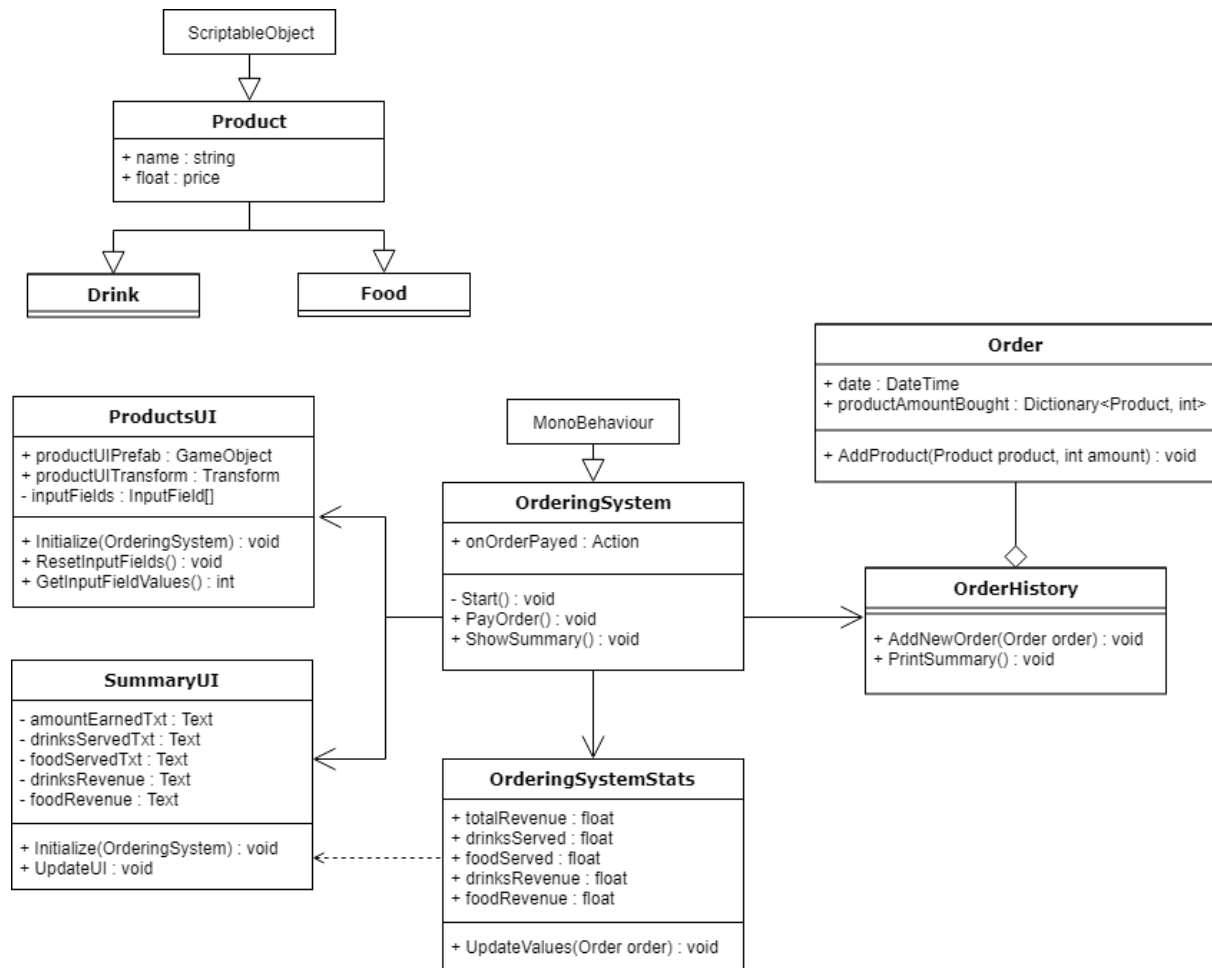


Ontwerpkeuzes Bestelsysteem van Jelle Dekkers

















Het belangrijkste script is `OrderingSystem.cs`, dit is het script dat de bestellingen afhandelt en is de enige die inherit van `MonoBehaviour`. Om alle logica toch zo gescheiden mogelijk te houden heeft `OrderingSystem` een field voor `ProductsUI` en `SummaryUI` die beiden hun eigen UI bijhouden. Op deze manier kan ik toch in de inspector de referenties vullen en blijft de logica van elkaar gescheiden.







`Drink` en `Food` inheriten van `Product`, die op zichzelf weer inherit van `ScriptableObject`. Op deze manier kan ik heel makkelijk nieuwe producten in de editor toevoegen. Aanvankelijk wilde ik alleen `scriptableObjects` gebruiken als placeholder, zodat ik het bestelsysteem kon opzetten en het daarna maken met een database (misschien een beetje overkill maar naar mijn mening wel de beste manier voor een bestelsysteem).

Het database gedeelte staat er (de php code heb ik in het project bijgevoegd) en kan in Unity worden opgevraagd. Deze is verder nog niet volledig geïmplementeerd omdat ik dan geen inheritance meer zou gebruiken, iets wat wel in de mail werd gevraagd. Ik heb verder ook geen interfaces gebruikt omdat die simpelweg niet nodig waren. Ik ben heel bekend met de werking van inheritance en interfaces en dit wilde ik ook aantonen.

+ Opties

				id	name	productTypeID
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	1	Coffee	1
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	2	Espresso	1
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	3	Cappucino	1
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	4	Latte	1
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	5	Tea	1
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	6	Apple Pie	2
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	7	Brownie	2
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	8	Muffin	2

+ Opties

				id	name
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	1	Drink
<input type="checkbox"/>	 Wijzigen	 Kopiëren	 Verwijderen	2	Food

Elke keer als er een order is gemaakt wordt er een nieuwe Order class gecreëerd die de datum, product en hoeveelheid bewaart. Hierna worden de OrderHistory en de OrderingSystemStats geupdate. OrderHistory houdt via een List<Order> alle betaalde orders bij en print deze op volgorde uit. OrderingSystemStats houdt de benodigde statistieken bij die SummaryUI gebruikt.