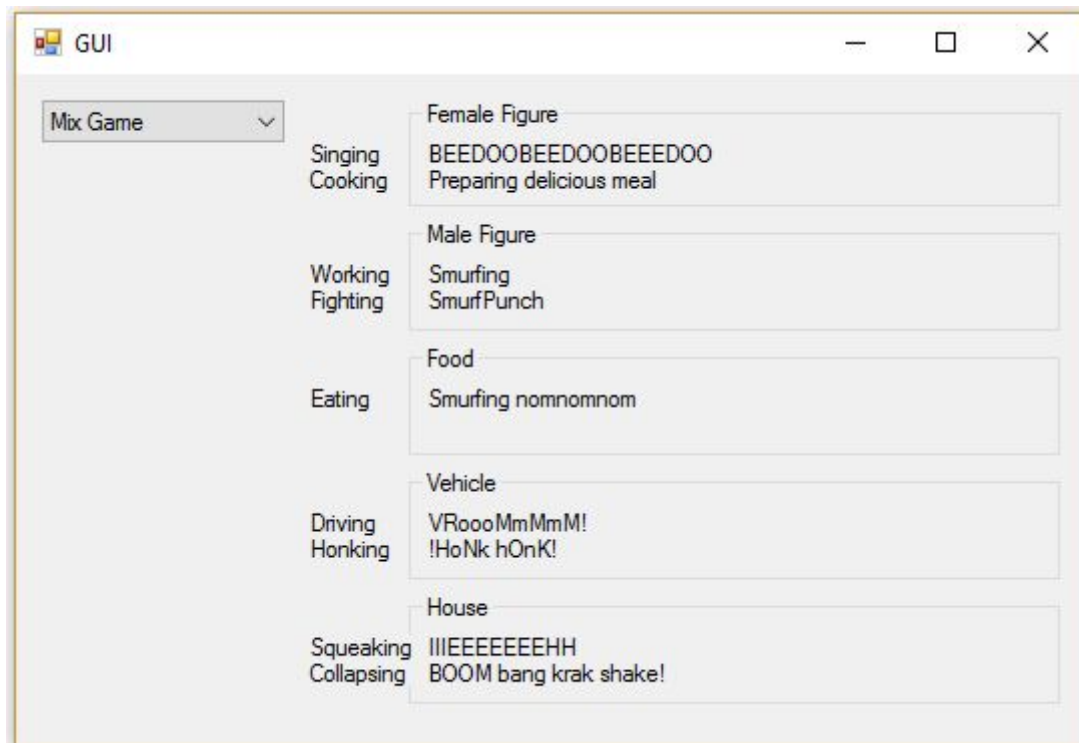


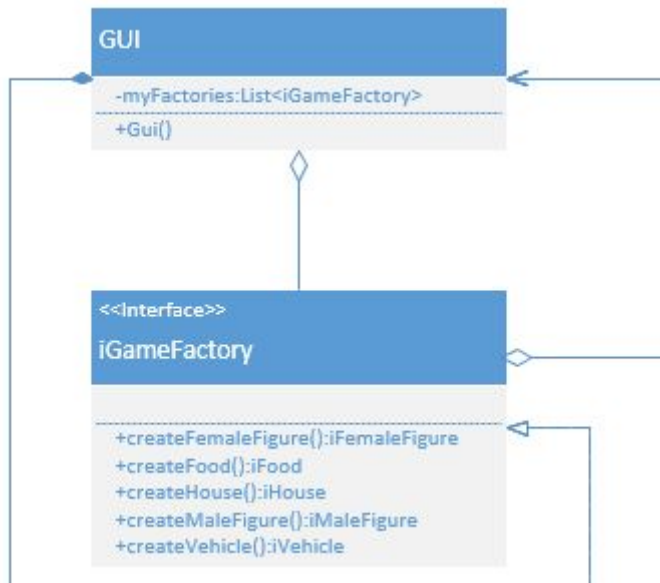
Factory Pattern Assignment

Tim Ackermans & Dominic Voets

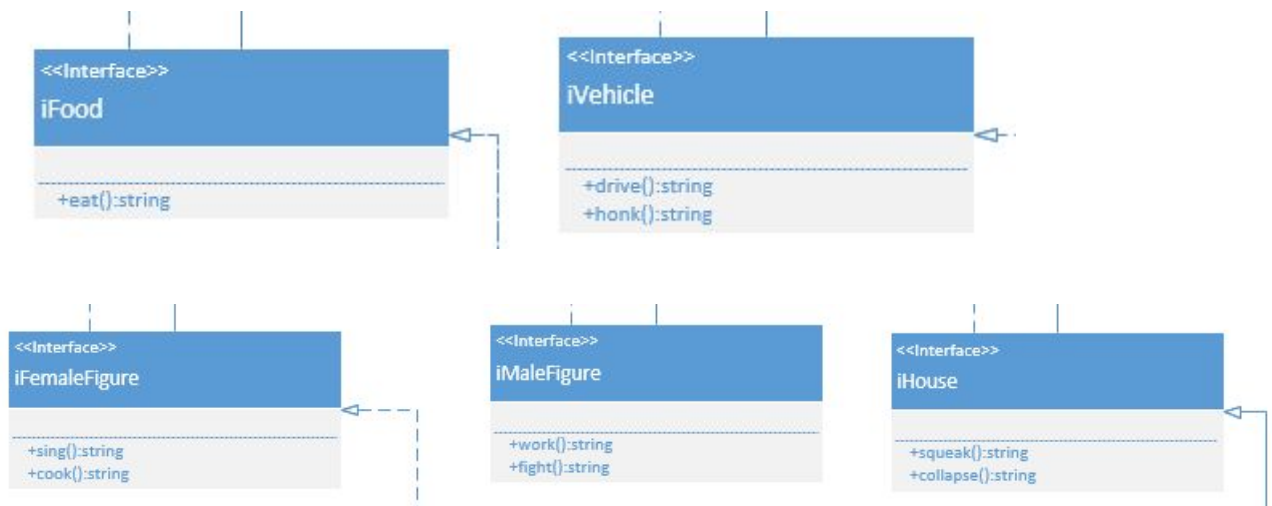


Factory pattern uitleg

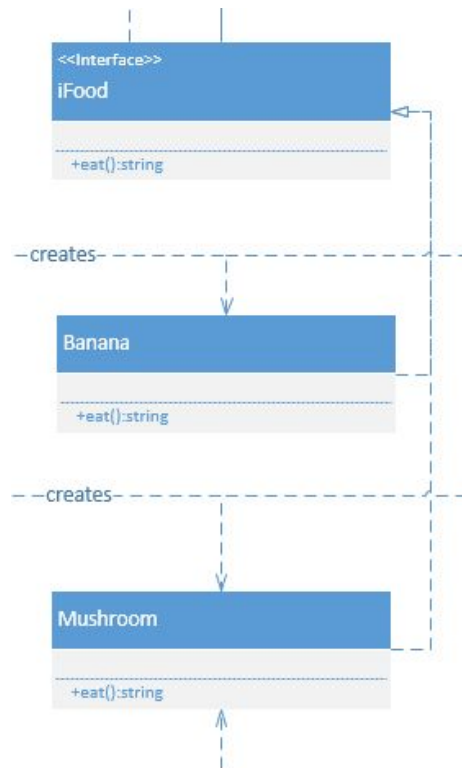
Een factory pattern wordt gebruikt als men met dezelfde bouwstenen meerdere configuraties wil maken. In ons voorbeeld zijn er minion en smurfen bouwstenen. In ons geval heeft de GUI een lijst met iGameFactory:



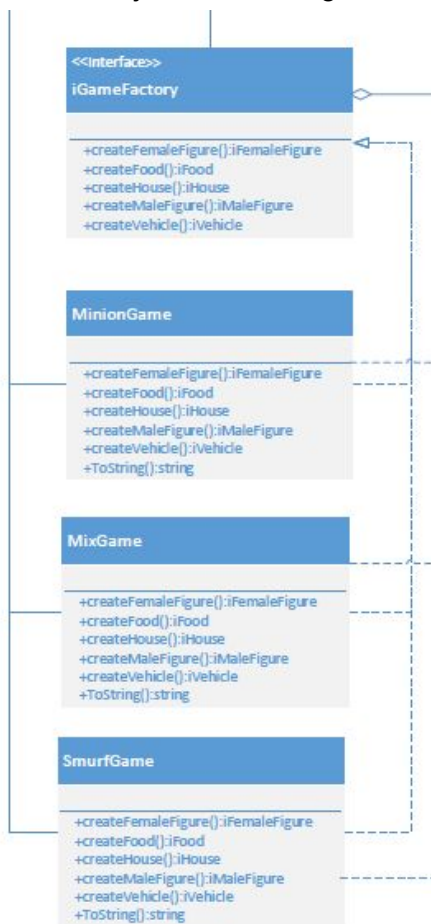
Elke iGameFactory maakt gebruik van onderstaande interfaces:



Elk interface een zijn eigen implementaties, de zogenaamde bouwstenen:



Daarnaast zijn er concrete game factories:



De concrete game factory kiest vervolgens welke bouwstenen deze wil gebruiken per game:

MinionGame:

- Banaan
- SuperMegaDeath
Car
- GruHouse
- Minion
- Minioness

SmurfGame:

- Mushroom
- ShanksPony
(benenwagen)
- MushroomHouse
- BigSmurf
- Smurfin

MixGame:

- Mushroom
 - SuperMegaDeath
Car
 - GruHouse
 - BigSmurf
- Minioness

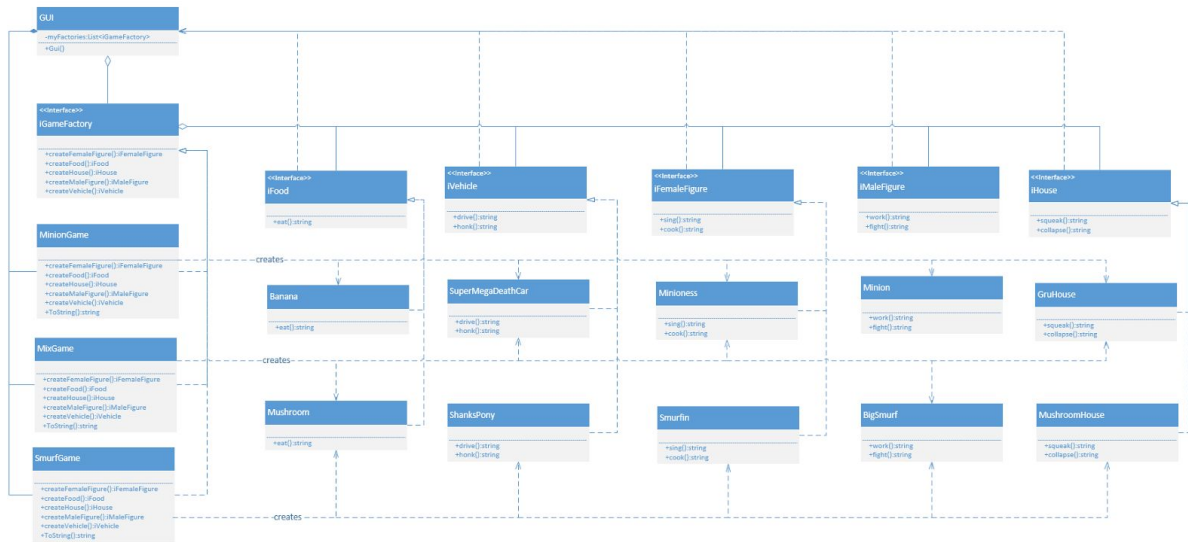
De client (GUI) kan via de iGameFactory de methodes van bijvoorbeeld iVehicle aanroepen. Wat er hierbij gebeurt is afhankelijk van welke concrete game factory er is gekozen.

Ontwerp

Uit de opdracht hebben we een aantal eisen kunnen opmaken:

- 1 Client class
- Client weet niks over de implementatie

Om het pattern te simuleren hebben we een GUI die de implementaties van de verschillende objecten laat zien aan de hand van labels. Deze text voor deze labels is afhankelijk van welke factory er is gekozen. Elke factory heeft zijn eigen keuzes per interface.



Grote en zoombare versie in de meegeleverde pdf.

Voor en nadelen van het Observer pattern

Reusability

- +De interfaces kunnen voor elk soort game worden hergebruikt waardoor het maken van nieuwe factories erg gemakkelijk gaat.
- +Objecten van verschillende games kunnen in andere games worden hergebruikt.

Maintainability

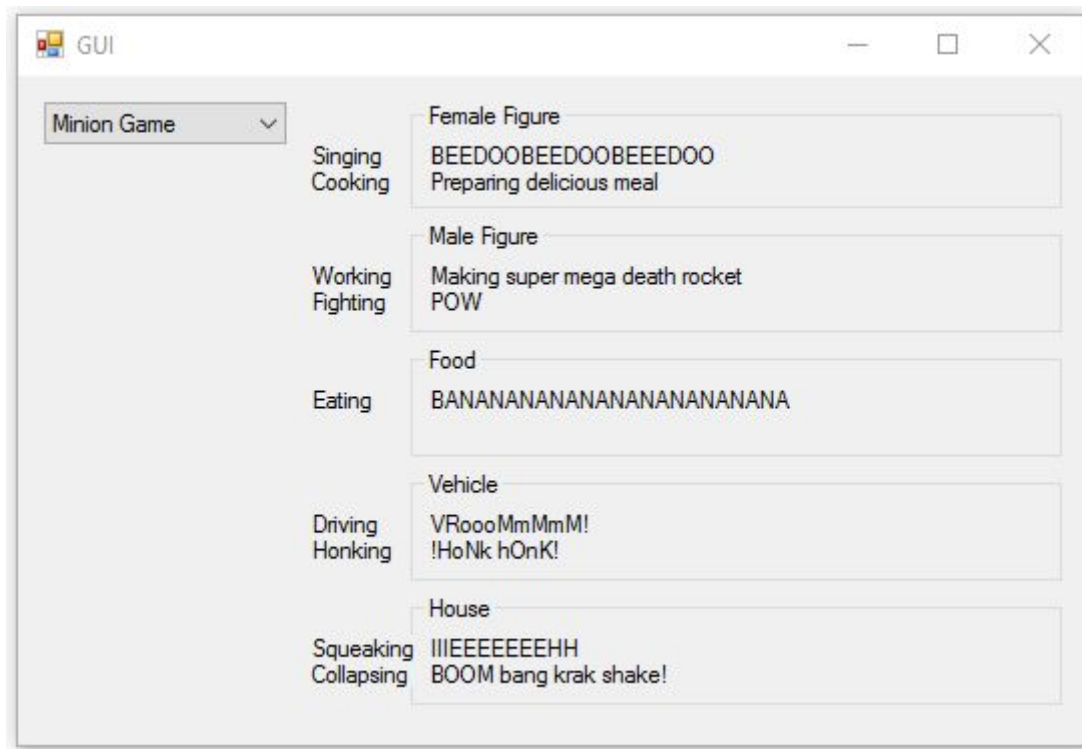
- +Als een van de objecten een ander gedrag moet gaan vertonen hoeft het bovenliggende spel niet aangepast te worden. (vb. Als de smurfin anders moet zingen hoeft alleen het zing stukje aangepast te worden)
- -Om veranderingen te maken moet dit in elke game apart worden gedaan
- -Het is moeilijker om code te lezen omdat alle code abstract is

Extensibility

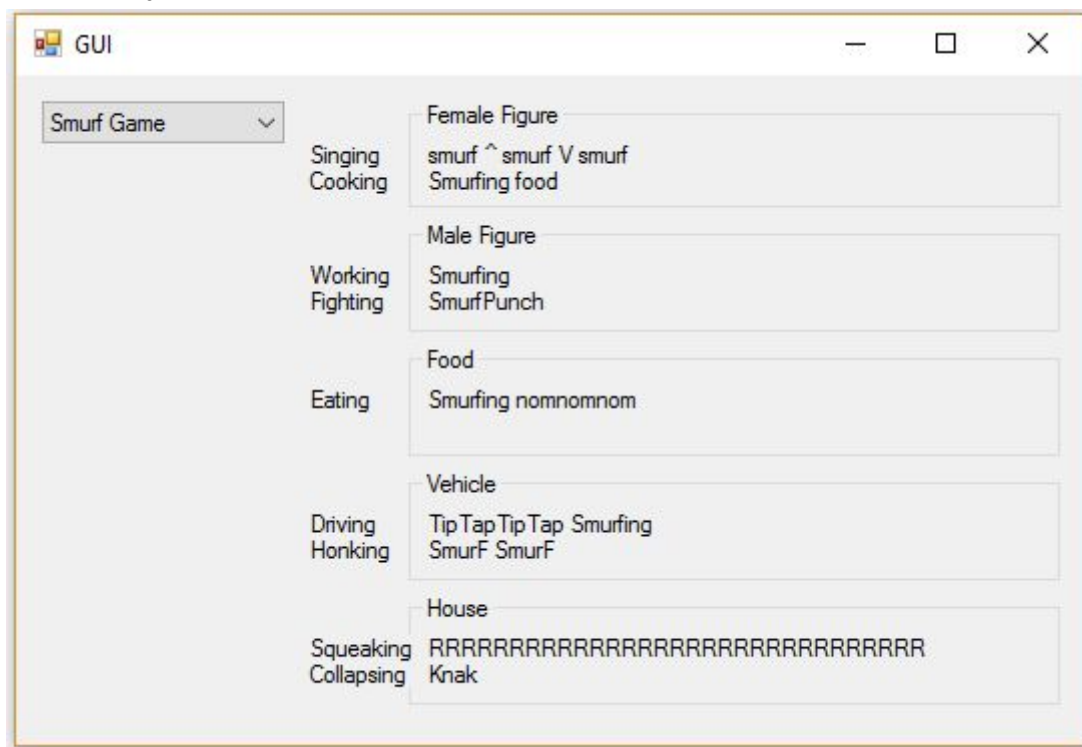
- +Als er een nieuwe factory toegevoegd moet worden dan is dit erg simpel.
- +Toevoegen van nieuwe objecten in bestaande categorie is heel makkelijk

Demo

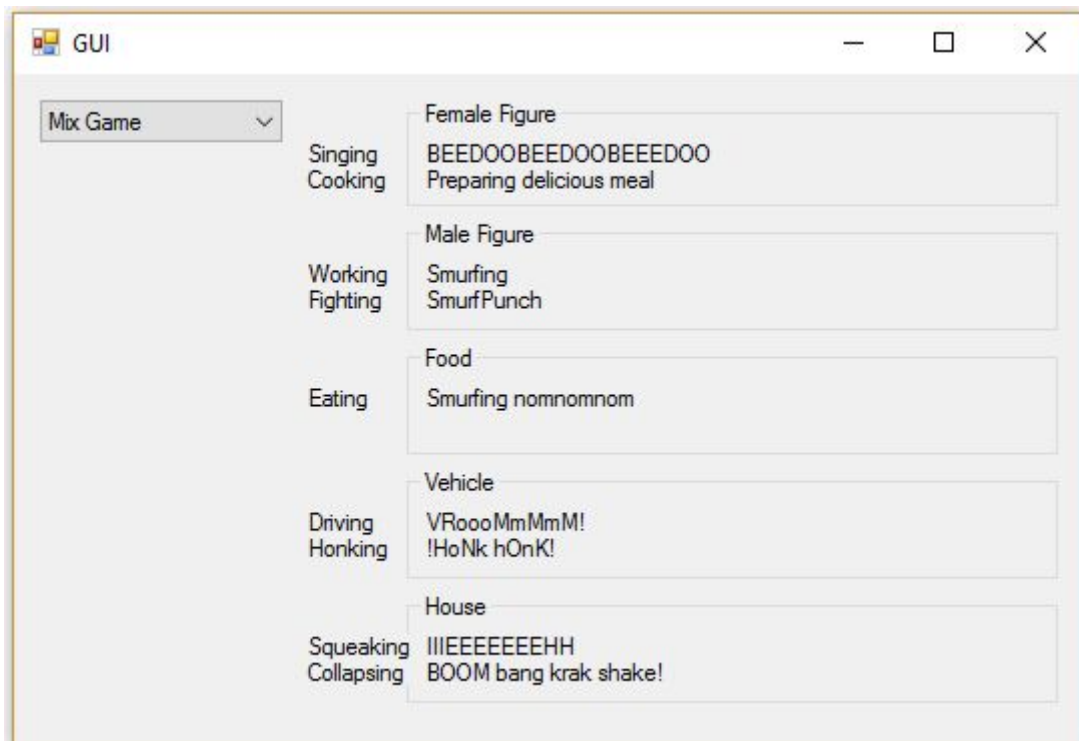
Resultaat bij Minion Game



Resultaat bij Smurf Game



Resultaat bij Mix Game



The screenshot shows a window titled "GUI" with a "Mix Game" dropdown menu. Below the menu, there is a list of actions and their corresponding sounds and descriptions.

Action	Sound	Description
Singing	BEEDOOBEEDOOBEEEDOO	Female Figure
Cooking	Preparing delicious meal	Female Figure
Working	Smurfing	Male Figure
Fighting	SmurfPunch	Male Figure
Eating	Smurfing nomnomnom	Food
Driving	VRoooMmMmM!	Vehicle
Honking	!HoNk hOnK!	Vehicle
Squeaking	IIIEEEEEEEHH	House
Collapsing	BOOM bang krak shake!	House