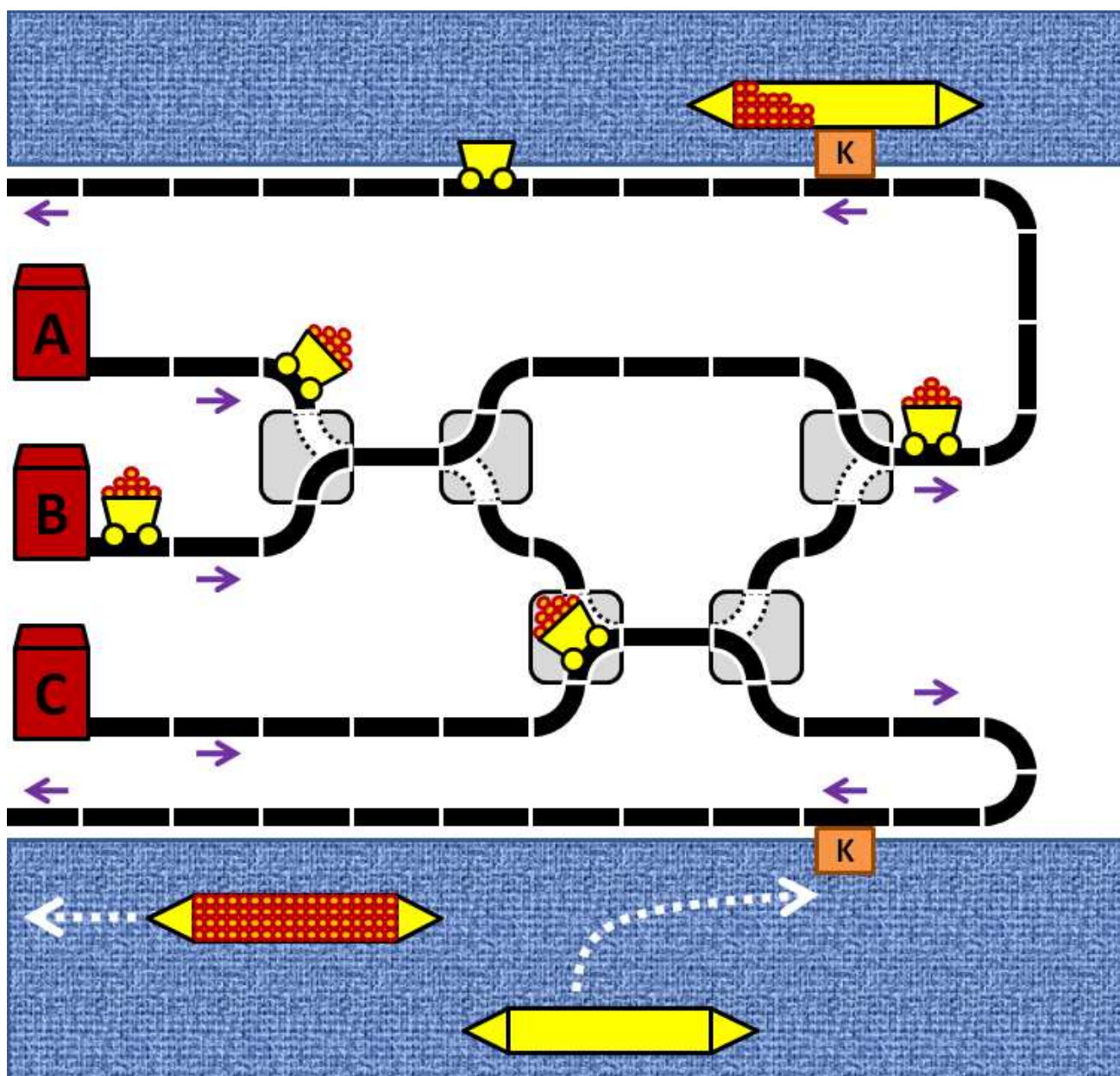




GOUDKOOFTS

Hieronder zie je een scherm van een computerspel.



De bediening (knoppen/ menu) en het scorebord staan niet afgebeeld, die moet je er zelf bij bedenken.



Random komen er volle karretjes uit loods A,B of C.

Ieder tijdsinterval rijden alle karretjes 1 stap als ze dat kunnen, dat wil zeggen, ze gaan van het huidige baanvak naar het volgende baanvak.

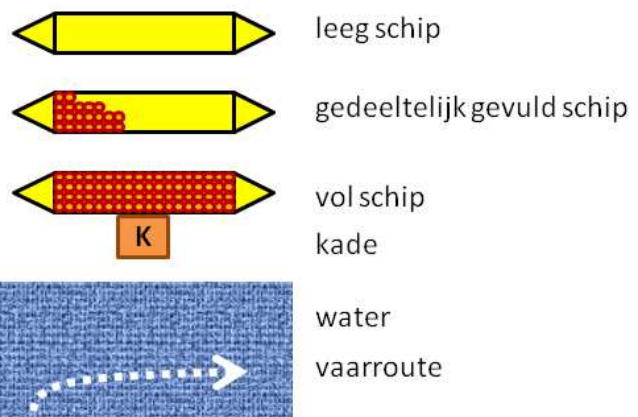
Iedere volle kar bevat evenveel lading.

Als een kar langs een kade is gereden verdwijnt hij uiteindelijk uit beeld en houdt op te bestaan. (Het is immers een virtuele wereld)

Het maximaal aantal karren in het spel is ongelimiteerd.

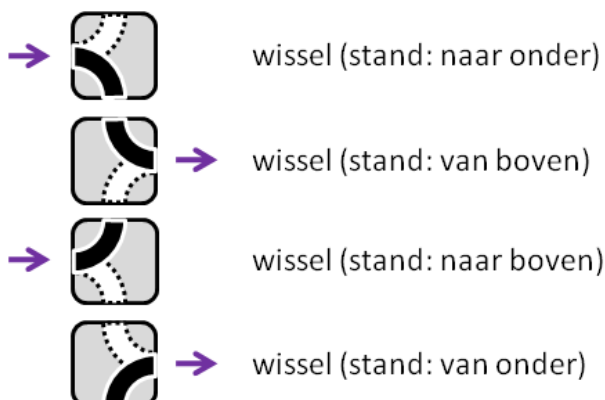
Een schip kan vol, leeg of gedeeltelijk gevuld zijn. Een schip kan onderweg zijn in het water van de rivier of aangemeerd aan de kade. Een schip heeft een bepaalde bestemming dat kan de speler zien aan de witte stippellijnpijl die de te varen route van het schip markeert. De speler heeft hier geen invloed op.

De schepen varen leeg naar de kade toe en vol er weer vandaan. Dat is een animatie die geen invloed heeft op het spel, behalve dat karretjes hun lading pas kunnen lossen als een schip aan de kade ligt dat niet vol is.



Ieder vol schip bevat evenveel lading. Er passen precies 8 kar-ladingen in 1 schip.

Als een schip vol is krijgt de speler 10 punten en daarna vertrekt het schip tot het buiten beeld is. Daar heeft de speler geen invloed op het is een animatie die een tijdje duurt. Er kunnen meerdere schepen in het spel zijn (geen maximum).



Hiernaast zie je 4 wissels.
Een wissel telt ook als 1 baanvak.

De speler kan de wissels bedienen, dat wil zeggen naar boven of naar beneden omzetten.

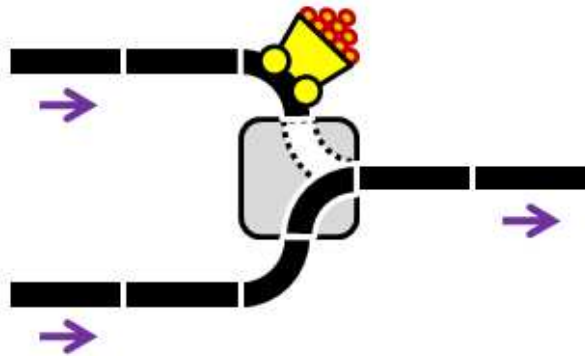
Een kar kan zich op een wissel bevinden.

Eenmaal erop kan de kar er na het volgende tijdsinterval af als dat baanvak vrij is.

Erop rijden kan alleen als de wissel de juiste kant op staat.



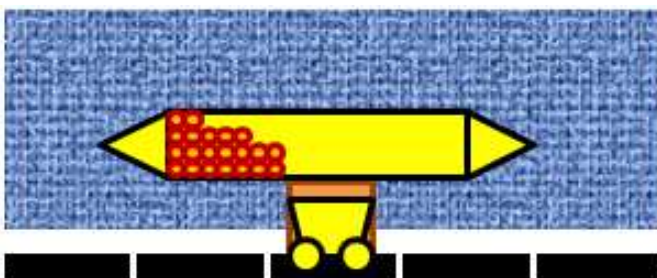
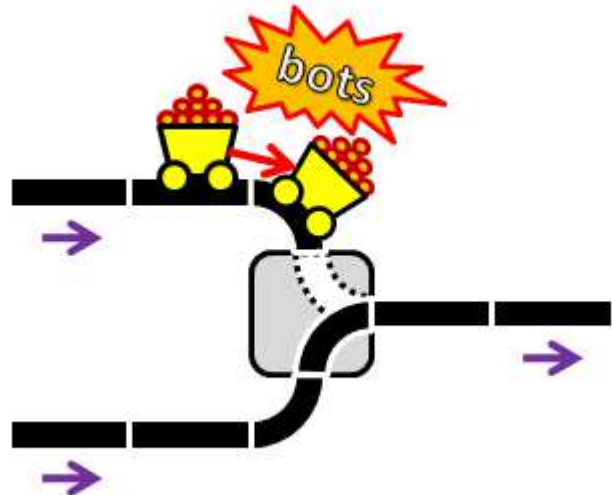
Een wissel met een karretje erop kan niet worden omgezet.



Als een wissel niet goed staat blijft een karretje in zijn baanvak staan. Gedurende het tijdsinterval mag de speler een wissel zo vaak omzetten als hij wil, echter bij het einde van de periode rijdt het karretje pas en op dat moment moet de wissel zodanig staan dat het karretje erdoor kan, anders blijft het nog een tijdsinterval staan.

Twee karren botsen omdat de voorste niet kan rijden omdat de wissel verkeerd staat, de achterste rijdt wel en komt nu in het baanvak van de voorste.

Het spel stopt meteen als twee karretjes botsen.



Een kar, die de kade met schip heeft bereikt, is net gelost.

De speler moet om punten te scoren de volle karretjes langs een van de twee kades leiden (door de wissels op tijd goed te zetten) en wel die kade kiezen waar een schip ligt dat nog niet vol is. Als een vol karretje bij een kade komt, lost het automatisch zijn lading in het schip, tenminste als er een schip is en nog niet vol is, anders rijdt het door zonder te lossen, dat is dan jammer want dat levert geen punten op. Voor dat lossen hoeft de speler niets te doen.

De speler krijgt 1 punt voor iedere kar die in een schip wordt gelost en nog eens 10 punten voor een vol schip.

In het begin is het spel makkelijk want dan zijn er weinig karretjes. Als de speler meer punten verzamelt neemt de moeilijkheidsgraad toe. Dan komen er steeds meer karretjes en zal het tijdsinterval ook korter worden, het spel gaat dan sneller.

Zoals je ziet in het speelscherm zijn er 5 wissels. Niet zichtbaar zijn de volgende elementen: De speler heeft 5 knoppen. Door op een knop te drukken wordt een wissel omgezet. Er is een countdown die aftelt. Als die op nul staat is het tijdsinterval voorbij. De bediening gaat even op slot. Dan rijden alle karretjes naar het volgende baanvak, lossen lading, wachten of botsen al naargelang de situatie. Als er geen botsing is gaat de bediening weer open, de countdown wordt opnieuw gestart.

opdracht

opdracht

Maak een ontwerp voor het programma volgens het MVC principe en meerlagenarchitectuur (multi tier).

- **Klassendiagram**
 - overzicht (abstract)
 - fragmenten behorenden bij de sequence diagrams
- Kies zelf **3 moeilijke plekken**, die het waard zijn om een **Sequence Diagram** van te maken, om daarmee de werking uit te leggen.

Maak 1 solution in Visual Studio waarin je de C#-code en de modellen bij elkaar houdt.

Maak een **Object-geOrienteerd domein model (DKD)** waarbij de objecten uit de "echte" wereld zo nauwkeurig mogelijk worden overgenomen als klassen in je code en waarbij het gedrag van een object in je code ook zo veel mogelijk lijkt op dat van het object uit de "echte" wereld. In dit DKD mogen geen zaken staan die iets te maken hebben met de GUI (presentation) of met de werking van je applicatie (process).

In dit DKD moet je gebruik maken van een linked-list van velden, van overerving en polymorfisme.

Bouw het programma in C# en maak een command-line interface (niets uitvoerigs, gewoon Console.WriteLine()) om aan te tonen dat jouw ontwerp ook echt werkt. Laat de modellen en de werkende code zien aan de practicumdocent en laat het aftekenen.