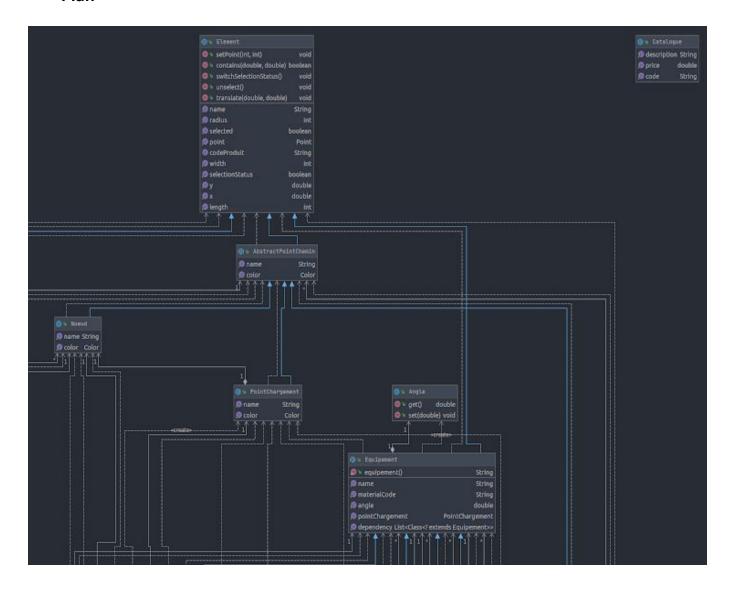


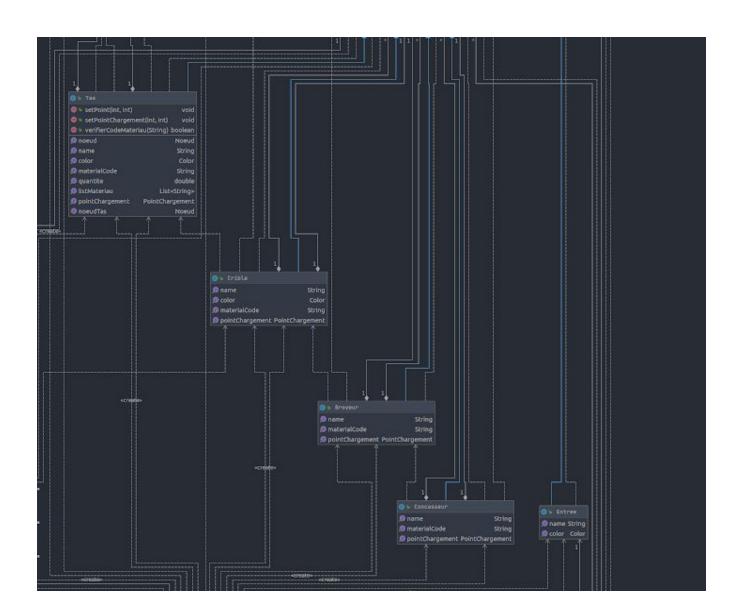
AffichageUtil

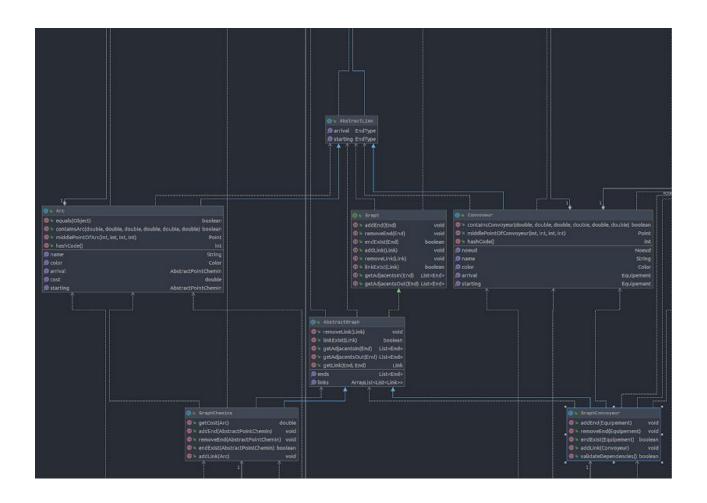




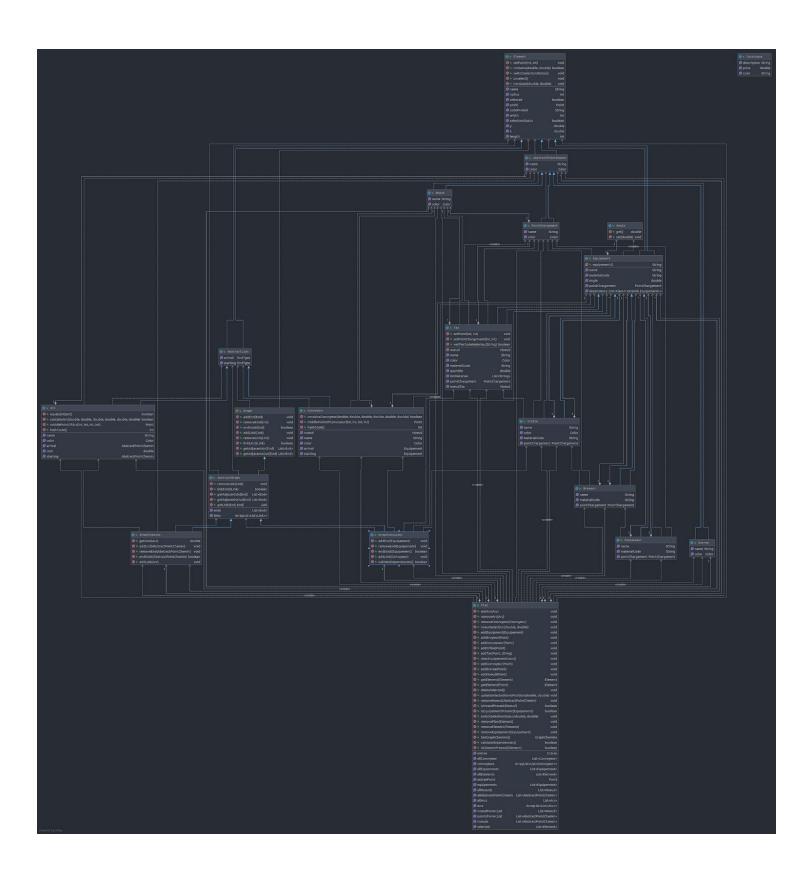
Plan



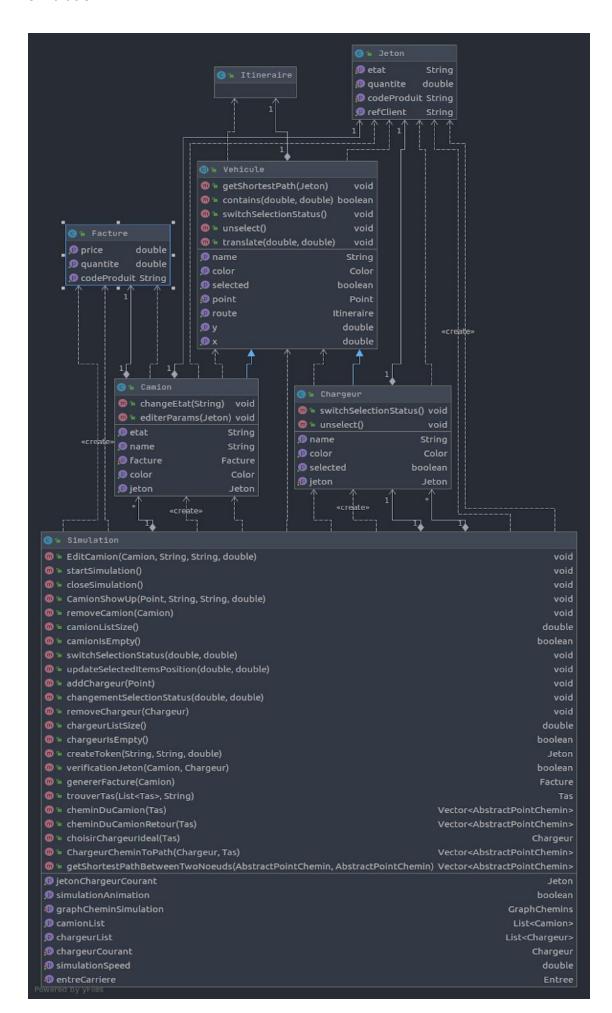




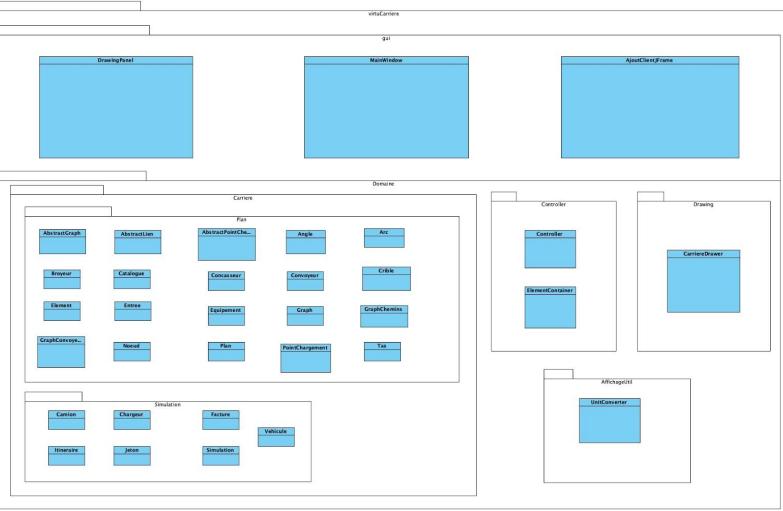
			nse-site-	
noreaton		11117	40%	ete- -create
			12	RIBIOGO DESIGNA
	a → Plan		A Mile Miles	
	addArc(Arc)		void	
	removeArc(Arc)		void	
	removeConvoyeur(Co		vold	
	noeudSelection(doub		void	
	addEquipment(Equip	ementi	void	
	addBroyeur(Point)		void	
	addConcasseur(Point	3	void	
	addCrible(Point)		void	
	 addTas(Point, String) 		biov	
	• clearEquipementCon		void	
	addConvoyeur(Point)		void	
	addEntree(Point)		biov	
	addNoeud(Point)		void	
	getElement(Element)		Element	
	getElement(Point)		Element	
	@ * deleteSelected()		vold	
	updateSelectedItems			
	removeNoeud(Abstra		biov	
	isNoeudPresent(NoeudPresent)		boolean	
	IsEquipementPresent		boolean	
	switchSelectionStatu		void	
	* removePlan(Element)		void	
	removeElement(Elem		vold	
	removeEquipement(I		void	
	GetGraphChemins()		hChemins	
	 validateDependencie 		boolean	
	lsElementPresent(Ele	iment)	boolean	
	@ entree		Entree	
	, all Convoyeur		onvoyeur>	
	convoyeurs	ArrayList <list<co< td=""><td></td><td></td></list<co<>		
	@ allEquipments		ulpement>	
	● alElements	List	<element></element>	
	@ entreePoint		Point	
	go equipements		ulpement»	
	. all Noeuds		st <noeud></noeud>	
	all AbstractPointChemin	List <abstractpoir< td=""><td></td><td></td></abstractpoir<>		
	@ allancs	governo	List-Arc-	
	,0 arcs		List <arc>></arc>	
	noeudForArtList		st <noeud></noeud>	
	pointsForArcList	List <abstractpoir< td=""><td></td><td></td></abstractpoir<>		
	noeuds	List <abstractpoir< td=""><td>ntChemin></td><td></td></abstractpoir<>	ntChemin>	
	selected		<element></element>	



Simulation



Packages



Powered By Visual Paradigm Community Edition

Contribution de chacun des membres de l'équipe

Les différentes tâches du livrable 3 ont tous été complétées individuellement à cause la nature du travail et des conditions sanitaires qui nous obligent à s'isoler. Chacun des membres à contribué au projet sensiblement également. Certains ont plus de "commits" que d'autres, mais c'est en grande partie à la taille des commits. Vous trouverez les contributions de chacun des membres dans le tableau suivant.

Tâches	Personne qui l'a complété
Modèle du domaine	Maxime
Modèle de conception	Vincent
Diagramme des cas d'utilisations	Francis
GUI (Affichage)	Majoritairement Maxime avec l'aide des autres membres
Sauvegarde et ouverture de projets	Francis
Plan (où se retrouve tous les éléments du plan)	Vincent
Algorithme du plus court chemin	Vincent
Grille magnétique et zoom	Maxime
Exécuter une simulation	Philippe
Création d'équipement à l'aide de la souris	Maxime
Création d'un tas à l'aide de la souris	Maxime
Éditer les paramètres d'un matériaux à l'aide du panneau d'édition	Maxime
Sélection/désélection d'équipements, de tas	Maxime
Supprimer/déplacer un équipement/un tas à l'aide de la souris	Maxime
Équipements enchaînés reliés par des convoyeurs	Francis
Lorsqu'on supprimer un équipement, tous les convoyeurs en sortie le seront aussi	Francis
Ajouter un camion	Philippe

Ajouter un chargeur	Philippe
Créer, déplacer et supprimer des noeuds avec la souris	Maxime
Créer des arcs	Francis
Créer des convoyeurs	Francis
Éditer les paramètres d'un camion/chargeur à l'aide du panneau à édition	Maxime