**Grand Psychedelic Space Voyage Experiment**

**Grupp 23**

**Verifiering och valideringsdokument**

**V 1.6**

**2017-05-11**

**Handlerade Gruppmedlemmar**

Farid Naisan Miran Amin

Nils Lindkvist

Oskar Lloyd

Petter Månsson

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datum | Version | Beskrivning | Författare |  |
| 170327 | 0.1 | Första genomgång av dokument | Miran Amin  Nils Lindkvist  Oskar Lloyd  Petter Månsson |  |
| 170330 | 0.2 | Whitebox-testning har uppdateras med en vald metod. BVA | Petter Månsson |  |
| 170411 | 0.3 | Testrapport för testfall BVA1 tillagd. Stycke under kodgranskning tillagt med beskrivning av kodgranskningsprocess. | Petter Månsson  Nils Lindkvist |  |
| 170412 | 0.4 | Ändrat i testrappor BVA1 | Petter Månsson |  |
| 170413 | 0.5 | Lagt till kodgranskningprotokoll KG001. Ändrat och städat i stycke Kodganskning, metodbeskrivning. | Nils Lindkvist |  |
| 170413 | 1.0 | Dokument granskat för inlämning till retrospekt 2 | Petter Månsson |  |
| 170427 | 1.1 | Whitebox code coverage | Oskar Lloyd |  |
| 170428 | 1.2 | Rensat upp mallens gamla texter och fixat till dokumentgranskningen snabbt | Miran Amin |  |
| 170428 | 1.3 | Fixat header och framsida version nummer. Uppdaterat innehållsförteckning | Petter Månsson |  |
| 170503 | 1.4 | Granskningsmöte samt protokoll och checklista | Miran Amin |  |
| 170503 | 1.5 | Skrivit tester för kravbaserad systemtesning | Oskar Lloyd  Petter Månsson |  |
| 170511 | 1.6 | Testrappot för BVA2 | Petter Månsson |  |
| 170517 | 1.6 | Testrapport Kravbaserad testning | Oskar Lloyd |  |
| 170517 | 1.6 | Coverage testning, bilagor och formatering | Oskar Lloyd |  |
| 170522 | 1.6 | Testrapport Kravbaserad testning | Oskar Lloyd |  |

**Innehåll**

[Verifiering och valideringsdokument 4](#__RefHeading___Toc1058_727641771)

[Syfte 4](#__RefHeading___Toc1060_727641771)

[Ordlista 4](#__RefHeading___Toc1062_727641771)

[Referenser 4](#__RefHeading___Toc1064_727641771)

[Testprocess 5](#__RefHeading___Toc1066_727641771)

[Granskning 6](#__RefHeading___Toc1068_727641771)

[Riktlinjer för kod 6](#__RefHeading__1075_1449797335)

[Checklista för granskningsmöte 6](#__RefHeading__1077_1449797335)

[Dokumentgranskning 7](#__RefHeading___Toc1070_727641771)

[Checklista för granskningsmöte 8](#__RefHeading___Toc1072_727641771)

[Granskningsprotokoll 9](#__RefHeading___Toc1074_727641771)

[Granskningsprotokoll Kodgranskning KG001 10](#__RefHeading___Toc1076_727641771)

[Testning 12](#__RefHeading___Toc1078_727641771)

[Kravbaserad systemtestning 12](#__RefHeading___Toc1080_727641771)

[Kravbaserad systemtestning testfall 12](#__RefHeading___Toc1082_727641771)

[Spårningsmatris kravbaserad systemtestning 12](#__RefHeading___Toc1084_727641771)

[White box-testning: Code coverage 13](#__RefHeading__1083_1449797335)

[White box-testning: Boundary Value Analysis 13](#__RefHeading__1085_1449797335)

[Testrapporter 14](#__RefHeading___Toc1086_727641771)

[Testrapport BVA1 14](#__RefHeading___Toc1088_727641771)

[Testrapport BVA2 15](#__RefHeading___Toc1090_727641771)

[Spårningsmatris whiteboxtester 16](#__RefHeading___Toc1092_727641771)

[Testrapport Kravbaserade Systemtester KSys1 16](#__RefHeading___Toc1094_727641771)

[Spårningsmatris Kravbaserade Systemtester 16](#__RefHeading___Toc1096_727641771)

[Testrapport Coverage 1 16](#__RefHeading___Toc1185_2433838990)

[Testrapport Coverage 2 17](#__RefHeading___Toc1187_2433838990)

[Bilagor 18](#__RefHeading___Toc1189_2433838990)

# Verifiering och valideringsdokument

# Syfte

Dokument som innehåller information om hur projektet utför granskningar av kod, dokument och vilken testning som utförs.

# Ordlista

BVA Boundary Value Analysis

IDE Integrated Development Environment, program/miljö för programmering med textredigerare för källkodsredigering, kompilator för givet programspråk och debugger

# Referenser

### [1] TSUI, F. F., KARAM, O. AND BERNAL, B.

#### **Essentials of software engineering**

F. Tsui, O. Karam and B. Bernal, *Essentials of software engineering*, 3rd ed. John & Bartlett Learning, 2014, p.204-205 p.207-208. ISBN-13: 9781449691998

# Testprocess

Testning sker i slutet av varje inkrementell release av produkten.

Testningen kommer utföras av 2 av projektets deltagare. De kommer ansvara för att utforma testerna och välja lämpliga testmetoder.

Dokumentering för tester sker enligt riktlinjer som finns i detta dokument. (Se avsnitt om dokument granskning).

Fel som identifieras vid testning dokumenteras i en log som sedan alla utvecklare har tillgång

till. Felen kommer vara prioriterade så att mer kritiska problem kan lösas först.

Blackbox och whitebox testning kommer utföras. Testklasser bifogas som bilaga.

# Granskning

**Kodgranskning**

Kodgranskning utförs utifrån metoden ”Fagan inspection” [1, Ch. 10.5] med en del anpassningar för att bättre fungera med en mindre projektgrupp om 4 personer.

1. **Planering** sker mer informellt än beskrivet i metoden. Moderatorn meddelar utvalda granskare om vilken kod/klasser som ska granskas och ser till att koden är exekverbar.
2. **Overview** skippas helt då alla är insatta i koden sedan innan.
3. **Förberedning** utförs enligt metodbeskrivningen.
4. **Granskningsmöte** ändras lite från beskrivningen i boken. Eftersom alla personer i projektgruppen skriver kod blir det svårt att inte vara ”author” till kod som ska granskas. Detta bortses därför och moderatorn agerar ”reader”.
5. **Rework** utförs efter arbetsfördelning i slutet av granskningsmötet. Mindre korrigeringar som namnändringar och indentering utförs av moderatorn. Större omarbetningar delas upp och utförs i regel av personen som ursprungligen skrivit koden för detta område.
6. **Follow-up** utförs som beskrivet i metoden.

## Riktlinjer för kod

**Källkods filer:**

Samtliga filers text kodning skall vara UTF-8

**Källkods struktur:**

Metoder ska separeras med en tom rad.

Main metod ska alltid placeras i slutet av en källkodsfil.

Samtliga mönster skall implementera interfacet PatternInterface.

**Formatering:**

Formatering av fil sker via vald IDE’s auto formatering.

Måsvingar ska användas för att öppna och stänga en selektion eller loop.

Tomma kod block skall startas och avsluta på samma rad.

Indentering ska göras med en tabs avstånd.

**Namngivning:**

Filnamn skall anges med små bokstäver och inga mellanrum. Mellanrum ersätts med ett understreck. Endast a-z och 0-9 får användas.

Klassnamn: Första bokstaven i varje ord skall vara stor t.ex AhlgrensBilar.

Metodnamn: Första bokstaven i varje metodnamn skall vara liten. Resterande ord skall alltid börja med stor bokstav t.ex setColorAhlgrensBil.

Variabelnamn: Små bokstäver används. Om variabelnamnet innehåller flera ord skall varje ord utom det första börja på stor bokstav t.ex. byteArray.

## Checklista för granskningsmöte

Är indenteringen korrekt utförd? Inga onödiga blanksteg/indrag och inga dubbelhopp i blankrader?

Är klasser, metoder och variabler rätt namngivna?

Finns det motsägelser i booleska utrycka i t.ex. selektioner eller iterationer?

Finns önskade konstruktorer och är de tydliga/korrekt utformade? (Tilldelas instansvariabler som tänkt eller görs detta på andra ställen i koden?)

## Dokumentgranskning

Dokumentgranskningen utförs utifrån metoden ”Fagan inspection” [1, Ch. 10.5] fast med ändringar i hur utförande kommer att göras på grund av att gruppmedlemmarna är lättillgängliga och kommunikativa samt endast 4 personer.

Projektets granskningar sker i form **planeringar** av när och hur granskningarna skall göras. Granskningarna kommer då att gås igenom inom grupp där det bestäms vilken projektdeltagare som ansvarar för vilket dokument. Deadline sätts för en **individuell granskning** där samtliga medlemmar skall ha gjort granskningen för ansvarat dokument. Under vecko/discord möte ingår det ett **dokumentgranskningsmöte** där medlemmarna går ihop i par och granskar/markerar fel i dokumenten som sedan vidarebefordras till nästa par och vice versa. Efter slutförda granskningar så får medlemmarna tillbaka de ansvarade dokumentet som skall **rättas individuellt**. Projektets gång kommer bygga vidare på dokumenten som leder till vidare dokumentgranskningar av samma dokument(iterationer).

Dokumenten som granskas, anpassas retrospektsvis utifrån vilka dokument som är aktuella för redovisning samt äldre dokument vid förändringar med tidens gång. Detta är prioritering för att gruppen skall vara i fas enligt projektets struktur i form av uppdelningar i retrospekter samt bibehålla rätt dokumentation genom hela processen för alla dokument.

### Riktlinjer för dokument

**Typsnitt och formatering:**

Typsnitt: Det typsnitt som används i den mall man jobbar i. Ifall ett nytt dokument skapas utan mail skall Times New Roman användas.

Formatering: En blank rad mellan rubrik. Undantag för rubrik med underrubrik.

**Enhetlig ordlista:**

Projektdeltagare – ej gruppmedlem

Produkt – ej app,tjänst eller program

Användare – ej slutanvändare.

## Checklista för granskningsmöte

Author: Miran Amin

Närvarande: Nils Lindkvist

Oskar Lloyd

Petter ”Pidde” Månsson

Hur har formateringen gått under skrivandet av dokumenten?

Vilket typsnitt? Storlek? Radavstånd? Storlek på rubriker alternativt underrubriker?

Blank-rad mellan stycken

Protokoll för gemensam granskning av VoV-dokument:

VoV-dokumentet granskas på grund av att det är det största dokumentet att granska som också kan leda till fler misstag.

Allting är rättad och ser bra ut.

# Granskningsprotokoll

|  |  |  |  |
| --- | --- | --- | --- |
| **Id granskning** | **Datum** | **Kod/dokument version** | **Utfört av** |
| KG001 | 17-04-11 | V0.3 | Nils Lindkvist (moderator, granskare)  Petter Månsson (granskare) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Granskningsprotokoll Kodgranskning KG001

### MainActivity.class

Indentering/formatering: ok

Klassnamn: ok

Metodnamn: ok

Variabelnamn: ok

Konstruktorer: ok

Övrigt: -

### PatternActivity.class

Indentering/formatering: ok

Klassnamn: ok

Metodnamn: ok

Variabelnamn: ”sC” ska döpas om till ”soundConverter”.

Konstruktorer: ok

Övrigt: Instansvariabler ”demo” och ”integer” används ej och ska raderas.

Case ”demo” på rad 36, tilldelning till ”demo”-objekt ska raderas, tilldelning new VisualizerDemo sker direkt till pApplet.

### SoundConverter.class

Indentering/formatering: Dubbla/felaktiga blankrader ska raderas.

Klassnamn: ok

Metodnamn: ”getSoundBytes()” namn ska ändras till ”getFftBytes()” för ökad tydlighet.

”link()” namn ska ändras till ”init()” då metodens syfte har ändrats.

Variabelnamn: ”mFFTBytes” namn ska ändras till fftBytes.

”mSoundBytes” namn ska ändras till ”waveBytes”.

Konstruktorer: ok

Övrigt: Redundans i metoder ”updateVisualizer()” och ”updateVisualizerFFT()”, dessa ska raderas och kod flyttas ner till ”onWaveFormDataCapture()” och ”onFftDataCapture()”.

Get-metod ska läggas till för returnering av ”waveBytes[]”.

Javadoc önskad, läggs till vid senare tillfälle.

### VisualizerDemo.class

Indentering/formatering: Dubbla/felaktiga blankrader ska raderas.

Klassnamn: Namn ska ändras till PatternController då klassens syfte har ändrats.

Metodnamn: ok

Variabelnamn: ”soundBytes” namn ska ändras till ”fftBytes” för konsistens.

”sC” namn ska ändras till ”soundConverter”.

Konstruktorer: ok

Övrigt: -

### PiddePattern.class

Indentering/formatering: Dubbla/felaktiga blankrader ska raderas.

Indentering i metod ”drawShape()” ska justeras.

Klassnamn: Namn ska ändras så att alla Pattern-klassers namn inleds med ”Pattern” följt av specifikt namn.

Metodnamn: ”setOK()” och ”getOK()” namn ska ändras till något mer beskrivande.

”setSoundBytes()” namn ska ändras till ”updatePattern()” och parametern ”bytes[]” namn ska ändras.

Variabelnamn: ok : boolean, namn ska ändras till något mer beskrivande.

Konstruktorer: ok

Övrigt: Instansvariabeln ”soundBytes” är överflödig och ska raderas. Alla användningar i koden ska ersättas till metodvariabler av samma typ.

# Testning

## Kravbaserad systemtestning

Kravbaserad testning sker med hjälp av blackboxtestning[1]. Fall kommer utformas efter de krav som faställts i kravdokumentet. Testerna utförs inför släpp av produkten som helhet innan sprint 3.

**Priotering av tester**

Tester kommer utföras för samtliga must krav.

# Kravbaserad systemtestning testfall

**Starta produkt och visa mönster**

**1.1:** Starta produkt utan musik och visa mönster

1. Produkt startas utan musik
2. Gå vidare från huvudmeny
3. Val av mönster i meny

Förväntat resultat: Mönster visas men reagerar ej då ingen musik spelas från enheten.

**1.2:** Starta produkt med musik och visa mönster

1. Produkt startas med musik från enheten
2. Gå vidare från huvudmeny
3. Val av mönster i meny

Förväntat resultat: Mönster visas och reagerar på musik som spelas från enheten.

**Skalbarhet**

**2.1:** Rotera skärm

1. Rotera skärm vid visning av mönster

Förväntat resultat: Mönster anpassar sig efter den nya vyn.

**2.2:** Mönster anpassas efter mobilenhets skärmstorlek

1. Produkt startas på enhet

Förväntat resultat: Mönster visas i helskärm.

**Skapa egna mönster**

**3.1:** Skapa eget mönster

1. Välj PatternEditor i mönstermenyn

2. Lägg till en bakgrund och förgrund från menyn

Förväntat resultat: Mönster visas med vald bakgrund och förgrund

# Spårningsmatris kravbaserad systemtestning

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tester | ANV1 | ANV2 | SyS1 | SyS2 | SyS3 | SyS4 | SyS5 | SyS6 | AB1 | AB2 |
| 1.1 | X |  |  |  | X |  |  |  |  | X |
| 1.2 | X |  | X | X | X | X |  | X | X | X |
| 2.1 |  |  |  |  |  |  | X |  |  |  |
| 2.2 |  |  |  |  |  |  | X |  |  |  |
| 3.1 |  | X |  | X | X |  |  |  |  | X |

## White box-testning: Code coverage

För att säkerställa att tillräckligt med kod har blivit testat med whitebox-tester kommer samtliga tester generera en täckningsrapport för att avgöra hur mycket som blivit testat. Rapporter kommer genereras med hjälp av Gradle plugin för Android Studio. I den första fasen av testning sammanställs hur mycket av koden som blivit täckt av tester och används som underlag för att avgöra vilka mer tester som kommer behövas. En första rapport kommer produceras inför sprint 3 och en slutrapport produceras inför sprint 4.

### Prioritering

## White box-testning: Boundary Value Analysis

Tester under denna del kommer vara designade för att testa gränsvärden. Boundary Value Analysis(BVA)[1] kommer användas som metod för att finna gränsvärden för testning.

Produkten skall generera mönster beroende på värden som skickas in i olika algoritmer.

Då passar BVA utmärkt för att testa så att mönstrena reagerar som förväntat på den data man skickar in.

BVA går ut på att identifiera vart i ens kod där intervaller eller värden används för att något specifikt ska hända. Sedan identifieras vilka värden som ligger på gränsen för uttrycket och det är de värdena som skall vara med i testfallen.

Ett exempel: ett intervall som är på 0-10. För att testa gränsvärden skall värden som ligger på gränsen matas in. Då måste t.ex värden som -1,0,10,11 för att se till att koden uppför sig som förväntat.

**Priotering**BVA tester kommer utföras på kod där det finns gränsvärden att testa.

# Testrapporter

Här samlas alla projektets testrapporter. Var testrapport får en egen rubrik och detta registreras sedan i en spårningsmatris.

## 

## Testrapport BVA1

Testklass:TestPatternPidde, finns att hitta i projektets källkod.

Testmetod: soundByteArrayTest(byte[])

Testar gränsvärden i PatternPidde som bestämmer hur linjer ritas ut i ett fönster beroende på värden som matas in i en byte array. I PatternPidde finns 4 st olika linjer vars längd bestäms av

värdet i en variabel som kallas line + en siffra för att representera 1 av 4 linjer.

Förväntat resultat: Var linjes värde ökas när rätt värde för linjens intervall matas in.

Intervallet är uppdelat i 4 olika delar som ökar var linjes längd beroende på värde. Alla negativa värden omvandlas till positiva för att korrekt öka linjenslängd.

Intervall av giltiga värden: 0-127.

line1 intervall: 0-31

line1 förväntat värde: 2

line2 intervall: 32-63

line2 förväntat värde: 64

line3 intervall 64-95

line3 förväntat värde: 128

line4 intervall: 96-127

line4 förväntat värde: 192

Identifierade gränsvärden: 0,-1,1,-32,32,-64,64,-96,96,128

Värdena testas genom att matas in i en byte array. Denna array tas emot av en metod och ökar värde i olika variabler beroende på värdet.

Värdet 128 ger fel i testkoden då det värdet inte kan hållas i en byte array.

Alla andra värden ger förväntat resultat.

## Testrapport BVA2

Testklass: TestPatternPidde, finns att hitta i projektets källkod.

Testmetod: updatePatternTest(byte[],byte[])

Testning av den metod som ser till att var linje som ritas i mönstret PatternPidde får rätt värdet tilldelat. Var linje har ett bestämt intervall av värden den kan ta emot.

Testet körs som ett unit test med anrop till assertTrue() där ett booelskt uttryck matas in för att representera det förväntade resultatet.

En getter metod finns deklarerat i PiddePattern för att kunna hämta ut var linjes värde.

Linjens värde sparas i en array där position 0-7 representerar vars en linje. Position 0 är lika med linje 1 osv.

Gränsvärden har identifierats för var intervall. Nedan redogörs vilka värden och förväntat resultat.

Intervall av giltiga värden: 0-127.

Identifierade gränsvärden: -1,0,1,-16,16,-32,32,-48,48,-64,64,-80,80,-96,96,-112,112,128

linje1 intervall: 0-15

linje1 förväntat värde: 2

linje2 intervall: 16-31

linje2 förväntat värde: 32

linje3 intervall 32-47

linje3 förväntat värde: 64

linje4 intervall: 48-63

linje4 förväntat värde: 96

linje5 intervall: 64-79

linje5 förväntat värde: 128

linje6 intervall: 80-95

linje6 förväntat värde: 160

linje7 intervall: 96-111

linje7 förväntat värde: 192

linje8 intervall: 112-127

linje8 förväntat värde: 224

Efter utfört test har samtliga linjer förväntat värde. 128 gick ej att mata in då det värdet ej kan sparas i en byte[].

## Spårningsmatris whiteboxtester

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id testfall** | **datum** | **Kod/dokument version** | **Utfört av** | **Resultat** |
| BVA1 | 17-04-11 | V0.3 | Petter Månsson | line1 värde: 2  line2 värde: 64  line3 värde: 128  line4 värde: 192 |
| BVA2 | 17-05-11 | 1.6 | Petter Månsson | Linje1: 2  Linje2: 32  Linje3: 64  Linje4: 96  Linje5: 128  Linje6: 160  Linje7: 192  Linje8: 224 |
|  |  |  |  |  |
|  |  |  |  |  |

## Testrapport Kravbaserade Systemtester KSys1

1.1: Mönster startar och står still. Som förväntat.

1.2: Mönster rör sig. Som förväntat.

2.1: Mönster vrids och anpassas. Som förväntat.

2.2: Mönster visas i helskärm. Som förväntat.

## KSys2

3.1: Eget mönster visas. Som förväntat

## Spårningsmatris Kravbaserade Systemtester

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id testfall** | **datum** | **Kod/dokument version** | **Utfört av** | **Resultat** |
| KSys1 1.1 | 17-05-17 | V1.6 | Oskar Lloyd | Resultat som förväntat |
| KSys1  1.2 | 17-05-17 | V1.6 | Oskar Lloyd | Resultat som förväntat |
| KSys1  2.1 | 17-05-17 | V1.6 | Oskar Lloyd | Resultat som förväntat |
| KSys1  2.2 | 17-05-17 | V1.6 | Oskar Lloyd | Resultat som förväntat |
| KSys2 3.1 | 17-05-22 | V1.6 | Oskar Lloyd | Resultat som förväntat |

## Testrapport Coverage 1

Testare: Oskar Lloyd

Datum: 17-04-27

Se bilaga Coverage rapport 1.

Dom första unittesternas täckning dokumenteras. Klasser som prioriteras till nästa fas i testningen kommer vara java-klasser som inte är beroende av externa bibliotek t ex PApplets. Mer täckning på controllers och patterns kommer behövas.

## Testrapport Coverage 2

Testare: Oskar Lloyd

Datum: 17-05-17

Se bilaga Coverage rapport 2.

Samtliga controller klasser har samtliga metoder som inte är beroende av externa bibliotek testade. PatternOskar och PattenPidde så gott som heltäckt. Pga att rapporten dokumenterat metoder som inte är tänkta att testas är täckningen något missvisande men närmare titt på enstaka kodrader tyder på högre täckning.

# Bilagor

Coverage rapport 1

Coverage rapport 2

Testklasser