

## Readme Typescript Game

LIVE-URL: <https://stud.hosted.hr.nl/0881520/websites/AquariumGame/dist/>

### Installatie

Open de index.html die in de dist map te vinden is. Mocht je zelf de code willen aanpassen, kun je dit aanpassen in de files die te vinden zijn in de dev map. Na het aanpassen van de code, moet de code gecompileerd worden. Zorg dat je een compiler hebt die Typescript om kan zetten naar javascript.

### Singleton Pattern

De Singleton Pattern wordt gebruikt in de Game Class. Op deze manier zorg ik dat er altijd maar 1 instantie kan zijn van de game. Deze kun je overal vandaan benaderen door de Game.getInstance() functie aan te roepen.

### Strategy Pattern

De Strategy Pattern wordt gebruikt bij de gedrag van de Fish class. In de draw function wordt de behavior.draw aangeroepen. Afhankelijk van de class die in behaviour zit, zal deze verschillend gedrag tonen. Als ik behavior = new swimming(this) gebruik, zal het visje rondzwemmen. Maar als ik behavior = new swimToBait(this) gebruik, zal het visje naar het voedsel zwemmen.

### static utility method

Deze is momenteel nog niet aanwezig in het project. Ik wil een aparte Utility class maken waar een collision detection function in komt. Dit zal een static method worden die overal vandaan te benaderen is.

### Interface

Bij Behavior wordt een Interface gebruikt. Alle classes die het gedrag aangeven implementeren de Behavior interface.

- Encapsulation, Composition, Inheritance

### Encapsulation

Er wordt overal gebruik gemaakt van encapsulation. Bij de vissen is dit goed te zien. Hier komen parameters van zowel public, private als protected in voor.

### Composition

Composition komt ook voor bij de vis klas. Zonder de vis zal er geen muntje ontstaan. De mus wordt gecreëerd in de vis om de x-aantal seconden.

### Inheritance

Inheritance is ook bij de vis te vinden. De GreenFish, BlueFish en LightFish erven allemaal vanuit de Fish class. Hierdoor is het mogelijk om vissen met soortgelijk gedrag te creëren en toch met een ander uiterlijk.