

Inlämningsuppgift 2

De största problemen som har uppkommit under utvecklingen av programmet är strukturella problem. Eftersom det inte finns några klasser innebär det att alla variabler är publika och instansernas i .c filen. Detta får till följd att alla filer som har blivit inkluderade i filen och de filer som är inkluderade i de inkluderade filerna har tillgång till alla variabler och funktioner vilket skapar en stor röra av funktioner och variabler. Detta ställer höga krav på utvecklaren att planera strukturera för att uppnå inkapsling i så hög grad det är möjligt. Jag är inte nöjd med min egen struktur gällande denna delen då jag anser att "spelen skulle endast blivit anropade av GameManagern och de skulle i sin tur inte innehaft någon vetskap om GameManagen men för att kunna återgå till huvud menyn tvingades jag anropa manager() i GameManagern. Dessutom hade jag en InputValidation fil vilken inkluderades i GameManager och i samtliga spel filer i och med detta gjorde jag det möjligt för flertalet av filerna att få tillgång till varandra. För att öka på inkapslingen hade en lösning varit att varje fil innehöll de funktioner som InputValidation filen innehöll vilket hade inneburit duplicering av kod, med andra ord jag har ingen lösning på problemet. Det hade uppskattats om jag kunde få feedback på min struktur. Det borde ingå något avsnitt i kursen hur man strukturerar upp sin kod i C eftersom det är ett stort problem, är tanken att vi ska skriva OOP C eller gammal dags C? Vidare har jag insett att struktureringen i GameManagern där jag använder enums för att bestämma tillstånden var effektivt och koden enkel att läsa, GameManager filen är definitivt en fil jag kommer återanvända och utveckla vidare. Input hanteringen i C är svår då det upplevs som om det finns en uppsjö med olika scenarion som ska kontrolleras och inga hjälp funktioner, i programmet använde jag mig av _flushall vilken tömmer streamen och genom detta undviker jag att föregående inmatade värde genererar en oändlig loop. Metoden att använda _flushall för detta syfte känns som ett felaktigt tillvägagångssätt men då jag inte vet några andra alternativ för att lösa uppgiften och säkerställa stabiliteten i programmet ses detta som en godtagbar lösning.