Trein Simulatie

Het idee is om een command line Trein simulator te gaan ontwikkelen. Gedurende de simulatie krijgt de speler constant 4 opties, wat deze opties zijn verschild per scenario. Wanneer de game start krijgt de speler een persoon met treinticket, het doel is om deze persoon succesvol naar zijn bestemming te krijgen.

Er zijn verschillende type reizigers in de trein met variërende emoties en reacties op situaties. Deze reizigers worden met behulp van een **template method pattern** gemaakt.

Wanneer de reiziger en ticket zijn toegewezen krijgt de spelen 4 opties in verschillende situaties. Bijvoorbeeld welke coupé de reiziger instapt, hoe de reiziger reageert op een luidruchtig persoon of hoe de reiziger zich gedraagt. Deze opties hebben allemaal effect op de emoties van de reiziger, en kunnen mogelijk tot consequenties leiden zoals de trein uit gezet worden of overlijden tijdens de reis. Ook kan de trein via de **observer pattern** omroepen dat deze vertraging heeft, ook dit heeft invloed op de emoties en acties van de reiziger.

Er zijn 3 verschillende states waarin de game zich kan bevinden namelijk de beginstate, de gamestate en de endstate. In de applicatie kan de speler altijd 4 knoppen indrukken namelijk 1, 2,3 en 4. Wat deze knoppen doen is afhankelijk van de state waar de game zich in bevind. Dit is de **state pattern**. Er wordt voor elke ‘ronde’ aan keuzes een nieuwe gamestate met andere werkingen voor de knoppen aangemaakt vanuit een XML-reader.

De trein wordt aangemaakt met verschillende coupés waar de reiziger gebruik van kan maken. Alle coupés worden gemaakt met behulp van en **factory method**. Er zijn 5 type coupés namelijk de tweedeklascoupé, eersteklascoupé, stilte coupé, muziek coupé en bar coupé.

De reiziger heeft verschillende emoties zoals blijdschap, verdriet en boosheid. Ook heeft de reiziger modifiers op deze emoties, welke startwaardes en modifiers deze emoties hebben is afhankelijk van het type reiziger. Zo zal een “Karen” reiziger sneller boos worden dan een “zakenman”. Elke situatie staat beschreven in een Xml-file met daarin een situatiebeschrijving voor de speler, 4 opties waaruit gekozen kan worden met behulp van de buttons in een state en de consequenties van elke optie eventueel met speciale uitkomsten in een specifieke coupé. Zo zal een optie een reiziger boos kunnen maken, blij kunnen maken of verdrietig kunnen maken. Een reiziger zal in een stilte coupé bozer worden om schreeuwende jongeren, en een “Karen” zal bozer worden dan een jongere.

Per reiziger type staan er een aantal limieten op elke emotie, wanneer deze wordt overschreden van de game verloren zijn. Zo zal een zakenman bij te veel woede een hartaanval krijgen van zijn hoge bloeddruk, terwijl een Karen een woedeaanval zal hebben en de trein wordt uitgezet.

**State pattern** 🡪 states in de game worden constant aangemaakt aan de hand van een XML lijst met scenario’s, opties en consequenties. De speler heeft altijd 4 opties om uit te kiezen maar wat deze opties doen is afhankelijk van de huidige state.

**Template method pattern** 🡪 Reizigers maken met behulp van een abstract reiziger. Hier verschilt hoe de reizigers zich voelen en hoe ze reageren op situaties.

**Observer**  🡪 De observer luistert naar de trein. Wanneer deze trein een bericht geeft dat deze te laat komt ontvangen reizigers deze en reageren ze hierop.

**Factory method** 🡪 Maakt alle coupés voor de trein, er zijn 5 type coupés. Namelijk de Stilte coupé, eersteklascoupé, tweedeklascoupé, bar coupé en de muziek coupé.

