Slutrapport

**Programbeskrivning**

**A**

Vi ska göra en musikspelare. Så det talar för sig själv att vi ska ha en java-applikation som kan spela upp musik. Hittills är vår idé att man ska kunna spela upp mp3-filer från en angiven mapp. De funktioner vi har kommit fram till att vi vill ha är knappar för play och pause, en lista med all musik som man kan scrolla igenom, ett sätt att justera volymen, möjlighet att kunna skapa spellistor, en shufflefunktion, en eller två repeatfunktioner. Där vi inte riktigt kommit fram till om vi vill kunna spela om en låt eller en hel lista. Sedan vill vi också ha knappar för att hoppa till nästa eller föregående låt och om vi lyckas en överblick över hur långt det är kvar av en låt och möjlighet att hoppa fram eller bak i en låt. Vi ska också ha en liten meny högst upp på spelaren med bland annat en ”hjälpknapp” och lite andra beskrivningar.

**B**

Vi har implementerat play, pause, lista med scrollfunktion, spellistor, shufflefunktion, två repeatfunktioner (spelar om en hel lista och spelar om en enstaka låt), möjlighet att byta låt (både framåt och bakåt), en meny med ett par knappar, bland annat ”hjälp”, en knapp för att köa låtar och en ”slider” som visar hur långt man har kommit i en låt. Det vi inte lyckats/hoppat över att implementera är möjlighet att ändra volymen och att kunna spola i låten med hjälp av en ”slider”.

**C**

Först och främst valde vi att exkludera en ”spolningsfunktion” där man kunde hoppa i låtarna då vi ansåg att vi inte skulle hinna med detta. Volymfunktion kikade vi lite på men insåg att java hade dåligt stöd för detta när man jobbar med mp3-filer. Vi valde att prioritera andra saker, bland annat att få befintliga funktioner att fungera bättre och tänkte oss att om vi fick tid över skulle vi försöka lägga till det andra.

En annan funktion vi kände oss tvungna att bortprioritera var Sök/Filter-funktionen. Detta var en funktion vi kom på att vi ville ha alldeles för sent, och när vi väl började jobba på den uppkom ett stort antal buggar som vi inte ansåg oss redo att lösa.

Det blev lite mer jobb än vi förväntade oss, främst då det inte finns optimalt stöd för uppspelning och hantering av mp3-filer i Java. Dessutom är det många småsaker som vi inte tänkte på innan som vi kommit att implementera, främst vad det gäller alla knappar för uppspelning från låtlistan. Vi arbetade väldigt mycket med att få detta att fungera bra, då det är kärnan i vårt program.

**Användarbeskrivning**

**A**

En musikspelare kan i princip alla använda. Vi har ingen särskild målgrupp och ska försöka skapa en så intuitiv design som möjligt, så att det inte krävs särskilt stor teknisk kunskap för att kunna använda programmet. Det är dock några frågor som vi kommer ställas inför, bland annat ska sökvägen anges där musiken finns. Personer utan någon teknisk kunskap i området kommer få problem här.

**B**

Inte mycket har ändrats här. Alla kan använda vår musikspelare förutsatt att de har en mapp med mp3-filer och klarar av att ange sökvägen till den mappen. Vi är nöjda med hur designen har utvecklats och det är ungefär hur vi förväntade oss att det skulle se ut. Det ska inte vara några svårigheter för vanliga personer att kunna klicka sig fram och lära sig på någon minut. Vi har bland annat gjort det möjligt att kunna klicka på play på tre olika sätt.

**C**

Vi har som sagt inte ändrat på mycket under denna punkt. Personer i alla åldrar, undantag riktigt små barn, använder musikspelare så som Spotify så vi ansåg inte att det var vettigt att frångå detta och anpassa musikspelaren till en mer specifik målgrupp.

**Användarscenarier**

**A**

Scenario 1: Första scenariet är en mer teknisk person som har erfarenhet av andra musikspelare så som Spotify och itunes. Programmet startas och personen har inga problem att ange sökvägen till mappen där personen har lagrat sin musik. Nu kommer all musik komma upp i listan och personer börjar scrolla för att hitta en låt. Personen markerar en låt i listan och klickar på play. Efter en stund pausar personen låten och bestämmer sig för att skapa en lista. Personen kan skapa en lista enkelt men får tänka lite innan han/hon kommer på hur man lägger till nya låtar i listan. Men det löser sig till slut. Då tidigare erfarenhet med Spotify löser situationen.

Scenario 2: En lite mindre teknisk person. Personen har använt Spotify lite grann tidigare. Har först lite problem att ange sökvägen till sin mapp där musikfilerna ligger, men löser det med hjälp av tipsen som ges i programmet. Sedan ser personen att all musik kommer upp i listan och börjar klicka runt för att sedan trycka på play och en låt börjar spelas. Personen klickar sedan på shuffle-knappen och låter spelaren vara på. Ser sedan att det går att skapa listor men väntar med detta tills han/hon har lärt känna programmet lite bättre.

**B**

Scenario 1: Första scenariet är en mer teknisk person som har erfarenhet av andra musikspelare så som Spotify och itunes. Programmet startas och personen har inga problem att ange sökvägen till mappen där personen har lagrat sin musik. Nu kommer all musik komma upp i listan och personer börjar scrolla för att hitta en låt. Personen dubbelklickar på en låt i listan och låten startas. Efter en stund pausar personen låten och bestämmer sig för att skapa en lista. Personen kan skapa en lista enkelt och har inte heller några problem med att lägga till låtar i den, då detta sker på samma sätt som i Spotify. Personen lägger till ett par låtar i sin spellista, öppnar den och sätter på repeat och börjar lyssna.

Scenario 2: En lite mindre teknisk person. Personen har använt Spotify lite grann tidigare. Har först lite problem att ange sökvägen till sin mapp där musikfilerna ligger, men löser det med hjälp av tipsen som ges i programmet. Sedan ser personen att all musik kommer upp i listan och börjar klicka runt för att sedan trycka på play när en låt är markerad den börjar sedan spelas. Personen klickar sen på shuffle-knappen och låter spelaren vara på. Ser sedan att det går att skapa listor men väntar med detta tills han/hon har lärt känna programmet lite bättre.

**C**

Vi ändrade bara lite i det första scenariet för att få det lite mer anpassat till de funktioner vi har implementerat.

**Testplan**

**A**

Vi kommer främst använda oss utav användartestning. Det kommer inte vara särskilt många matematiska beräkningar i vår kod så vi kommer inte behöva ha en lika utförlig testklass som vi skulle valt i vanliga fall. Men vi kommer kontinuerligt testa de funktioner vi lägger till genom att testa att de gör vad vi vill men även testa om de kan göra saker vi inte vill. I slutet kommer vi även låta andra personer testa vårt program.

**B**

Vi har inte skrivit någon testklass eller liknande när vi testat koden vi skrivit utan har snarare försökt skriva ”påfrestande” kod för att se vad som passat och vad som inte passat bra och använt oss av olika input. Det har blivit väldigt mycket grafisk kod och bara enstaka små klasser där vi faktiskt räknar på vissa saker och matematiska fel kan uppstå. Vi utgick från några exempel för att bygga en grund till hur man hanterar och avkodar frames från musikfiler och detta var i princip det enda där vi fick göra en del matematiska tester. Väldigt mycket av vår testning har handlat om användartestning där vi själva agerat användare.

Vi har även gjort lite användartester med utomstående personer. En person var mer teknisk kunnig och hade mycket erfarenhet med t ex Spotify. Personen fick börja med att utföra samma scenario som jag skrivit om under användarscenarier. Det var inga problem att ange sökvägen till en mapp där personen hade några musikfiler. Det var inga problem att starta en låt och sedan skapa en spellista. Sedan kom dock några hinder. Det tog lite tid innan personen uppfattade att man lade till låtar i listan genom att högerklicka på en låt och välja ”add to playlist -> namnet på listan”. När personen hade lagt till några låtar och spelat dem i listan uppstod lite problem. Det tog lite tid för personen att hitta knappen som tog en tillbaka till ”huvudlistan”.

Ett problem som vi märkte att vi fick var att eftersom programmet grafiskt sett är relativt likt Spotify, så förväntade sig användaren att vårt program skulle ha samma funktionalitet, däribland att man skulle kunna "dra" låtar till spellistor och att man skulle kunna trycke på "Ctrl+F" för att få fram en sök/filter-funktion.

Personen fick även använda programmet fritt och kunde hitta några småsaker som vi hade missat, som egentligen inte var buggar utan misstag i koden på hur uppspelningen skulle hanteras. Dessutom fick vi kritik för att vi inte har någon funktion för att ändra volymen eller att spola i låten.

**C**

Testningen har gått bra, användartesterna har gett oss mycket och vi tycker att vi har fixat till de funktioner vi satsat på riktigt bra. Vi fick göra en del prioriteringar (bland annat att välja bort volym- och spolningsfunktion), då vi föredrog att ha färre funktioner som fungerar bra än flera som fungerar halvdant. I slutet av projektet när vi kände att vi var hyfsat klara, satsade vi mer på användartestning för att göra funktionerna ännu bättre istället för att försöka oss på att implementera ny saker.

Om vi ska kommentera lite av resultaten vi fick från testningen, kan vi börja med de saker som personer hade problem med att uppfatta. Att det inte är mer intuitivt att hoppa tillbaka till ”huvudlistan” beror på problem att implementera just det och vi valde ett lite sämre alternativ designmässigt, men som fungerar bra. Vi ser det inte som ett problem att folk inte uppfattar att man kan högerklicka för att utföra funktioner omedelbart, utan man måste pröva sig fram lite.

Att vårt program användartestades av människor med varierande nivåer av tekniska kunskaper visade sig vara ett bra verktyg, då det gav en bra bild av hur olika användare använde programmet. Vissa användare snappade direkt upp hur programmet skulle hanteras, medans andra fastnade på saker som vi kanske ser som triviala.

Att vi inte har volym- eller spolningsfunktion är givetvis tråkigt, då detta egentligen är essentiella delar i en musikspelare. Men vi tog besluten att inte prioritera de då vi trodde att spolningsknappen skulle vara svår att implementera och volymfunktionen sköt vi upp på lite för länge. När vi väl insåg att det var knepigt att implementera en volymfunktion i Java avstod vi även från detta. Fortsätter någon av oss koda på detta projekt kommer dessa två funktioner dock få hög prioritet igen.

**Programdesign**

**A**

Vår huvudklass kommer vara själva musikspelaren. Sedan är vi inte helt hundra än på hur många klasser vi kommer ha. Men troligtvis delar vi upp det så att varje del som behöver lite mer kod för att utföra får en egen klass. Vi kommer ha en egen klass för listan med alla låtar och en egen klass för designen och layouten. Vi kommer även ha några andra paneler för t ex alla knappar som sköter uppspelandet av musiken och hanterandet av spellistor och troligtvis kommer dessa också få egna klasser.

Update: Vi kommer ha en huvudklass som är själva musikspelaren. Sedan en klass med all design och layout. Vi kommer ha en klass som hanterar musikfilerna och som använder sig utav att vi tittar på ”frames” i filerna och hämtar ut allt nödvändigt för att lösa problem med t ex pausknapp. Många av de funktioner som behövs i musikspelaren kommer att implementeras i klassen som hanterar frames. Så det kommer förhoppningsvis inte bli för mycket kod i huvudklassen.

**B**

Vi har en huvudklass som sköter hela GUI’t och allt som har med layouten att göra. Denna klass använder sig utav flera olika mindre klasser för bland annat skapandet av spellistor eller själv låtlistan. Sedan har vi en ”huvudklass” för alla uppspelningsfunktioner. I den klassen har vi bland annat play, paus och lite annat. Denna klass har dock en instans till en annan klass som används för att kunna hantera musikfilerna i form av frames. Så det är relativt lite kod i klassen som hanterar uppspelningsfunktioner då många av metoderna bara är som hjälpmetoder.

**C**

Slutresultatet blev ungefär som vi hade tänkt oss. Undantagen blev att delar i programmet som vi först hade tänkt ha som egna klasser istället blev flera metoder i huvudklassen, GUI’t. Detta på grund av att de jobbade med samma variabler och vi tyckte att det var bästa lösningen (t ex panelen där vi startar och pausar låtar). Kortfattat blev den slutliga strukturen en huvudklass (GUI) som använde sig utav flera mindre klasser för att bygga upp designen och två klasser som sköter uppspelning och hantering av mp3-filer.

**Tekniska frågor**

**A**

Hur ska vi lösa problemet med sökväg till musikmappen (alla kanske inte vet hur man gör, dessutom är det osmidigt att behöva mata in detta manuellt), ska vi ha möjlighet att använda flera olika musikmappar?

Panelen som visar hur långt man har kommit i en låt kommer bli svår att implementera, särskilt om vi vill kunna hoppa i låten genom att användaren flyttar på en pekare med musen. Detta kommer vi få klura på en hel del, tar det för långt tid tar vi inte med det.

Ett annat problem är hur vi hanterar spellistorna, hur lägger man till låtar på ett användarvänligt sätt som inte ser dåligt ut designmässigt, hur ser vi till att de sparas även när man stänger ner programmet?

Pausfunktionen är också ett steg i processen som blir jobbigare att implementera, då vi måste arbeta med ”frames”.

**B**

Vår lösning på den första frågeställningen är att man navigerar fram till mappen med alla låtar i. Denna plats sparas sedan, så att du endast behöver mata in den en gång.

Här har vår lösning helt enkelt varit att vi inte har prioriterat att implementerat detta och troligtvis inte heller har gjort det innan inlämningen (återstår att se).

Den tredje frågan löste vi genom att det går att högerklicka på en låt och klicka på ”add to playlist” och för att spara listorna använder vi oss utav en config.properties-fil.

Den sista löste vi också. Vi har tittat på mycket exempelkod hur man kan göra detta och fick det att fungera.

**C**

Vi tycker att vi har löst flera av problemen på ett bra sätt. Vi är väldigt nöjda med lösningen av spellistor och att vi löste pausproblemet som tog lite tid i början. Vi gjorde det även med bättre resultat än andra exempel vi tittat på (mindre fördröjning).

Vi valde att inte prioritera en ”spolningsfunktion” då vi antog att det skulle ta för lång tid och drabba de andra viktiga funktionerna negativt om vi skulle börja lägga ner tid på detta.

Då det Java-bibliotek vi använde oss av inte var det mest avancerade som finns var vi redan från början ganska medvetna om att vårt program aldrig kunde spela upp och pausa låtar lika snabbt som t.ex. Spotify eller iTunes. Om vi hade fått i uppgift att skapa en "felfri" musikspelare hade vi varit tvungna att sätta oss in i hur -mp3 filer är uppbyggda på ett mycket mer djupgående sätt. Så som det blev använder vi egentligen andras bibliotek, ska man göra det "rätt" bör man skapa egna bibliotek och vara helt insatt i hur ljudfilerna ska hanteras.

**Arbetsplan**

**A**

Vi kommer använda oss utav GitHub och från och med den 22 april använda oss utav det för att koordinera vårt arbete. Det vi gjort hittills är att vi har suttit och diskuterat om hela projektet i allmänhet. Sedan har vi tillsammans gått igenom en del kod som Marcus redan har kodat sen tidigare när han byggt ett litet program som hanterar mp3-filer. Så vi båda har satt oss in i projektet. Vi har hunnit lägga en grund till projektet med listor och uppspelning av mp3-filer. Vi har också tagit en del designbeslut.

**26 april:** Till denna dag ska vi ha en grundläggande design med visning av lista och eventuellt några andra funktioner.

**3 maj:** Nu ska vårt program i princip vara klart. Vi ska ha med alla ”måste”-funktioner och vid tid lägga till ytterligare funktioner innan 15 maj.

**15 maj:** Mellan 3 maj och detta datum ska vi ha hunnit göra användartester och ha åtgärdat eventuella buggar.

**B**

Vi har använt oss utav GitHub under hela projektet. Vår uppdelning blev ganska bra också. Marcus jobbade mest med design och GUI’t, Mark jobbade mer med hantering av olika events kopplade till de andra två uppspelningsklasserna.

Vi låg i fas med planeringen den 26 april, men inte till den 3 maj. Utan vi hade i princip en färdig musikspelare med de viktigaste funktionerna klar den 8 maj. Det återstår att se hur mycket vi hinner förbättra innan inlämningen, men vi har i alla fall en musikspelare som vi är nöjda med att redovisa.

**C**

GitHub fungerade sådär. Vi använde det hela tiden och det fungerade för oss, men vi använde det nog på lite fel sätt ibland och det skulle helt klart varit bättre för projektet om vi satt oss in i GitHub ännu mer innan vi startade (dock något som vi kände att vi inte riktigt hade tid med). Planeringen var lite konstig i mitten. Det var ett lite långt spann att sätt deadline nästan två veckor innan inlämningen. Vi har dock legat bra i fas med att bli tillräckligt färdiga för att ha något bra att redovisa den 15 maj. Mittendeadlinen borde kanske legat den 9 maj istället, det skulle varit vettigare.

**Sammanfattning**

Grupparbetet har varit väldigt givande. Vi har blivit introducerade till GitHub, fått lära oss mycket om att skapa ett GUI och hur det kan gå till att arbeta i grupp. Det har även varit givande att faktiskt få skapa ett program som uppfyller flera funktioner, till skillnad från inlämningarna under året där man gör flera små enskilda uppgifter och bara löser små problem.

Troligtvis kommer vi fortsätta att försöka implementera några funktioner så att vi får en mer komplett musikspelare. Dessutom har Marcus varit intresserad av att ha en musikspelare för eget bruk så därför lär den användas och utvecklas mer.