Space Adventures

Grupp 37

Verifiering och valideringsdokument

V. 1.0

170418

Dokumenthistorik

Datum	Version	Beskrivning	Författare
170404	1.0	Dokumentet skapas	Fredrik Johnson
170411	1.0	Allmän text om syfte, testprocess och testfall.	Fredrik Johnson
170412	1.0	Text för dokumentgranskning tillagd. Text för white-box testing.	Carl-Magnus Klang, Fredrik Johnson
170414	1.0	Text för kodgranskning påbörjad.	Andreas Undfors
170417	1.0	Text för kodgranskning avslutad.	Andreas Undfors
170416	1.0	Text för användbarhetstestning	Daniel Medjedovic
170417	1.0	Omformulering och rättstavning, spårningsmatris, systemtestning	Oliver Liefke, Carl-Magnus Klang
170418	1.0	Testprocess och omformulering	Oliver Liefke, Carl-Magnus Klang, Fredrik Johnson
170505	1.0	Testrapporter	Carl-Magnus Klang

Innehåll

Dokumenthistorik 2
Verifiering och valideringsdokument 4
Syfte 4
Ordlista 4
Referenser 4
Testprocess 5
Granskning 5
Kodgranskning 5-6
Dokumentgranskning 6-8
Testning 8
Kravbaserad systemtestning 8-9
White box-testing: Statement Coverage 9
Användbarhetstestning: Test rapport 9
Testfall kravbaserad systemtestning 10-11
- Funktionella Krav 10-11

Spårningsmatris 12
Testfall/materiel för annan metod 13
Granskningsprotokoll 13
Testrapporter 13
Analysrapporter 13

Verifiering och valideringsdokument

Syfte

Syftet med verifierings och valideringsdokumentet är att ge en översikt av de granskningar och tester som sker kring olika delar i projektet.

Validering ger gäller huruvida projektgruppen bygger rätt system.
Verifiering avser huruvida gruppen bygger systemet på rätt sätt.

Ordlista

Testfall	Data som exekverar system för att testa en systemegenskap.
Testdata	Data specificerad för användning i testfall.
Teststeg	Lista av de steg som skall genomföras under ett test.

Referenser

Tsui, Frank. 2013.Essentials of software Engineering. 3. uppl. Jones & Bartlett Publishers, Inc.

Testprocess

Då projektet utgår ifrån ett agilt arbetssätt kommer testningen av systemet i huvudsak att ske kontinuerligt under projektgången, och inte endast under specifika tillfällen. Systemtestning utifrån kraven sker främst vid implementation av nämnda krav. Dock så kommer projektgruppen använda sig av speciella tillfällen då koden granskas av hela gruppen, där alla får en chans att ordentligt lära sig koden och vad varje individuell del gör. På så sätt bildas en förståelse för projektet som en helhet, istället för att endast de medlemmar som arbetade med en viss metod är de som förstår den. Det ser också till så att alla gruppmedlemmar godkänner arbetet som gjorts så att alla är på samma sida. Dessa större testerna kommer att ske då gruppen når en milstolpe för projektet, samt inför varje sprint.

Det som kommer att testas är de testfall som dokumentet beskriver. Allt dokumenteras direkt under testprocessen utav den ansvarande för testet, vilket huvudsakligen blir den medlem som ansvarade för kravet.

Förutom de manuella testerna så används även de inbyggda funktionerna i både “Visual Studio”, vars debugg funktion varnar när det uppkommer komplikationer i koden, samt i “GitHub” där programmet kontrollerar att inga ändringar gjorde i dokumenten eller koden krockar med andra ändringar som har skett.

Granskning

Kodgranskning

Koden kommer granskas genom att en av programmerarna går igenom koden inför de andra. Denna metod kallas “Software Walkthrough” och används i detta sammanhang för att höja kvalitén på koden samt att få alla på samma tankebana gällande koden. Per rekommendation av IEEE 1028 kommer granskningen att bestå av tre roller.

- Författare, som är personen som skrivit koden.
- Granskare, kallas även för “Walkthrough Leader” och är personen som håller i granskningen och går igenom koden. Författare och Granskare är vanligtvis samma person.
- Sekreterare, som är personen som antecknar alla förslag och påpekningar under granskningens gång.

Kodgranskningen kommer ske iterativt genom att man tar den senaste versionen av kod och låter granskaren gå igenom den inför resterande medlemmar av gruppen. Den utsedda sekreteraren antecknar eventuella frågetecken som dyker upp under granskningen. När granskningen är slutförd så går gruppen gemensamt igenom punkterna som antecknats för att komma fram till en lösning som alla förstår och är överens om.

Riktlinjer för kod

Regler som skall eftersträvas:
Namngivning ska i allmänhet ske konsekvent. Liten bokstav på variabler och stor bokstav på klasser. Om en stor bokstav förekommer i ett variabelnamn så ska de vara för att representera ett nytt ord. Kod ska skrivas på så sätt så att de är lättläst och läsaren ska exempelvis inte behöva scrolla för att kunna läsa en hel kodrad. Utöver detta ska namnen, i koden så väl som filerna, vara enkla att förstå och ska inte givas en långsökt döpning.

Checklista för granskningsmöte

De som först och mestadels ska tittas på är om alla riktlinjer uppfylls. Fortsättningsvis så ska man se om man kan finna några metoder eller variabler som inte leder någonstans eller anses “onödiga”. Slutligen ska man överse om all kod i allmänhet uppfyller de krav projektgruppen ställt och så att inga krav har missats.

Dokumentgranskning

För dokumentgranskningsprocessen kommer gruppen som grund använda tre olika roller. Dessa är *Godkännare, Författare* samt *Granskare,* vilka är hämtade och beskrivna ifrån PPS modellen. Dock är dessa roller inte helt anpassade enligt gruppens projekt, så justering görs därför efter behov. *Godkännaren*, enligt PPS modellen, är den som är huvudanledningen till att dokumentet skapades. I gruppens fall kommer det inte finnas en godkännare, utan gruppen som en helhet måste tycka att dokumentet är korrekt skrivit innan det kan anses att det är färdigt. *Författaren* är den gruppmedlemmen eller medlemmarna som ansvarade för att majoriteten av dokument skrevs. De är även de som kommer att justera och ändra dokumentet ifall dokumentgranskningen leder till att det behövs. Författaren är ej färdig förrän dokumentet är helt godkänt. *Granskare* är de gruppmedlemmar som inte var delaktiga i skrivningen av dokumentet. De är ansvariga för frågor och synpunkter gällande dokumentet, och använder checklistan för att se till så att dokumentet kan bli godkänt.
Det första steget för dokumentgranskingen är en förgranskning. Godkännaren, eller i detta fall gruppen, läser igenom dokumentet och försäkrar sig att författaren har förstått uppgiften. Det är en granskning som inte behöver dokumenteras. Den är endast till för att försäkra gruppen att dokumentet berör det som den ska, och att uppgiften hanteras på rätt sätt.

Nästa steg är en preliminärgranskning. I huvudsak är det författaren som ansvarar för denna granskning. Anledningen till att en preliminärgransking utförs är att författaren ska se till så att dokumentet är tillräckligt bra för att granskas av resterande gruppmedlemmar under ett granskningsmöte. Denna granskning utförs då författaren anser sig själv vara till största delen färdig med dokumentet. Eftersom den som preliminärgranskar ett dokument behöver inneha god kännedom om texten samt syftet med den, så tilldelas den uppgiften till författaren. Likt förgranskningen krävs det inte heller att denna granskningen dokumenteras, förutom i kallelsen till granskningsmötet, då det ska framföras vem som ansvarade och genomförde den preliminäragranskningen.

Efter författaren utfört preliminärgranskningen så skall en kallelse till granskningsmöte skickas ut till gruppmedlemmarna. I gruppens fall så delas redan alla dokumenten mellan gruppmedlemmarna via GitHub, men ett dokument innefattande vilka aspekter av dokumenten som skall granskas ska skrivas, då utav författaren. Den ska förklara vad the granskande medlemmarna skall fokusera på inför granskningsmötet.

Under granskningsmötet så går gruppen gemensamt igenom hela dokumentet. Varje del kommenteras av de granskande där de berättar om de har synpunkter eller förslag för förbättring. Detta skall författaren skriva ned, för att sedan kunna justera dokumentet om det så behövs. Är allt i ordning och hela gruppen godkänner dokumentet kan författaren markera den som färdigarbetad. Annars får författaren ändra dokumentet enligt synpunkter, och sedan uppdatera dokumentet så att de granskande kan läsa den nya versionen och komma med eventuella synpunkter, eller endast meddela att de godkänner dokumentet.

Riktlinjer för dokument

Varje dokument skall innefatta de delar som är listade i respektive mall. Gemensamt för alla dokument är syfte, en framsida, dokumenthistorik, innehållsförteckning, en ordlista samt referenser. Därefter skall dokumentet struktureras efter given mall.

All text skall skrivas med typsnittet Arial. Alla överrubriker skall skrivas i fetstil med teckenstorlek *24.* För rubriker gäller fetstil med teckenstorlek *20.* För eventuella underrubriker som används för att förtydliga indelningen av texten skall skrivas i fetstil, teckenstorlek *14.* All vanlig text skrivs i normalstil, teckenstorlek *11.*

Överrubriker skall användas för att markera det övergripande innehållet för den text som följer. Rubriker används för att dela in informationen i grupper av information, där de som hör ihop befinner sig under samma rubrik. Underrubriker används för att ytterligare strukturera informationen om det så behövs. All skriven text, förutom framsidan samt titeln för dokumentet, skall vara vänsterjusterad.

Filer skall namnges på följande sätt. <Dokumentnamn, Grupp Nummer>. Detta gör det enkelt att identifiera vilket dokument det är, samt vilken grupp som har gjort det.

Inom dokumenten förekommer termer associerade med speldesign och spelutveckling. Finns det en svensk översättning av engelska ord skall dessa användas, och en beskrivning ges i ordlistan. Finns det inte en passande översättning skrivs det engelska ordet i kursiv stil, med en tillhörande förklaring i ordlistan. Alla dokument har gemensamt att ett neutralt och objektivt språk skall användas. Ord som “vi” eller “jag” skall undvikas.

Krav skall ges ett unikt ID-nummer. Det är baserat på den grupp kravet tilldelas, där bokstaven för ID-numret är den första bokstaven i gruppens namn. Därefter tilldelas kravet en siffra som speglar vilket krav det är. Är kravet det sjunde att skapas blir det då krav sju.

Checklista för granskningsmöte

- Ta namn på de som deltar i granskningen.
- Författaren kallar till mötet.
- Roller inför nästa möte tilldelas.
- Gruppen som en helhet måste godkänna eller underkänna dokumentet i fråga.
- Författaren ansvarar för att dokumentera granskningen.
- Granskarna ansvarar för sakfrågor med författaren.
- Alla synpunkter skall lämnas innan granskningsmötets slutdatum.

Allmänna Granskningsaspekter

- Begriplighet - Förstår läsaren vad dokumentet handlar om?
- Korrekthet - Stämmer dokumentets innehåll överens med vad det skall innehålla?
- Otvetyghet - Går det att tolka det som skrivits på bara ett sätt?
- Konsistent - Är dokumentet fritt från motstridigheter?
- Sakligt - Är dokumentet fritt från onödig fakta. Håller det sig till syftet?
- Språkligt korrekt - Är texten rättstavad med korrekt grammatik?

Testning

Kravbaserad systemtestning

Den kravbaserade systemtestningen går ut på att de krav som har satts för projektet ska testas på ett strukturerat sätt. De krav som implementeras i programmet väljs ut efter prioriteringen som utformats i kravdokumentent, och deras funktionalitet testas.

Prioritering

Krav prioriteras efter den gradering som de givits i kravdokumentet.

White box-testing: Statement coverage

I metoden statement-coverage så testas varje programsats minst en gång vid varje test. Denna metod täcker av koden mer än både branch- och path-coverage. Då varje möjlig variation i koden testas så ökar chansen att hitta problem i strukturen och därigenom öka systemets stabilitet.

Vid varje test så slås det fast vilket resultat koden borde ge för det värde som förs in. Därefter körs programmet med nämnda värde och resultatet jämförs med det önskade. Överensstämmer dessa inte så görs en mer ingående granskning av koden för att fastställa var felet är och hur det kan åtgärdas.

I och med att kollision prioriteras, se underrubriken “*Prioritering”*, så kommer granskning i huvudsak att ske i klasserna *GameObject*, *MovingObjects* samt *GameWorld* då de innehåller relevanta metoder för kollision mellan objekt i systemet.

Prioritering

Kod som hanterar kollision prioriteras för testing. Detta eftersom kollision med världen och fiender är en central del av plattformsspel. Detta innefattar sammanstöter mellan karaktärer i spelet, plattformar och projektiler. Fungerar något element med kollision inte korrekt så skadas spelupplevelsen troligen i hög grad, varför testning av nämnda kod prioriteras. I detta syfte sker testing i *GameWorld* klassen där den mesta av interaktioner mellan objekt hanteras. Om fel upptäcks kan det vid behov vara lämpligt att testa även klasserna *GameObject* och *MovingObject* då de innehåller metoder för kollision.

Användbarhetstestning: Test rapport

Speltest #1:
Den första personen som spel-testade för space adventures version 1.12. Speltestaren arbetar inom en annan grupp som jobbar på spelet *Lethal Weapon.* Efter speltestet fick man responds som var ganska kort. I och med Space Adventures (version 1.12) inte har så mycket än några plattform samt en animerad gubbe som springer och hoppar var respons från spel testaren:
Han hade inte mycket att kommentera än att den animerade spelaren ser bra ut. Sedan kom lite mera respons som lutar sig en aning vad som måste göras. Plattformarna fångar upp spelaren redan när han hoppar underifrån eller bara nuddar kanten. Kan bero på någon slags debugg som måste fixas.
Den andra var att spelaren inte kan gå lika långt, då det inte går att gå lika långt åt höger som åt vänster. Annat än det fanns det inga andra kommentarer på från speltestaren.

Detta projektet, Space Adventures utgår från metoden *Test rapport,* då det hjälper mycket att få feedback ifrån andra grupper som kan hitta fel eller ge förslag och tips etc. På samma sätt som andra grupper speltestar projektet så speltestar man även för deras spelprojekt och på sådant sätt hjälper man varandra samt att tidigare har man fått feedback ifrån andra grupper ifrån krav dokument samt tidigare kodstycke/prototyp.

Testfall kravbaserad systemtestning

Funktionella Krav

FK1 Kollision med plattform

Förberedelser: Startat programmet
Teststeg:

1. Spelaren ställer sig vid en plattform
2. Spelaren hoppar upp på plattformen
3. Använd så många steg som behövs

Förväntat resultat: Spelaren landar på plattformen och förblir där utan att falla igenom den.

FK2 Kollision med fiende

Förberedelser: Programmet är startat
Teststeg:

1. Spelaren positionerar sig vid en fiende
2. Spelaren kommer i kontakt med fienden
3. Upprepa tills spelarens livskraft blir noll.

Förväntat resultat: Spelaren dör efter tre kollisioner, och förflyttas tillbaka till startpunkten för spelaren.

FK3 Kamera rörelse

Förberedelser: Starta programmet
Teststeg

1. Spelaren trycker på D
2. Spelaren trycker på A
3. Spelaren trycker på Space
4. Spelaren faller den ifrån en plattform.

Förväntat resultat: Kameran förföljer karaktären och fokuserar på den.

FK4 Grundläggande rörelse

Förberedelser: Starta programmet
Teststeg:

1. Tryck på d.
2. Tryck på a.
3. Tryck på mellanslag.

Förväntat resultat: Spelarens karaktär ska flytta sig åt höger, sedan vänster och sist göra ett hopp.

FK5 Skjutbart Vapen

Förberedelser. Starta programmet.
Teststeg:

1. Sikta åt höger.
2. Klicka på skjutknappen.
3. Sikta åt vänster.
4. Klicka på skjutknappen.
5. Klicka på skjutknappen under ett hopp.

Förväntat resultat: Ett skott avfyras först åt höger, sedan vänster, samt medans karaktären befinner sig i ett hopp.

FK6 Döda fiender

Förberedelser: Starta programmet, spelaren rör sig till en fiende.
Teststeg:

1. Sikta på fienden.
2. Klicka på skjutknappen.
3. Skjut tillräckligt med skott för att fiendens livskraft ska bli noll.

Förväntat resultat: Fienden dör och försvinner efter att den har blivit träffad av tillräckligt med skott.

Spårningsmatris

	FK1	FK2	FK3	FK4	FK5	FK6
G1	X		X	X		
G2					X	X
G11		X				

Testfall/materiel för annan metod

Granskningsprotokoll

Läsare hänvisas till respektive granskningsprotokoll dokument.

Granskningsprotokoll

Testrapporter

Testrapport Funktionella Krav: TS1

Id Testfall	Datum	Kod version	Utfört av	Resultat
FK1	170505	1.3	Carl-Magnus Klang	Förväntat resultat.
FK3	170505	1.3	Carl-Magnus Klang	Förväntat resultat.
FK4	170505	1.3	Carl-Magnus Klang	Förväntat resultat.
FK5	170505	1.3	Carl-Magnus Klang	Förväntat resultat.

Analysrapporter

Analysrapport