1. 需求描述：設計一款遊戲，需要時常和使用者回應才能讓遊戲不會單調乏味，因此我在使用者輸入指令後都會有相對應的回應，並用簡短但明確的訊息來使使用者不須花太多力氣就能的到完整的資訊。
2. 程式流程：

1. Object/Class 敘述：

object 的設計是基於能讓程式更將方便的維護或是調整，並且能讓程式的易讀性提升，也能防止編寫者輸入時輸入錯誤的訊息，同時能讓編寫時的錯誤率降低。

object 具備的特性和行為：封裝是其中一個行為，讓編寫者只需使用指令，而不需要去明白其中所進行的原理，也能保護資料不會輕易被修改。

在程式中其行為是提供所含的資料給程式做比對。

object 間的互動關係：main會呼叫fish1、2和shrimp1、2的資料，用以進行程式。

1. 使用說明：程式開始時，會提供指令供使用者參考，輸入相對應的指令時會出現對應的變化，如add F1 1 為加入一條魚。

在程式中輸入--help會有完整的說明。程式中也會有許多的特殊情境如寒流來襲或是蝦子脫殼或是餵食飼料等，為使用者提供更加有趣的體驗。

1. 其他：注意蝦子會脫殼，且當重量為 3, 6, 9 時都要脫殼一次，當他到達脫殼臨界值時，就會有三天就 算吃飼料也不會成長。而當蘭流來襲時如果魚沒辦法忍受當下的溫度則會暴斃，務必小心使用。指令輸入時為小寫，魚的代碼為大寫如F1 F2 S1 S2。