

Projekt med Makey-Makeys

Indledning

I forbindelse med et informatikprojekt, der har kørt fra december 2022 til februar 2023, og som har haft fokus på implementering af løsninger med Makey-Makeys, har vi udarbejdet denne projektrapport.

Rapportens formål er at dokumentere elevernes viden om og evne til at udvikle og implementere en alternativ løsning, som kan gøre en mere eller mindre kedelig lokation på skolen mere spændende, vha. Makey-Makeys.

Rapportens Indhold

Kravspecifikation	2
Overordnede krav	2
Specifikke krav	2
Design	2
Redegørelse for idégenerering og valg af lokation	2
Redegørelse for produktprincip og formål	3
Redegørelse for og brug af <i>Affordance</i>	6
Implementering	6
Hvad er en Makey-Makey?	6
Redegørelse for produktets fysiske udformning og funktionalitet	7
Redegørelse for udviklingsmiljø	8
Beskrivelse af relevante dele af koden	9
Prototypen i aktion	10
Konklusion	11
Diskussion af implementering ift. design og kravspecifikation	11
Mulige forbedringer	11
Litteraturliste	12

Kravspecifikation

Dette projekts kravspecifikation er delt op i to kategorier: *overordnede krav*, som betegner de generelle krav for produktet og rapporten og *specifikke krav*, som betegner vores krav til produktet som gruppe.

Overordnede krav

- Produktet skal bestå af en installation, som gør brug af en Makey-Makey.
- Produktet skal være tilpasset et specifikt sted på skolen.
- Rapporten skal indeholde en redegørelse for vores system. Altså de ting, vi har brugt til at få installationen til at fungere, samt hvordan komponenterne spiller sammen.
- Rapporten skal indeholde en redegørelse for basis-idéen, og hvordan den skal "ændre" det sted, installationen er opsat.
- Rapporten skal indeholde en redegørelse for valg af kodesprog, og hvorfor vi har valgt at arbejde i det udviklingsmiljø, som vi har.

Specifikke krav

- Produktet skal bestå af en installation, tilpasset til fredagscaféen.
- Installationen skal være i stand til at tilføje karakter til den valgte lokation ved afspilning af lyd ifm. interaktion.
- Systemet skal have et fysisk aspekt, som skal sørge for, at det går i et med omgivelserne.

Design

Redegørelse for idégenerering og valg af lokation

Ved ethvert produktudviklingsforløb er det nødvendigt at gennemføre en planlæggende idégenereringsfase - især, hvis man arbejder ud fra designtækningsmodellen¹ eller den iterative udviklingsproces.

Hovedformålet i vores idégenerering var at sikre, at produktet ville være i stand til at opfylde kravspecifikationen, og samtidig kunne fungere som kilden til et lærerigt, og ikke mindst sjovt, forløb.

Idégenereringen bestod af, at gruppen gik en tur rundt på skolen, imens vi slog øjne og ører op for steder, der manglede lidt liv. Vi startede med at gå ned i fredagscaféen, hvor vi hurtigt lagde mærke til et aflåst, mørkt rum, som ses på starten af næste side.

¹ Københavns Universitet (u.d.)



I samspil med det aflåste jerngitter, giver det mørke rum et dystert og mystisk indtryk. Mørket sørger for, at man ikke ved, hvad der sker - eller måske er sket - i rummet. Samtidig sikrer gitteret, at man ikke er i stand til at finde ud af, hvad mørket gemmer på.

I denne situation vil de fantasifulde og nysgerrige måske begynde at undre sig over, hvad rummet gemmer på, og hvorfor det mon er aflåst. De mindre fantasifulde sjæle vil måske slet ikke lægge mærke til, at rummet er der.

Efter vi havde overvejet en række andre, mindre attraktive idéer, fastlagde vi hurtigt, at vi ville udnytte det mørke rums mystik og fremstille et produkt, som skulle være i stand til at give rummet en ny funktion og en historie.

Redegørelse for produktprincip og formål

For at give det aflåste rum en ny og spændende funktion, besluttede vi os for, at vi ville udvikle en pressure-plate, som gennem en Makey-Makey kunne afspille en eller flere lyde i mørket. Disse lyde skulle naturligvis være dystre og hentyde til, hvad der mon var sket i dette rum, som gjorde, at det måtte aflåses.

Selvom rummet egentlig bare er et gammelt opbevaringsrum, ville vi på denne måde give det en historie igennem storytelling.

Essensen i produktet er, at når en person stiller sig på pressure-platen, som ligger lige foran gitteret, afspilles en lyd. Selve fredagsbaren er placeret lige ved siden af gitteret, og derfor vil området omkring gitteret være et område med høj trafik under skolens fredagscaféer.

Flowchart over produktets funktionalitet

For at illustrere produktets og programmets funktionalitet, har vi udarbejdet et flowdiagram.

Et flowdiagram, der også kendes som et "flowchart", er et værktøj, der i høj grad anvendes indenfor software- og webudvikling - men også udvikling og test af fysiske produkter. Flowchartet anvendes også indenfor naturvidenskaben og andre fag, hvor det bruges til at illustrere simple og komplicerede kredsløb.

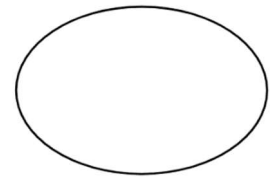
Der er mange måder at opstille et flowchart på indenfor digitalt design og udvikling, men helt generelt kan man sige, at et flowdiagram beskriver produktets "flow" - altså sammenhængen mellem produktets forskellige komponenter, om det er digitalt eller fysisk - hvor der ofte er fokus på brugerens interaktion med produktet.

Herunder ses en tegnforklaring over de figurer, der indgår i dette flowdiagram. Efterfølgende præsenteres og gennemgås hjemmesidens flowdiagram.

Tegnforklaring:

Oval:

Indenfor flowcharting kendes ovalen som *terminatoren*. Terminatoren markerer et start- eller slutpunkt for programmet/produktet. Dette flowchart indeholder kun én terminator, som indikerer, at programmet startes og produktet sættes op.



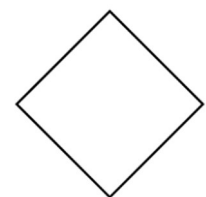
Rektangel:

Rektanglet indikerer en handling eller proces, som foretages i programmet. I dette flowchart består en handling f.eks. af, at programmet afspiller en lyd, når der opstår elektrisk kontakt.

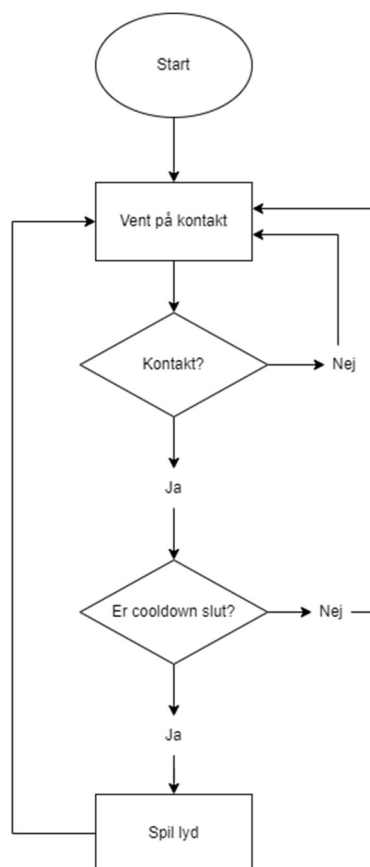


Rombe:

Romben indikerer et valg eller et tjek, der foretages af programmet. I romben skrives et spørgsmål, der indikerer, hvilken betingelse programmet tjekker på. Ud fra romben skrives "Ja" og "Nej", som fører videre til det næste step i flowchartet alt efter, om betingelsen er opfyldt eller ej.



Produktets flowdiagram ses og forklares på næste side.



Inden produktet kan tages i brug, skal det først sættes op fysisk og digitalt. Dette indikeres af terminatoren med teksten "Start".

Når produktet er sat op, begynder programmet at tjekke, om der er elektrisk kontakt til MakeyMakey'ens elektrode. Dette indikeres af rektanglet med teksten "Vent på kontakt" og romben "Kontakt?".

Hvis der ikke er kontakt, fortsætter programmet med at vente på kontakt. Dette indikeres af pilen fra "Kontakt?" til "Nej", som derefter kører tilbage til "Vent på kontakt". Hvis der er kontakt, kører programmet videre til næste step. Dette indikeres af pilen fra "Kontakt?" til "Ja", som kører videre til romben med teksten "Er cooldown slut?".

Nu tjekker programmet, om cooldown stadig er aktiv. Hvis cooldown stadig er aktiv, kører programmet tilbage til "Vent på kontakt." Hvis cooldown ikke er aktiv, spiller programmet en lyd. Dette indikeres af pilen fra "Er cooldown slut?" til "Ja", som fortsætter til rektanglet "Spil lyd".

Når lyden er blevet afspillet, kører programmet tilbage til "Vent på kontakt", og på denne måde fungerer programmet og produktet som en lukket cyklus, der tager et input i form af kontakt til Makey-Makey'ens elektrode, når nogen træder på måtten, og et output i form af en lyd.

Redegørelse for og brug af Affordance

Designprincippet *affordance* blev introduceret af den amerikanske perceptionspsykolog ved navn James J. Gibson i 1966, men blev viderebehandlet i 1988 af den amerikanske forsker, forfatter og professor i datalogi og kognitiv psykologi Donald Norman.

Normans tolkning af princippet bygger på, at genstande "afforder" til brugeren på baggrund genstandens egenskaber. F.eks. afforder en dørs håndtag til, at man kan åbne døren med det, mens en fodbold afforder til, at man kan sparke til den. Denne *affordance* er naturligvis afhængig af brugerens kognitive egenskaber. Brugeren skal være i stand til at forstå genstandens *affordance* – ellers vil der opstå misforståelser ifm. brugen af genstanden.

På den måde kan man sige, at en tings *affordance* fungerer som hentydninger til, hvordan genstanden skal anvendes. Disse hentydninger er baserede på menneskets kognitive egenskaber, og man kan altså ikke antage, at andre arter vil opfatte genstandes *affordance* på samme måde som mennesker. Dog skal det understreges, at målgruppen for vores produkt er mennesker, og derfor kan vi tage udgangspunkt i menneskers kognitive egenskaber.

Vores produkt er ikke baseret på, at brugeren skal vide, at det har en bestemt funktion og kan interageres med. Alligevel indgår en række elementer, som er tiltrækkende og altså afforder til brug af produktet. Her er der primært tale om den dørmåtte, pressure-platen ligger under.

Selvom man ikke ved, at lokationen har en bestemt funktion, er en dørmåtte noget, alle mennesker har en idé om, hvordan man bruger. Når der ligger en dørmåtte på gulvet, vil der med stor sandsynlighed altid være nogle, der på et eller andet tidspunkt træder på den og derved aktiverer produktet. Man kan altså sige, at det i vores tilfælde ikke er et problem, at brugeren "bare går forbi". Selvom brugeren bare går forbi, aktiveres produktet, og det er her det først bliver rigtig sjovt - når produktet formår at overraske brugeren.

Samtidig kan det mørke rum og gitteret virke tiltrækkende, fordi mennesket af natur er nysgerrigt og ønsker at finde ud af, hvad mørket gemmer på. Af den grund vil nogle gå hen til gitteret, for at se, hvad der er inde i mørket, og på samme tid vil de aktivere pressure-platen. I dette tilfælde er der ikke helt tale om *affordance*, men det invaliderer ikke det faktum, at mørket og gitteret kan virke tiltrækkende for nogle personer.

Implementering

Hvad er en Makey-Makey?

En Makey-Makey er et såkaldt udviklings-kit, der bygger på, at stort set alle materialer er strømledende – i større eller mindre grad. Makey-Makey'en består af et circuit-board, ledninger med krokodillenæb samt et USB-kabel, som kan tilsluttes til en computer.

Med Makey-Makey'en har man som udvikler mange muligheder for at lave små sjove eksperimenter – enten hvor Makey-Makey'en står alene eller med Makey-Makey'en som komponent i et større system².

Ved brug af en Makey-Makey, kan man lave alle objekter, der kan lede strøm, om til en knap. Dette sker ved at tilkoble krokodillenæbene til det ønskede objekt. Man skal også sørge for at tilkoble jord-stikket til noget i kredsløbet – ellers vil kredsløbet ikke fungere. Dette er helt basal ellære, og grunden til, at vi har to lag aluminiumsfolie frem for et.

Det simple princip åbner mange muligheder for, hvad man kan bruge en Makey-Makey til. Det kan være alt fra at spille simple platformer-spil med frugt som knapper, til at spille musik ved at bruge en trappe som klaver.

Makey-Makey'en byggede oprindeligt på Atmel 32U4 mikrokontrolleren³, som bruger alle sine 12 input-pins, kombineret med en række af pull-up-resistorer til at genkende lavspændings-signaler fra et ledende materiale. I dag bygger de nyeste modeller på en GPCE4096UA kontroller⁴.

Redegørelse for produktets fysiske udformning og funktionalitet

I afspilningen af lydeffekter indgår en række forskellige komponenter.

Som det første er hele produktet naturligvis bygget op omkring en Makey-Makey. Mere om, hvad en Makey-Makey egentlig er i afsnittet *Hvad er en Makey-Makey?*

Dernæst er det essentielt, at vi har en computer til rådighed, hvor programmet kan køre, og afspilningen af lyden foregå. I første omgang ville vi bruge en RaspBerry Pi 4, men pga. internationale, samfundsmæssige og økonomiske tilstande, er det på nuværende tidspunkt ikke muligt at få en RaspBerry Pi 4 hjem. Derfor bruger vi i stedet et af gruppe-medlemmernes computer.

I en reel implementering af systemet, vil det ikke være muligt at bruge et af gruppemedlemmernes computer, men i og med, at vi i dette projekt blot skal teste produktet i en afgrænset tidsperiode, bliver dette ikke et problem.

Sidst men ikke mindst er det helt essentielt, at brugeren har en måde at aktivere produktet. Dette skal, som nævnt, foregå med en pressure-plate, bestående af i alt fem lag. Det første lag består af en robust, tynd papirskive, der fungerer som den primære kontaktflade, der skal trædes på for at aktivere produktet. Det andet og fjerde lag består af to stykker aluminiumsfolie, som fungerer som elektrisk ledende overflader. Det ene lag er tilkoblet Makey-Makey'ens jordforbindelse, mens det andet fungerer som inputlag. Det tredje er et isoleringslag og består af en meget tynd skive papir, der sikrer, at der ikke hele tiden er kontakt mellem de to lag aluminiumsfolie. Det femte og sidste lag er et beskyttende lag, bestående af almindeligt papir, som beskytter det skrøbelige aluminiumsfolie mod skader.

² Robotnørdieriet (u.d.)

³ Microchip (u.d.)

⁴ DatasheetsPDF (u.d.)

Når en person stiller sig på denne pressure-plate, skabes der kontakt mellem de to lag sølvpapir, igennem hullerne i mellemlaget, og på denne måde sendes der et signal til computeren, som afspiller en lyd.

Til sidst lægges en dørmåtte ovenpå pressure-platen. Denne sikrer både, at produktet holdes skjult, men skaber også affordance i og med, at en dørmåtte er et kendt objekt fra hverdagen, som alle ved, hvordan man bruger - man går på den. Mere herom i afsnittet *Redegørelse for og brug af Affordance*.

Samlet set er pressure-platen designet til at være holdbar og stabil, da den giver en pålidelig og præcis signaludgang, når den aktiveres. Den unikke sammensætning af forskellige materialer sikrer, at pressure-platen kan holde til gentagen brug og slid, uden at der gås på kompromis med funktionalitet og ydeevne.

Redegørelse for udviklingsmiljø

Det program, der styrer Makey-Makey'en, er skrevet i IDE'en Microsoft Visual Studio Code. VS Code er en editor, der kan anvendes i forbindelse med mange forskellige programmeringssprog og digitale værktøjer/miljøer. Programmet har en lang række brugbare funktioner, og samtidig findes der et hav af extensions - lavet af fællesskabet.

Programmet er skrevet i Python, som er et interaktivt, dynamisk programmeringssprog, der tillader struktureret, object-oriented og funktionel programmering. Pythons designfilosofi kan opsummeres med de følgende udsagn, der stammer fra dokumentet "The Zen of Python"⁵:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

Python er altså et alsidigt programmeringssprog, der er forholdsvist let at lære og bruge, sammenlignet med andre sprog. Dets designfilosofi fokuserer på læsbarhed, enkelthed og klarhed, og det er i høj grad denne filosofi, der har været med til at gøre det så populært, som det er i dag. Kombineret med understøttelsen af struktureret, OOP (objektorienteret programmering) og funktionel programmering, betyder dette, at Python er et kraftfuldt værktøj til udvikling af forskellige programmer - især når der er tale om mindre erfarne udviklere.

⁵ Peters, Tim (2004)

Beskrivelse af relevante dele af koden

Her kommer en beskrivelse af de relevante dele af programmets kode. Der vises billeder af den omtalte kode, som gennemgås linje for linje.

Herunder ses programmets bærende funktion, "listen_for_keypress()", som styrer lydafspilleren:

```
31 def listen_for_keypress():
32     print("Press a key to play a sound: ")
33     skip = False
34     while True:
35         if not isTimeoutPassed():
36             time.sleep(1)
37             continue
38
39         if skip:
40             skip = False
41             keyboard.read_key()
42             continue
43         else:
44             skip = True;
45
46         keyboard.read_key()
47         sound = random.choice(os.listdir("Sounds"))
48         thread = threading.Thread(target=play_sound, args=('Sounds\\' + sound,))
49         thread.start()
50         setNewTimeout()
51
52     listen_for_keypress()
```

På linje 31 defineres funktionen "listen_for_keypress()", som strækker sig til linje 50. "

Til at starte med (linje 32), skriver programmet "Press a key to play a sound:" på skærmen. Dette er primært implementeret for lettere at kunne identificere funktionen i debugging-processen, og vil ikke være synligt for brugeren.

På linje 33 defineres en ny primitiv variabel, "skip", som sættes til *false*. Denne skal bruges til at kunne springe hver anden tastaturhændelse over, senere i programmet. Grunden til, at hver anden tastaturhændelse skal springes over er, at programmet registrerer alle tastaturhændelser. Her er der både tale om key-down og key-up, så hvis koden ikke sprang hver anden hændelse over, ville resultatet være, at der blev afspillet to lyde, hver gang man trykkede på knappen én gang.

Stort set hele funktionen køre i et while-loop, som startes på linje 34.

På linje 35 tjekker et if-statement, om programmets timeout, der er defineret tidligere i koden, er slut. Hvis timeouten ikke er slut, udskydes kodeeksekveringen med 1 sekund (linje 36), og while-loopet brydes på linje 37.

På linje 39 til 44 sørger et if/else-statement for, at hver anden tastaturhændelse springes over. Dette gøres med udgangspunkt i variabelen "skip", som blev defineret på linje 33. Ved første eksekvering af programmet, køres koden under "else:", hvor "skip" sættes til *true*. Næste gang koden køres, vil koden under "if:" køres, fordi "skip" nu har værdien *true*. Her sættes "skip" igen til *false*, og aflæser tastaturberøringen i funktionen "keyboard.read_key()" (linje 41). Herefter fortsætte loopet igen.

På linje 47 vælger programmet en tilfældig lyd i mappen "Sounds", og på linje 48 afspilles denne lyd. Hver gang en lyd afspilles, sætter programmet en ny timeout med funktionen "setNewTimeout()" på linje 50.

Denne kode fungerer med Makey-Makey'en, fordi den emulerer et keypress på computerens tastatur. Med andre ord narrer den computeren til at tro, at et menneske har trykket på tastaturet, og den derfor skal spille en lyd.

Funktionen "listen_for_keypress()" køres på linje 52. Grunden til, at den ikke kører direkte er, at den blot defineres på linje 31 og altså skal kaldes et andet sted i programmet for at blive kørt.

Prototypen i aktion

Da udformningen af pressure-plate og program var slut, kunne prototypen testes. Formålet med denne test var at undersøge, om produktet fungerede, som det skulle, og hvordan eventuelle fejl kunne løses. Under testen har vi ikke brugt de rigtige lydfiler, og mappen "Sounds" er bare fyldt med en masse tilfældige lyde. I en rigtig implementering af produktet, vil disse lyde være erstattede af dystre og uhyggelige lyde.

Testen forløb ved, at vi placerede pressure-platen på gulvet og tilsluttede den til Makey-Makey'en, som blev tilsluttet til computeren. Herefter gik et af gruppens medlemmer henover den for at undersøge, om den aktiverede og spillede lydene korrekt.

Vi sørgede for at dokumentere testen i videoformat. Vi optog i alt to videoer, som kan findes her:

https://www.youtube.com/watch?v=tJ_N_K_kbjw
<https://www.youtube.com/watch?v=AlUuqfbgKRw>

Da vi havde testet funktionaliteten af prototypen, var vi overordnet set tilfredse med resultatet. Det er dog tydeligt, at programmets timeout skal sættes op, så der ikke spilles en masse lyde oveni hinanden.

Samtidig kunne vi konkludere, at prototypen hurtigt stoppede med at virke i kælderen. Mere herom i afsnittet *Mulige forbedringer*.

Konklusion

Diskussion af implementering ift. design og kravspecifikation

Projektets formål var at udvikle og implementere en løsning, vha. en Makey-Makey, som skulle kunne tilføje en funktion og mere aktivitet til et sted på skolen, der førhen var uden funktion, tomt og kedeligt.

Produktet skulle bestå af en installation, som gør brug af en Makey-Makey-mikrochip, og som er tilpasset en udvalgt lokation i fredagscaféen, hvor det kan tilføje karakter ved afspilning af lyd ifm. interaktion. Samtidig skulle produktet have et fysisk aspekt, som sørger for, at det går i et med omgivelserne.

Alle kravene er mere eller mindre opfyldt. Vi har udviklet en løsning, som er i stand til at udfylde tomheden på den valgte lokation, og samtidig skabe en form for historie i forbindelse med dette. Dog står det klart, at der er plads til en række forbedringer.

Mulige forbedringer

Under testen af prototypen, blev vi hurtigt opmærksomme på, at pressure-platen hurtigt stoppede med at fungere i fredagscaféen. Dette skyldes formentlig, at der i fredagscaféen - selvom der er tale om en kælder - er lavere luftfugtighed end i undervisningslokalet, hvor prototypen er konstrueret. Derfor var systemet ineffektivt på den valgte lokation, fordi det ikke var i stand til at lede elektronerne i el-kredsløbet.

Prototypen var funktionel i undervisningslokalet, hvor den blev udviklet. Grunden til dette kunne være, at der i dette lokale er højere luftfugtighed som resultat af udånding og dårlig udluftning.

I en eventuel videreudvikling af prototypen, er det vigtigt at tage højde for lokationens miljømæssige faktorer, såsom luftfugtighed, temperatur m.v. Havde man taget højde for disse, i starten af projektet, kunne man sandsynligvis udvikle et mere effektivt produkt og opnå et bedre slutresultat.

Alt i alt har projektet været en succes, og overordnet set er produktet blevet tilfredsstillende. Vi har lært en masse om Makey-Makeys, og vi har udviklet vores evner indenfor problemløsning og teknisk implementering de løsninger, der udvikles.

Litteraturliste

DatasheetsPDF (u.d.): "*GPCE4096UA Controller Datasheet PDF*", datasheetspdf.com.

Udgivet u.d., set 30/1-2023:

<https://datasheetspdf.com/datasheet/GPCE4096UA.html>

Københavns Universitet (u.d.): "*Design-tænkning*", Københavns Universitet, ku.dk. Udgivet

u.d., set 6/3-2023:

<https://innovation.sites.ku.dk/model/design-thinking/>

Microchip (u.d.): "*ATmega32U4*", microchip.com. Udgivet u.d., set 30/1-2023:

<https://www.microchip.com/en-us/product/ATmega32u4#>

Peters, Tim (2004): "*The Zen of Python*", python.org. Udgivet 19/8-2004, set 6/2-2023:

<https://peps.python.org/pep-0020/>

Robotnørderiet (u.d.): "*MAKEY MAKEY*", sites.google.com/a/pha.dk/robotnoerderiet.

Udgivet u.d., set. 30/1-2023:

<https://sites.google.com/a/pha.dk/robotnoerderiet/kreds!%C3%B8b/makeymakey?pli=1>