LINKÖPINGS UNIVERSITET

TDP005 Projekt: Objektorienterat system

Kodgranskningsprotokoll

Författare

Albin Dahlén, albda746@student.liu.se
Filip Ingvarsson, filin764@student.liu.se

INSTITUTIONEN FÖR DATAVETENSKAP

Höstterminen 2022
Version 1.0

2022-12-08

1 Revisionshistorik

Ver.	Revisionsbeskrivning	Datum
1.0	Första utkast	22-12-08

2 Möte

Före mötet såg vi till att beställargruppen hade tillgång till vår kod på Gitlab samt att vi hade tillgång till deras. Före själva mötet hade båda grupperna granskat varandras kod och antecknat synpunkter som sedan tog upp på mötet. Mötet ägde rum 8 December klockan 14:00.

3 Granskat Projekt

Spelet som granskades var ett plattformsspel där spelaren styrde en karaktär med ett visst antal liv och målet var att nå slutet av banan. Under speletsgång möter spelaren fiender som kan skada spelaren om de kolliderar.

3.1 Kommentering

Koden saknade nästan helt kommentarer vilket gjorde det väldigt svårt att förstå vad kodstyckena gjorde. Detta är även ett krav i projektbeskrivningen då Doxygen ska användas för att generera dokumentation.

3.2 Const och referenser

Koden saknade const och referenser i sina funktioner, detta kan göra att koden gör saker man inte tänkt och kan vara väldigt svårt och tidskrävande att lägga till i slutet av projektet. Programmet tar även mer minne då den kommer skapa kopior till alla istället för att skicka referenser.

Flera av klasserna innehåller getters som inte är markerade const vilket de borde vara.

3.3 Koduppreprepning

Beställargruppen hade några funktioner i sina klasser som var väldigt snarlika exempelvis, i Move-funktionerna för player kallar de på Left() och Right() beroende på om användaren klickar på höger eller vänster piltangent. Left och right funktionerna gör dock i princip samma sak, ändrar på samma värden men till olika saker. Ett förslag hade varit att ha det i samma funktion men ta någon parameter som avgör vad som ändras enligt Don't repeat yourself principen.

Visa funktioner finns i flera klasser men är inte virtuella, exempelvis funktionen getSprite finns i både block och player klassen och dessa klasser ärver från klassen object. Förslagsvis hade funktionen då placerats i object klassen, samma sak gäller för variabeln sprite som även den finns i båda klasserna.

3.4 Formatering

Beställargruppen använder sig av olika kodstilar, exempelvis i state machine har de blandat PascalCase och camelCase här borde bara en av dessa använts. Sedan i player-delen har de blandat snake_case samt om snake_case skrivs med små och stora bokstäver i varje ord och slutligen blandat in funktions namn som ibland har stor bokstav och ibland liten bokstav.

3.5 Variabelnamn

Player har en variabel som heter 'On_ground' samt en variabel som heter 'is_jumping'. När vi granskade deras kod innan mötet tänkte vi bara att dessa variabler var motsattsen mot varandra men visade sig att 'on_ground' hade att göra med vilket håll gravitationen var riktad. Detta missförstånd hade kunnat undvikas vid bättre variabelnamn kanske 'gravity_down' istället för 'on_ground', eller kommentering som göra att en ny läsare av koden kan förstå vad som menas.

3.6 Klasser och arv

Player klassen har ett multipelt arv av både object och sf::Sprite, vi ser ingen anledning till att göra på detta sätt och tycker att det skulle generera mer problem än vad det skulle hjälpa dem. De använder heller aldrig de funktioner de får från sprite klassen någon gång.

3.7 Positiv feedback

3.7.1 Design

De har följt sin design väldigt bra, vilket tyder på att de tänkt igenom projektet ordenligt innan de började.

3.7.2 Relevanta klasser och arv

Generellt har beställargruppen fått till bra klasstrukturer och arv.

4 Vårat Projekt

4.1 Kommentering

Beställarengruppen påpekade att majoriteten av projektet var bra kommenterat, men sedan att visa klasser och delar av kodstycken saknade förklarande kommentarer. Vi har planerat att när ett tillfredsställande resultat uppnåtts ska koden gås igenom och kommentarer ska uppdateras och förbättras eventuellt läggas till om något stycke inte är helt klart.

4.2 Formatering

Beställargruppen påpekade att vi använt oss av flera olika kodstilar, däribland snake_case och camelCase vilket gjorde koden jobbigare att läsa då den inte var konsekvent. Detta är något vi efter påpekan ska gå igenom och fixa så att det är samma standard över hela projektet.

4.3 Projektiler

Beställargruppen påpekade att våra projektiler bara kan träffa enemies just nu och att vi i vår designspecifikation har sagt att enemies också ska kunna skjuta på spelaren. Anledningen till detta är att vi inte hade kommit tillräckligt lång i spelet för att ha implementerat det ännu. Detta kommer dock implementeras inom de närmaste dagarna.

4.4 Testa Programmet

Beställargruppen påpekade att det inte fanns någon instruktion hur man körde programmet. En temporär lösning skapades för att kunna köra programmet. En faktisk lösning ska lösas snarast så att programmet inte är beroende av Clion för att köras.