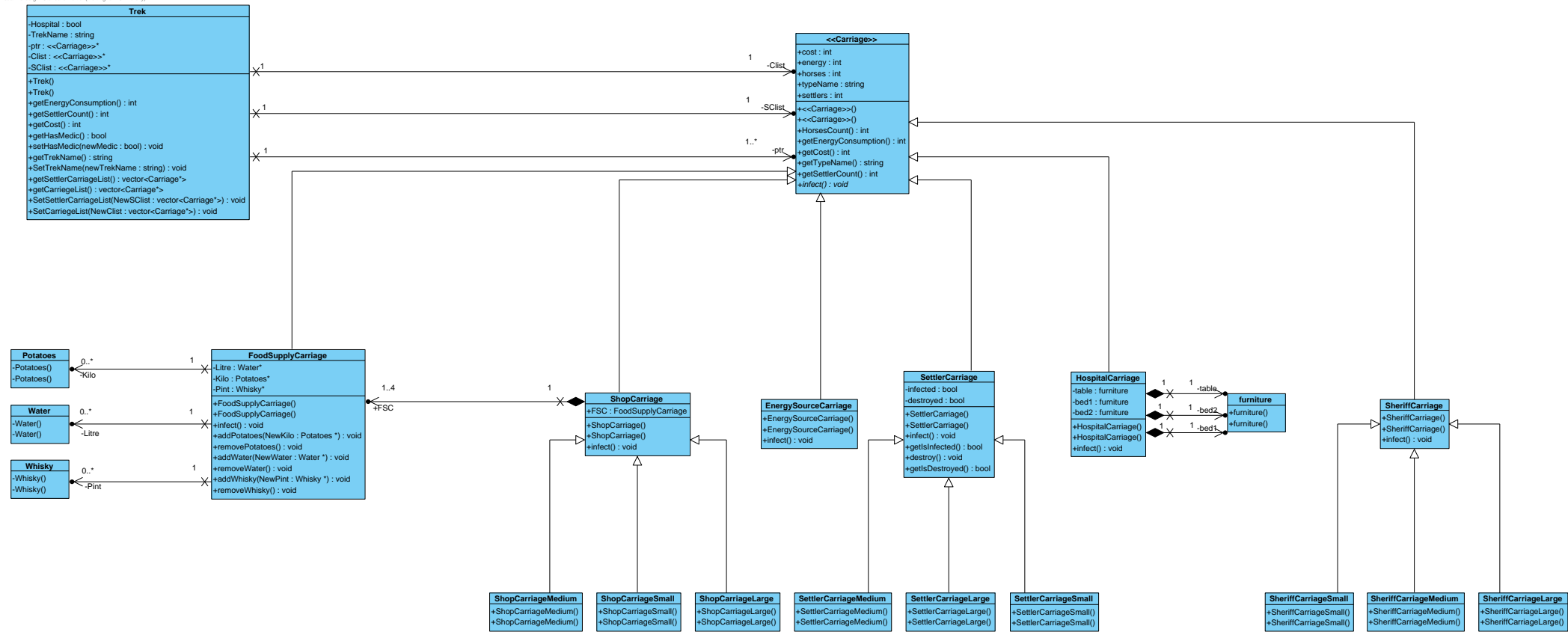# Description of the design.

In the design of this interface for a simulation, it was prioritized order and hierarchy as well as the object oriented principle: easy to maintain, easy to extend. A class was implemented for every different type of carriage. First, it was implemented the abstract class Carriage that contains a pure virtual function to be inherited to the rest of the carriage. This class inherits variables and methods directly to SettlerCarriage, ShopCarriage, FoodSupplyCarriage, EnergySourceCarriage, HospitalCarriage, and SheriffCarriages. A second level of inheritance was implemented from SettlerCarriage to SettlerCarriageSmall, SettlerCarriageMedium, and SettlerCarriageLarge, from ShopCarriage to ShopCarriageSmall, ShopCarriageMedium, and ShopCarriageLarge and finally from SheriffCarriage to SheriffCarriageSmall, SheriffCarriageMedium, and SheriffCarriageLarge. It was an intensive use of inheritance and maybe it could have been used less, for example by using constructors. But the developer considered important to learn and understand the characteristics and properties of the inheritance itself. It was also beneficial for the purpose of diagram order and at the same time simplified the way the information was required by the class trek. It allows to easily change the definition of the methods in case the developer of the simulation decide to change the rules of the game. It was implemented a composition between ShopCarriage and FoodSupplyCarriage to add resources to Shop Carriage. There are aggregations for Potatoes, Water and Whisky to the FoodSypplyCarriage and a composition of two beds and a table to the HospitalCarriage. Finally, The Trek class could call straightforward all the carriage objects through an association with two vectors and a pointer of the class Carriage.

It would be importat to add that at the first attempt all methods were succesfully implemented in Carriage as pure virtual function for considering this more flexible for future modifications but this was changed to fulfill the requirements of the user and at the end only the method infected() was implemented as pure virtual function.

Trek

-Hospital : bool
-TrekName : string
-ptr : <<Carriage>>*
-Clist : <<Carriage>>*
-SClist : <<Carriage>>*

+Trek()
+Trek()
+getEnergyConsumption() : int
+getSettlerCount() : int
+getCost() : int
+getHasMedic() : bool
+setHasMedic(newMedic : bool) : void
+getTrekName() : string
+SetTrekName(newTrekName : string) : void
+getSettlerCarriageList() : vector<Carriage*>
+getCarriegeList() : vector<Carriage*>
+SetSettlerCarriageList(NewSClist : vector<Carriage*>) : void
+SetCarriegeList(NewClist : vector<Carriage*>) : void

<<Carriage>>

+cost : int
+energy : int
+horses : int
+typeName : string
+settlers : int

+<<Carriage>>()
+<<Carriage>>()
+HorsesCount() : int
+getEnergyConsumption() : int
+getCost() : int
+getTypeName() : string
+getSettlerCount() : int
+infect() : void

1   -Clist
1   -SClist
1..*   -ptr

Potatoes
-Potatoes()
-Potatoes()

0..*   -Kilo   1

Water
-Water()
-Water()

0..*   -Litre   1

Whisky
-Whisky()
-Whisky()

0..*   -Pint   1

FoodSupplyCarriage

-Litre : Water*
-Kilo : Potatoes*
-Pint : Whisky*

+FoodSupplyCarriage()
+FoodSupplyCarriage()
+infect() : void
+addPotatoes(NewKilo : Potatoes *) : void
+removePotatoes() : void
+addWater(NewWater : Water *) : void
+removeWater() : void
+addWhisky(NewPint : Whisky *) : void
+removeWhisky() : void

1..4   +FSC   1

ShopCarriage

+FSC : FoodSupplyCarriage

+ShopCarriage()
+ShopCarriage()
+infect() : void

EnergySourceCarriage
+EnergySourceCarriage()
+EnergySourceCarriage()
+infect() : void

SettlerCarriage

-infected : bool
-destroyed : bool

+SettlerCarriage()
+SettlerCarriage()
+infect() : void
+getIsInfected() : bool
+destroy() : void
+getIsDestroyed() : bool

HospitalCarriage

-table : furniture
-bed1 : furniture
-bed2 : furniture

+HospitalCarriage()
+HospitalCarriage()
+infect() : void

1   1   -table
1   1   -bed2
1   1   -bed1

furniture
+furniture()
+furniture()

SheriffCarriage
+SheriffCarriage()
+SheriffCarriage()
+infect() : void

ShopCarriageMedium
+ShopCarriageMedium()
+ShopCarriageMedium()

ShopCarriageSmall
+ShopCarriageSmall()
+ShopCarriageSmall()

ShopCarriageLarge
+ShopCarriageLarge()
+ShopCarriageLarge()

SettlerCarriageMedium
+SettlerCarriageMedium()
+SettlerCarriageMedium()

SettlerCarriageLarge
+SettlerCarriageLarge()
+SettlerCarriageLarge()

SettlerCarriageSmall
+SettlerCarriageSmall()
+SettlerCarriageSmall()

SheriffCarriageSmall
+SheriffCarriageSmall()
+SheriffCarriageSmall()

SheriffCarriageMedium
+SheriffCarriageMedium()
+SheriffCarriageMedium()

SheriffCarriageLarge
+SheriffCarriageLarge()
+SheriffCarriageLarge()

Class specifications.

Class	
Class name:	Trek
Base class:	
Derived class:	
Function of Class:	This class gets and manages the trek data.
Variables(incl.type):	**Description**
Hospital : bool	True if there is hospital, false otherwise
-TrekName : string	Name of the trek
-ptr : Carriage*	Pointer to Carriage objects
-Clist : Carriage*	Carriage vector container
-SClist : Carriage*	Carriage vector container
Operations (incl. Return type)	**Description**
+Trek()	Trek constructor
~Trek()	Trek destructor
+getEnergyConsumption() : int	Returns cumulative energy consumption of trek (carrots/day)
+getSettlerCount() : int	Returns cumulative settler count of the trek
+getCost() : int	Returns cumulative trek cost
+getHasMedic() : bool	Returns "true" if the trek has medic carriage
+setHasMedic(newMedic : bool) : void	Set the variable Hospital: bool
+getTrekName() : string	Returns the chosen name of your trek.
+SetTrekName(newTrekName : string) : void	Set the variable TrekName: string
+getSettlerCarriageList() : vector<Carriage*>	Returns a vector with pointers to all settlers carriages in a trek
+getCarriegeList() : vector<Carriage*>	Returns a vector with pointers to all carriages in a trek
+SetSettlerCarriageList(NewSClist : vector<Carriage*>) : void	Set the variable SClist : vector<Carriage*>)
+SetCarriegeList(NewClist : vector<Carriage*>) : void	Set the variable Clist : vector<Carriage*>)

Class	
Class name:	Abstrac Class: Carriage
Base class:	
Derived class:	SettlerCarriage, SettlerCarriageSmall, SettlerCarriageMedium,SettlerCarriageLarge, FoodSupplyCarriage,HospitalCarriage,ShopCarriage, ShopCarriageLarge, ShopCarriageSmall, ShopCarriageMedium, EnergySourceCarriage, SheriffCarriage, SheriffCarriageSmall, SheriffCarriageMedium, SheriffCarriageLarge.
Function of Class:	Abstrac Class which contains the main methods to inherit.

Variables(incl.type):	Description
+cost: int	This variable gives the information of the cost to the method getCost() for every class that inherit from Carriage.
+energy: int	This variable gives the information of the Energy consumption to the method getEnergyConsuption() for every class that inherit from Carriage.
+horses:int	This variable gives the information of the amount of horses to the method HorsesCount() for every class that inherit from Carriage.
+typeName: string	This variable gives the information of the Name to the method getTypeName() for every class that inherit from Carriage.
+settler:int	This variable gives the information of the amount of settlers to the method getSettlerCount() for every class that inherit from Carriage.
Operations (incl. Return type)	**Description**
+Carriage()	Class constructor
~Carriage()	Class destructor
+HorsesCount() : int	Virtual Function that counts horses
+getEnergyConsumption() : int	Virtual function that gets CarriagesEnergy Consumption
+getCost() : int	Virtual function that gets the cost of the carriages
+getTypeName() : string	Virtual function that gets the name of the carriages
+getSettlerCount() : int	Virtual function tha gest the number of settlers.
+infect():void	Pure virtual function that allows the user changes the state of the carriage from not infected to infected.

Class		
Class name:	SettlerCarriage	
Base class:	Carriage	
Derived class:	SettlerCarriageSmall, SettlerCarriageMedium, SettlerCarriageLarge	
Function of Class:	Generalization of Settler Carriages.	
Variables(incl.type):	Description	
-infected : bool	True if the carriage is infected.	
-destroyed : bool	True if the carriage is destroyed.	
Operations (incl. Return type)	Description	
+SettlerCarriage()	Class constructor	
~SettlerCarriage()	Class destructor	
+infect() : void	True if the carriage is infected	
+getIsInfected() : bool	Get infect() state	
+destroy() : void	True if the carriage is destroyed	
+getIsDestroyed() : bool	Get destroy() state	

Class	
Class name:	SettlerCarriageSmall
Base class:	SettlerCarriage
Derived class:	
Function of Class:	It contains specific characteristics of SettlerCarriageSmall in the constructor.
Variables(incl.type):	**Description**
Operations (incl. Return type)	**Description**
+SettlerCarriageSmall()	Class constructor
~SettlerCarriageSmall()	Class destructor

Class	
Class name:	SettlerCarriageMedium
Base class:	SettlerCarriage
Derived class:	
Function of Class:	It contains specific characteristics of SettlerCarriageMedium in the constructor.
Variables(incl.type):	Description
Operations (incl. Return type)	**Description**
+SettlerCarriageMedium()	Class constructor
~SettlerCarriageMedium()	Class destructor

Class	
Class name:	SettlerCarriageLarge
Base class:	SettlerCarriage
Derived class:	
Function of Class:	It contains specific characteristics of SettlerCarriageLarge the constructor.
Variables(incl.type):	Description
Operations (incl. Return type)	**Description**
+SettlerCarriageLarge()	Class constructor
~SettlerCarriageLarge()	Class destructor

Class	
Class name:	FoodSupplyCarriage
Base class:	Carriage
Derived class:	
Function of Class:	It contains specific characteristics of FoodSupplyCarriage in the constructor.
Variables(incl.type):	Description
Operations (incl. Return type)	**Description**
+FoodSupplyCarriage()	Class constructor

~FoodSupplyCarriage()	Class destructor
+addPotatoes(NewKilo : Potatoes *) : void	Add a pointer Potatoes to the class
+removePotatoes() : void	Gives the value Null to the pointer Potatoes
+addWater(NewWater : Water *) : void	Add a pointer Water to the class
+removeWater() : void	Gives the value Null to the pointer Water
+addWhisky(NewPint : Whisky *) : void	Add a pointer Water to the class
+removeWhisky() : void	Gives the value Null to the pointer Water

Class	
Class name:	ShopCarriage
Base class:	Carriage
Derived class:	ShopCarriageSmall,ShopCarriageMedium,ShopCarriageLarge.
Function of Class:	Generalization of Shop Carriages classes.
Variables(incl.type):	Description
+FSC: FoodSupplyCarriage	Makes a composition for adding horses to FoodSupply Carriage.
Operations (incl. Return type)	Description
+ShopCarriage();	Class constructor
+ShopCarriage();	Class destructor
+infect():void	Infects the ShopCarriage

Class	
Class name:	ShopCarriageSmall
Base class:	ShopCarriage
Derived class:	
Function of Class:	It contains specific characteristics of the ShopCarriageSmall in the constructor.
Variables(incl.type):	It contains specific characteristics of ShopCarriageSmall.
Operations (incl. Return type)	Description
+ShopCarriage();	Class constructor
+ShopCarriage();	Class destructor

Class	
Class name:	ShopCarriageMedium
Base class:	ShopCarriage
Derived class:	
Function of Class:	It contains specific characteristics of ShopCarriageMedium
Variables(incl.type):	Description
Operations (incl. Return type)	Description
+ShopCarriage();	Class constructor
+ShopCarriage();	Class destructor

Class	
Class name:	ShopCarriageLarge.
Base class:	ShopCarriage
Derived class:	
Function of Class:	It contains specific characteristics of ShopCarriageLarge.
Variables(incl.type):	Description
Operations (incl. Return type)	Description
+ShopCarriage();	Class constructor
+ShopCarriage();	Class destructor

Class	
Class name:	EnergySourceCarriage
Base class:	Carriage
Derived class:	
Function of Class:	It contains specific characteristics of EnergySourceCarriage in the constructor.
Variables(incl.type):	Description
Operations (incl. Return type)	**Description**
+EnergySourceCarriage()	Class constructor
~EnergySourceCarriage()	Class destructor
+infect():void	Infects EnergySourceCarriage.

Class	
Class name:	SheriffCarriage
Base class:	Carriage
Derived class:	.
Function of Class:	Generalization of SheriffCarriage classes
Variables(incl.type):	Description
+SheriffCarriage()	Class constructor
~SheriffCarriage()	Class destructor
Operations (incl. Return type)	Description

Class	
Class name:	SheriffCarriageSmall
Base class:	SheriffCarriage
Derived class:	
Function of Class:	It contains specific characteristics of SheriffCarriageSmall in the constructor.
Variables(incl.type):	Description

Operations (incl. Return type)	Description
+SheriffCarriageSmall()	Class constructor
~SheriffCarriageSmall()	Class destructor

Class	
Class name:	SheriffCarriageMedium
Base class:	SheriffCarriage
Derived class:	
Function of Class:	It contains specific characteristics of SheriffCarriageMedium in the constructor.
Variables(incl.type):	Description
Operations (incl. Return type)	Description
+SheriffCarriageMedium()	Class constructor
~SheriffCarriageMedium()	Class destructor

Class	
Class name:	SheriffCarriageLarge
Base class:	SheriffCarriage
Derived class:	
Function of Class:	It contains specific characteristics of SheriffCarriageLarge in the constructor.
Variables(incl.type):	Description
Operations (incl. Return type)	Description
+SheriffCarriageLarge()	Class constructor
~SheriffCarriageLarge()	Class destructor

Class	
Class name:	Water
Base class:	
Derived class:	
Function of Class:	Resources for FoodSupplyCarriage
Variables(incl.type):	Description
Operations (incl. Return type)	Description
-Water()	Constructor
-~Water()	Destructor

Class	
Class name:	Whisky
Base class:	
Derived class:	
Function of Class:	Resources for FoodSupplyCarriage

Variables(incl.type):	Description
Operations (incl. Return type)	Description
-Whisky()	Constructor
-~Whisky()	Destructor

Class	
Class name:	Potatoes
Base class:	
Derived class:	
Function of Class:	Resources for FoodSupplyCarriage
Variables(incl.type):	Description
Operations (incl. Return type)	Description
-Potatoes()	Constructor
-~Potatoes()	Destructor

Class	
Class name:	furniture
Base class:	
Derived class:	
Function of Class:	Resources for HospitalCarriage
Variables(incl.type):	Description
Operations (incl. Return type)	Description
-furniture()	
-~furniture()	

Relationships

Relationship	
From class:	SettlerCarriage
Multiplicity	
Justification	
To Class:	SettlerCarriageSmall
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	SettlerCarriage inherits methods to infects and destroys SettlerCarriageSmall and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	SettlerCarriage
Multiplicity	
Justification	
To Class:	SettlerCarriageMedium
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	SettlerCarriage inherits methods to infect and destroy SettlerCarriageMedium and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	SettlerCarriage
Multiplicity	
Justification	
To Class:	SettlerCarriageLarge
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	SettlerCarriage inherits methods to infect and destroy SettlerCarriageLarge and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	SheriffCarriage
Multiplicity	
Justification	
To Class:	SheriffCarriageSmall
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	SheriffCarriage inherits methods to infect SheriffCarriageSmall and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	SheriffCarriage
Multiplicity	
Justification	
To Class:	SheriffCarriageMedium
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	SheriffCarriage inherits methods to infect SheriffCarriageMedium and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	SheriffCarriage
Multiplicity	
Justification	
To Class:	SheriffCarriageLarge
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	SheriffCarriage inherits methods to infects SheriffCarriageLarge and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	ShopCarriage
Multiplicity	
Justification	
To Class:	ShopCarriageSmall
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	ShopCarriage inherits methods to infect ShopCarriageSmall and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	ShopCarriage
Multiplicity	
Justification	
To Class:	ShopCarriageMedium
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	ShopCarriage inherits methods to infect ShopCarriageMedium and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	ShopCarriage
Multiplicity	
Justification	
To Class:	ShopCarriageLarge
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	ShopCarriage inherits methods to infect ShopCarriageLarge and all the variables from Carriage that needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	Carriage
Multiplicity	
Justification	
To Class:	SettlerCarriage
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	Inherits all methods from the abstract class carriage. and pass them to SettlerCarriageSmall, SettlerCarriageMedium and SettlerCarriageLarge.

Relationship	
From class:	Carriage
Multiplicity	
Justification	
To Class:	FoodSupplyCarriage
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	Carriage has all the variables that FoodSupplyCarriage needs to save its particular characteristics and be collected straightforward by a list of carriages during runtime.

Relationship	
From class:	Carriage
Multiplicity	
Justification	
To Class:	ShopCarriages
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	Inherits all varibles from the abstract class carriage and pass them to ShopCarriageSmall, ShopCarriageMedium and ShopCarriageLarge.

Relationship	
From class:	Carriage
Multiplicity	
Justification	
To Class:	EnergySourceCarriage
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	Carriage has all variables that EnergySourceCarriage needs to save its particular characteristics using its methods and be collected straightforward by a list of carriages during runtime and also inherits a method to infect.

Relationship	
From class:	Carriage
Multiplicity	
Justification	
To Class:	HospitalCarriage
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	Carriage has all variables that HospitalCarriage needs to save its particular characteristics using its methods and be collected straightforward by a list of carriages during runtime and also inherits a method to infect.

Relationship	
From class:	Carriage
Multiplicity	
Justification	
To Class:	SheriffCarriage
Multiplicity	
Justification	
Type:	Inheritance
Justification of Type	Inherits all variables from the abstract class carriage and pass them to SheriffCarriageSmall, SheriffCarriageMedium and SheriffCarriageLarge.

Relationship	
From class:	Trek:Clist
Multiplicity	1
Justification	There is one trek for each simulation.
To Class:	Carriage
Multiplicity	1
Justification	Only one vector is needed to collects all the Carriage objects in the Trek.
Type:	Aggregation.
Justification of Type	Collects all the information for every object Carriage to be managed by trek afterwards.

Relationship	
From class:	Trek:SClist
Multiplicity	1
Justification	There is only one instance of the class Trek.
To Class:	Carriage
Multiplicity	1
Justification	Only one vector is needed to collects all the SettlerCarriage objects in the Trek
Type:	Aggregation
Justification of Type	Collects all the information for every object Carriage to be managed by trek afterwards.

Relationship	
From class:	Trek:ptr
Multiplicity	1
Justification	There is only one instance of the class Trek
To Class:	Carriage
Multiplicity	1..*
Justification	There is one pointer ptr for every object carriage and there is at least one of them in the carriage.
Type:	Aggregation
Justification of Type	This pointer allows to manage information from SClist and Clist, that contains instances of Carriage objects.

Relationship	
From class:	ShopCarriage
Multiplicity	1
Justification	A ShopCarriage can contains FoodSupplyCarriage.
To Class:	FoodSupplyCarriage
Multiplicity	1…4
Justification	ShopCarriage can contains until 4 FoodSupplyCarriage
Type:	Composition
Justification of Type	Each type of ShopCarriage use food supply carriages. With this composition, a FoodSupplyCarriages object is added to ShopCarriage.

Relationship	
From class:	Water
Multiplicity	0…*
Justification	Resource for FoodSupplyCarriage, user can modified the quantity and add more resources.
To Class:	FoodSupplyCarriage
Multiplicity	1
Justification	Each FoodSupplyCarriage can contain different types of resources.
Type:	Aggregation
Justification of Type	The user could add resources as much as he/she wants.

Relationship	
From class:	Potatoes
Multiplicity	0…*
Justification	Resource for FoodSupplyCarriage, user can modified the quantity and adds more resources.
To Class:	FoodSupplyCarriage
Multiplicity	1
Justification	Each FoodSupplyCarriage can contain different types of resources.
Type:	Aggregation
Justification of Type	The user could add resources as much as he/she wants.

Relationship	
From class:	Whisky
Multiplicity	0…*
Justification	Resource for FoodSupplyCarriage, user can modified the quantity and adds more resources.
To Class:	FoodSupplyCarriage
Multiplicity	1
Justification	Each FoodSupplyCarriage can contain different types of resources.
Type:	Aggregation
Justification of Type	The user could add resources as much as he/she wants.

Relationship	
From class:	furniture
Multiplicity	1
Justification	The amount of furniture is set previously.
To Class:	HospitalCarriage
Multiplicity	1
Justification	Every HospitalCarriage has capacity for furniture.
Type:	Composition
Justification of Type	The amount of furniture should not be modified.