**Grand Psychedelic Space Voyage Experiment**

**Grupp 23**

**Verifiering och valideringsdokument**

**V 0.5**

**2017-04-12**

# Dokumenthistorik

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
|  |  |  |  |

[Fler rader läggs till efterhand som det behövs. Utifrån beskrivningen ska det gå att förstå vad personen gjorde för typ av ändring. Så bara en text som säger ”ändring” räcker inte. Texten behöver exempelvis säga ”Lagt till testfall TF23-28” eller ”Infogat testrapport TR3”

Det kan finnas flera författare. Endast de som faktiskt är aktiva med att skriva texten listas som författare. Om man är två som sitter och jobbar tillsammans och en skriver men man hela tiden aktivt diskuterar det som skrivs så kan bägge personerna stå som författare. Om man har ett möte i hela gruppen där man diskuterar saker, en person tar anteckningar och skriver sedan rent detta i dokumentet så står endast denna person som författare.]

**Innehåll**

[Dokumenthistorik 2](#_Toc476748931)

[Verifiering och valideringsdokument 4](#_Toc476748932)

[Syfte 4](#_Toc476748933)

[Ordlista 4](#_Toc476748934)

[Referenser 4](#_Toc476748935)

[Testprocess 5](#_Toc476748936)

[Granskning 7](#_Toc476748937)

[Kodgranskning 7](#_Toc476748938)

[Dokumentgranskning 7](#_Toc476748939)

[Testning 9](#_Toc476748940)

[Kravbaserad systemtestning 9](#_Toc476748941)

[White box-testning: <Namn på specifik metod> 9](#_Toc476748942)

[Användbarhetstestning: <Namn på specifik metod> 9](#_Toc476748943)

[Användbarhetsanalys 11](#_Toc476748944)

[Användbarhetsanalys: <Namn på specifik metod> 11](#_Toc476748945)

[Testfall kravbaserad systemtestning 12](#_Toc476748946)

[<Eventuell underrubrik för indelning av testfall> 12](#_Toc476748947)

[<Eventuell underrubrik för indelning av testfall> 12](#_Toc476748948)

[Spårningsmatris 12](#_Toc476748949)

[<Testfall/materiel för annan metod> 13](#_Toc476748950)

[Granskningsprotokoll 14](#_Toc476748951)

[Granskningsprotokoll <granskningstyp> <Id för protokollet> 14](#_Toc476748952)

[Testrapporter 15](#_Toc476748953)

[Testrapport <testtyp/kategoritestfall> <Id för rapporten> 15](#_Toc476748954)

[Analysrapporter 16](#_Toc476748955)

[Analysrapport <analysmetod> <Id för rapporten> 16](#_Toc476748956)

# Verifiering och valideringsdokument

# Syfte

<Text som beskriver syftet med dokumentet. Vad det innehåller och hur det relaterar till projektet. Det kan också beskrivas vad som inte står här om det förtydligar vad syftet är eller inte är.>

# Ordlista

<ord> <förklaring>

<ord> <förklaring>

[Ta upp förkortningar och uttryck som används i dokumentet och som kanske inte är självklara för en läsare som inte är insatt i projektet. Tänk också på att ta upp begrepp som skulle vara svåra att förstå för en läsare som skulle kunna vara en presumtiv kund eller referensperson i forma v exempelvis slutanvändare. Ordlistan ska ordnas i bokstavsordning.]

# 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.207-208. ISBN-13: 9781449691998

<referens>

[Använd ett referenssystem och utforma referenser konsekvent enligt detta. Exempel på referenssystem enligt IEEE som är vanligt i tekniska dokument hittas på:

<http://libguides.murdoch.edu.au/c.php?g=246207&p=1640218>]

# Testprocess

<Text som beskriver den testprocess som används. Tänk på att denna ska matcha utvecklingsprocessen i projektplanen. Som minst ska kravbaserad systemtestning utföras med utgångspunkt i kravdokumentet. Beskrivningen ska ange hur testning genomförs, när testning sker i förhållande till utvecklingsprocessen och hur denna dokumenteras. Det ska också beskrivas hur eventuella problem som testningen identifierar ska hanteras.>

[Generellt om referenser till beskrivningar av metoder i detta dokument: Detta kan vara (kurs)litteratur eller någon annan relevant källa. Om en referens som utgörs av en webbsida används så ska denna vara att betrakta som information med bekräftad auktoritet på området. Det vill säga Wikipedia eller liknande fungerar inte. Webbreferenser ska vara etablerade expertsidor så som officiella webbsidan för XP, Agile Alliance eller webbsidor underhållna av erkända branschorganisationer så som IEEE eller ACM.]

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.

Endast whitebox testning kommer utföras.

# 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.

**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

[Del av individuell fördjupning. Någon i gruppen måste ta ansvar för denna.]

< Text som beskriver en process för en formell inspektion för dokumentgranskning och hur denna genomförs i det här projektet. Texten ska innehålla referenser till minst en beskrivning av den metod som man utgår ifrån. Den beskrivning av metoden man utgår ifrån behöver sannolikt anpassas till ert projekts förutsättningar. Texten ska beskriva vilka anpassningar ni gör och hur dokumentgranskning genomförs i ert projekt. Texten ska också ange hur man väljer ut vilka dokument som ska granskas och motivera varför detta prioriteringssätt är lämpligt. Checklistor och mängd dokument ska vara så omfattande att ett granskningsmöte på 2 timmar fylls ut. Se även information under granskningsprotokoll mot slutet av dokumentet.>

Projektets formella granskningar sker genom planeringar av när/hur granskningarna skall göras utefter handledarens kommentarer på dokumenten. 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)

### Riktlinjer för dokument

<Här listas de riktlinjer som ska användas av gruppen för de dokument man skriver. Dessa ska omfatta hur text struktureras, hur man namnger filer, vilka typsnitt och layout, här ska definieras vilka ord som ska användas för vissa saker så att man är konsekvent i sin text. Det krävs även vissa mer dokumentspecifika riktlinjer som anger hur exempelvis krav ska utformas, hur användningsfallsbeskrivningar ska skrivas i designdokumentet eller hur testfall för kravbaserad testning ska skrivas. Om man vill så kan dessa lyftas ut som ett separat dokument men det ska då ges en referens till detta dokument här.>

**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

<Här visas den checklista som ska användas för granskningsmötet. Om man vill så kan denna lyftas ut som ett separat dokument men det ska då ges en referens till detta dokument här.>

# Testning

## Kravbaserad systemtestning

<Text som beskriver kravbaserad systemtestning och hur denna genomförs i det här projektet. Texten ska innehålla referenser till minst en beskrivning av den metod som man utgår ifrån. Den beskrivning av metoden man utgår ifrån behöver sannolikt anpassas till ert projekts förutsättningar. Texten ska beskriva vilka anpassningar ni gör och hur kravbaserad testning genomförs i ert projekt.>

### Prioritering

<Text som beskriver hur krav prioriteras för testning med denna metod.>

## White box-testning: <xxXXxx> OSKI

[Två personer kan som sin individuella fördjupning utföra någon form av white-box-testning. Exakt vilken white-box-metod som används lämnas till personen/personerna och gruppen att bestämma – exempelvis enhetstest, statement coverage eller något annat. Om två personer arbetar med white-box-testning så måste dessa två personer arbeta med olika metoder för detta. Två avsnitt med varsin rubrik enligt ovan, en för varje metod, behövs då.]

<Text som beskriver den white-box-metod som används. Texten ska innehålla referenser till minst en beskrivning av denna metod som man utgår ifrån. Den beskrivning av metoden man utgår ifrån behöver sannolikt anpassas till ert projekts förutsättningar. Texten ska beskriva vilka anpassningar ni gör och hur testmetoden tillämpas i ert projekt.

Texten ska referera till de kodfiler med testkod/testfall som sannolikt behövs och tala om var dessa hittas och vad de heter. Filer med testkod ska lämnas in tillsammans med dokumentation när denna frågas efter (exempelvis vid RS-deadlines eller slutinlämning).

Det ska som ett absolut minimum finnas 20 relevanta testfall. Om en första prioriterad kod inte genererar så många testfall för metoden så ska ytterligare kod testas.>

### Prioritering

<Text som beskriver hur kod-moduler prioriteras för testning med denna metod. Prioriteringen ska motiveras till varför det är lämpligt att prioritera på detta vis.>

## 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.

# Testfall kravbaserad systemtestning

## <Eventuell underrubrik för indelning av testfall>

**<id><Namn på testfall>**

Förberedelser: <text>

Teststeg:

1. <text>
2. <text>
3. <använd så många steg som behövs>

Förväntat resultat: <text>

**<id><Namn på testfall>**

Förberedelser: <text>

Teststeg:

1. <text>
2. <text>
3. <använd så många steg som behövs>

Förväntat resultat: <text>

## <Eventuell underrubrik för indelning av testfall>

**<id><Namn på testfall>**

Förberedelser: <text>

Teststeg:

1. <text>
2. <text>
3. <använd så många steg som behövs>

Förväntat resultat: <text>

[Utformandet av testfallen kan utökas om behov finns. Se exempel i dokumentationsstödet.]

## Spårningsmatris

[Annan variant på diagram kan användas eller bild infogas. Matrisen blir betydligt större och kan delas upp i flera matriser utifrån olika kategorier av krav eller testfall. Ett X nedan i matrisen indikerar att kravet testas av testfallet.]

### <Eventuell underrubrik>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | <id krav> | <id krav> | <id krav> | <id krav> | <id krav> |
| <id testfall> | X | X |  |  |  |
| <id testfall> |  | X |  |  |  |
| <id testfall> |  |  | X |  |  |
| <id testfall> | X |  |  | X |  |
| <id testfall> |  |  |  |  | X |

# <Testfall/materiel för annan metod>

[Dokumentera andra testfall som används med andra metoder eller annan lämplig dokumentation så som exempelvis checklistor eller sammanställningar för annan typ av verifiering och validering.]

# Granskningsprotokoll

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

[Här infogas rapporter från granskningsmöten som granskar kod eller dokument. För varje rapport ska det anges vilken artefakt som granskades, tid och datum för granskningen, vilka som närvarade vid granskningen och i vilka roller. Det behöver tas fram en mall för hur granskningsprotokoll utformas. Detta ska dock ange vilken fil och rad/dokument och stycke ett problem noterats på, typ av problem, hur alvarligt problemet bedöms vara och vems ansvar det är att åtgärda problemet om det ska åtgärdas. Där ska även finnas plats för rekommendationer som ges fr hur ett problem åtgärdas eller om någon mer generell åtgärd bör ske av exempelvis process eller liknande.]

## Granskningsprotokoll <granskningstyp> <Id för protokollet>

<Här infogas protokollet.>

[Observera att det blir flera olika protokoll – minst två från kodgranskning och två från dokumentgranskning.]

## 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.

# Testrapporter

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

## Testrapport <BVA> <BVA1>

Testklass: TestPiddePattern , finns att hitta i projektet.

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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id testfall** | **datum** | **Kod/dokument version** | **Utfört av** | **Resultat** |
| BVA1 | 17-04-11 | V0.3 | Petter Månsson | Resultat som förväntat. |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

[Det kommer sannolikt att behövas fler rader och flera olika testrapporter. Vissa justeringar i utseende av rapporten kan behöva göras beroende på testtyp.]

# Analysrapporter

[Här samlas information om den eller de användbarhetsanalyser som genomförs.]

## Analysrapport <analysmetod> <Id för rapporten>