

A stylized, cartoonish illustration of Goofy from Disney. He is shown from the chest up, wearing his signature orange shirt and black tie. His large, white, floppy ears are prominent, and he has a wide, open-mouthed smile showing his tongue and teeth. His eyes are squeezed shut in a happy expression. The background is a soft, light green gradient.

### 3. Semesterprojekt - Goofy Candy Gun Gruppe 3

Rieder, Kasper  
201310514

Jensen, Daniel V.  
201500152

Nielsen, Mikkel  
201402530

Kjeldgaard, Pernille L.  
PK94398

Konstmann, Mia  
201500157

Kloock, Michael  
201370537

Rasmussen, Tenna  
201406382

22. april 2016

# Indhold

<b>Indhold</b>	<b>ii</b>
<b>Figurer</b>	<b>iv</b>
<b>1 Resumé</b>	<b>1</b>
<b>2 Abstrakt</b>	<b>1</b>
<b>3 Indledning</b>	<b>2</b>
<b>4 Termliste</b>	<b>3</b>
<b>5 Projektformulering</b>	<b>4</b>
5.1 Indledning . . . . .	4
5.2 Rigt Billede . . . . .	5
5.3 MoSCoW . . . . .	5
5.4 Opdeling af gruppen . . . . .	6
<b>6 Krav</b>	<b>7</b>
6.1 Aktørbeskrivelse . . . . .	7
6.2 Use case beskrivelse . . . . .	7
6.3 Ikke-funktionelle krav . . . . .	7
<b>7 Projektafgrænsning</b>	<b>9</b>
<b>8 Metode</b>	<b>10</b>
8.1 SysML . . . . .	10
8.2 Software Design Principper . . . . .	10
8.3 Designmetode . . . . .	10
<b>9 Systembeskrivelse</b>	<b>11</b>
<b>10 Systemarkitektur</b>	<b>12</b>
10.1 Signalbeskrivelse . . . . .	12
<b>11 Design og Implementering</b>	<b>13</b>
11.1 Valg og Begrundelse . . . . .	13
<b>12 Udviklingsværktøjer</b>	<b>14</b>
12.1 PSoC . . . . .	14
12.2 DevKit 8000 . . . . .	14
<b>13 Resultater og Diskussion</b>	<b>15</b>
13.1 Perspektivering . . . . .	15
13.2 Perspektivering til semesterets kurser . . . . .	15
13.3 Ingeniørfaglige Styrker og Svagheder . . . . .	15
<b>14 Fremtidigt Arbejde</b>	<b>16</b>
<b>15 Fejl og Mangler</b>	<b>17</b>

<i>INDHOLD</i>	iii
<b>16 Referencer</b>	<b>18</b>
<b>Litteratur</b>	<b>18</b>

## Figurer

1	Rigt Billede af det endelige produkt . . . . .	5
2	Illustration af Goofy Candy 3000 overordnet struktur . . . . .	11

**1   Resumé**

**2   Abstrakt**

### **3 Indledning**

## **4 Termliste**

## 5 Projektformulering

### 5.1 Indledning

Ønsket med dette projekt er at udvikle en mekanisk enhed der kan styres med en håndholdt controller. Med dette udgangspunkt blev forskellige muligheder undersøgt som inspirationskilde, hvor valget herefter faldt på en kanon til affyring af slik. Ideen med en kanon der affyrer slik er ikke original, som det kan ses ud fra projektet "*The Candy Cannon*" som er fundet på YouTube: <https://www.youtube.com/watch?v=VgZhQJQnnqA>. Til forskel fra *The Candy Cannon* og lignende projekter som affyrer projektiler uden et egentlig formål, vil der i dette projekt blive udviklet en kanon til brug i et spil. Kanonen affyres og styres af spillerne via en controller. Altså skal projektet ende med en kanon som indgår i et to personersspil, f.eks. til brug ved fester og andre sociale begivenheder.

I dette projektet skal der altså udvikles en slikkanon til spillet *Goofy Candygun 3000*. Denne slikkanon skal kunne skyde med slik. Dette kunne for eksempel være M&M's eller Skittle's.

Goofy Candygun 3000 er et spil til to personer, hvilket gør det velegnet i sociale sammenhænge. Spillet går ud på at opnå flest point ved at ramme et mål. Hver spiller får et bestemt antal skud. Efter skuddene er opbrugt, er vinderen spilleren med flest point.

Et typisk brugerscenarie er, at spillerne bestemmer antallet af skud for runden. Når dette er gjort, er spillet igang. Herefter går Wii-nunchucken på skift mellem spillerne for hvert skud. Dette fortsættes indtil skuddene er opbrugt. Vinderen er spilleren med flest point. Spillets statistikker vises løbende på brugergrænsefladen.

Det endelige produkt omfatter:

- En brugergrænseflade, hvor spilstatistikker fremvises til deltagerne. Dette er blandt andet:
  - Pointvisning
  - Kanonens vinkel
  - Antal resterende skud
- En motor, der drejer kanonen om forskellige akser
  - Dette styres med en Wii-nunchuck
- Et mål, der kan registrere spillernes skud

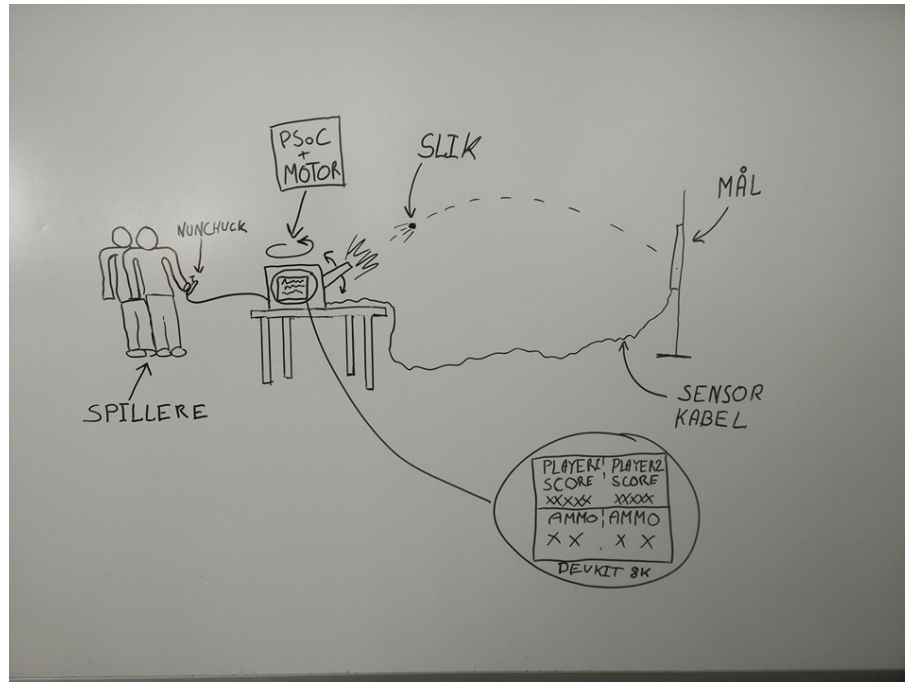
Med baggrund i krav stillet af organisationen IHA, bliver produktet udviklet med følgende funktioner:

- DevKit 8000 som den indlejrede linux-plattform til spillets grafiske brugergrænseflade.
- Motorer til styring af kanonen.
- Sensorer.
- PSoC 4, anvendt til styring af motorer og kommunikation med sensorer.



## 5.2 Rigt Billede

På figur 1 ses et rigt billede af det ønskede produkt. Billedet beskriver bruger-scenariet.



Figur 1: Rigt Billede af det endelige produkt

## 5.3 MoSCoW

I forbindelse med projektet gøres der brug af MoSCoW-princippet (<https://en.wikipedia.org/wiki/MoSCoW>) for at prioritere hvilke krav, der skal være implementeret ved projektets afslutning. Ifølge MoSCoW er prioriteringerne 'Must have', 'Should have', 'Could have' og 'Won't have'. Kravene er, som følger:

- Produktet must have:
  - En motor til styring af kanonen
  - En grafisk brugergrænseflade til visning af statistikker
  - En Wii-nunchuck til styring af motoren
  - En kanon med en afskydningsmekanisme
- Produktet should have:
  - Et mål til registrering af point
  - En lokal ranglistestatistik

- Produktet could have:
  - Partymode-indstilling til over to spillere
  - Trådløs Wii-nunchuckstyring
  - Afspilning af lydeffekter
- Produktet won't have:
  - Et batteri til brug uden strømforsyning
  - Online ranglistestatistik

#### 5.4 Opdeling af gruppen

I løbet af projektet vil projektgruppen blive opdelt i to hovedgrupper - 'hardware' og 'software'. Softwaregruppen vil desuden stå for grænsefladeprogrammering mellem software og hardware. Disse grupper vil have til ansvar at designe og implementere hhv. hardware og software til projektet. Hardwaregruppen vil bestå af de personer, der læser til elektroingeniør (Mikkel Nielsen og Pernille Kjeldgaard). Softwaregruppen vil bestå af de personer, der læser til IKT-ingeniør (Kasper Rieder, Michael Kloock, Tenna Rasmussen, Mia Konstmann og Daniel Jensen).

## 6 Krav

Ud fra projektformuleringen er der formuleret en række krav til projektet. Disse indebærer to use cases og et antal ikke-funktionelle krav.

Indsæt figur her.

### 6.1 Aktørbeskrivelse

På figur 1 ses aktør-kontekst-diagrammet for systemet. På dette ses det, at der i systemet er en enkelt aktør og to use cases. Den ene aktør er brugeren som samtidig er den primære aktør. Brugeren kan sætte de to use cases i gang, hvilket indebærer, at kunne spille Goofy Candygun 3000 og sætte en test af systemet i gang.

### 6.2 Use case beskrivelse

#### 6.2.1 Use case 1

Dette er den rigtige use case. Det er denne, der køres, når der skal spilles et spil på Goofy Candygun 3000, og den initieres af brugeren. Det første der sker i use case er, at brugeren skal vælge, hvilken type spil han/hun gerne vil spille. Det betyder, at det skal bestemmes, om det skal være et enpersonsspil, topersonerspil eller om det skal være party mode. Herefter skal der vælges, hvor mange skud et spil skal vare og disse skal puttes i magasinet. Når dette er gjort kan spillet gå i gang. Brugeren indstiller kanonen med Wii-nunchuck og affyrer den. Herefter lader systemet et nyt skud og samme procedure gentages. Til slut vises information om spillet på brugergrænsefladen, brugeren afslutter spillet ved at trykke på knappen på brugergrænsefladen og denne vender tilbage til starttilstanden.

#### 6.2.2 Use case 2

Use case 2 skal kun bruges til at teste systemet og dets kommunikationsprotokoller. Den skal bruges til at finde ud af om systemet virker og hvis det ikke gør, hvad det så er, der er gået galt og hvor det er gået galt. Use casen initieres af brugeren, hvor der herefter bliver sendt informationer ud til de forskellige dele af systemet, som så gerne skal sende svar tilbage igen. Hvis det sker for alle dele, vil brugergrænsefladen til sidst meddele, at use casen er gennemført.

### 6.3 Ikke-funktionelle krav

De ikke-funktionelle krav siger noget om, hvordan systemet skal bygges, og hvilke specifikationer det skal have. I dette tilfælde er der udarbejdet syv krav, hvor der er et der siger noget om, hvordan kanonen skal kunne drejes. Nogle siger noget om kanonens størrelse, hvor der også er specificeret, hvor stort projektilet den skal affyre må være og et siger, hvor langt den skal kunne skyde. Der er et, der siger, hvor lang tid det må tage at skyde et projektil afsted og hvor hurtig den skal være til at lade kanonen igen. Endelig er der et, der specificerer, hvordan den grafiske brugergrænseflade skal se ud. Deciderede værdier for de enkelte krav kan læses i bilag.

»»»> 544e5ad4b9e1d0001444d4366a94b0ac885878d2

## 7 Projektafgrænsning

Ud fra MoSCoW-princippet er der udarbejdet en række krav efter prioriteringerne 'must have', 'should have', 'could have' og 'won't have'. Dette er for at gøre det tydeligt, hvad der er vigtigt, der bliver udviklet først, og hvad der godt kan vente til senere. Disse krav er som følger:

- Produktet must have:
  - En motor til styring af kanonen
  - En grafisk brugergrænseflade til visning af statistikker
  - En Wii-nunchuck til styring af motoren
  - En kanon med en affyringsmekanisme
- Produktet should have:
  - Et mål til registrering af point
  - En lokal ranglistestatistik
- Produktet could have:
  - En partymode-indstilling til over to spillere
  - Trådløs Wii-nunchuckstyring
  - Afspilning af lydeffekter
- Produktet won't have:
  - Et batteri til brug uden strømforsyning
  - Online ranglistestatistik

I forrige afsnit blev 'must have'-kravene beskrevet. Det er de krav, der har højst prioritering for at blive implementeret i projektet. Det vil altså sige, at kravene under punkterne 'should have' og 'could have' har lavere prioritet. For at kravene under punktet 'must have' er opfyldt, skal use case 1, Spil Goofy Candygun 3000 implementeres. Dette kan dog først lade sig gøre, når use case 2, Test Kommunikationsprotokoller er implementeret. Derfor blev prioriteringen i dette projekt, at use case 2 skulle implementeres til fulde, inden der kunne startes på at implementere use case 1. Det havde dog også høj prioritet at have et produkt, der kunne skyde, hvilket betød, at selvom affyringsmekanismen ikke indgår i use case 2, blev det alligevel prioriteret højt at få implementeret denne i systemet.

## 8 Metode

I arbejdet med projektet er det vigtigt at anvende gode analyse- og designmetoder. Dermed er det muligt at komme fra den indledende idé til det endelige produkt med lavere risiko for misforståelser og kommunikationsfejl undervejs. Det er også en stor fordel, hvis de metoder, der anvendes, er intuitive og har nogle fastlagte standarder. Det gør det muligt for udenforstående at sætte sig ind i, hvordan projektet er udviklet og designet. Dermed bliver projektet og dets produkt i højere grad uafhængigt af enkeltpersoner, og det bliver muligt at genskabe produktet.

I forbindelse med udviklingen af Goofy Candygun 3000 er der anvendt use cases til at beskrive brugen af produktet. Gennem analyse af use cases er det muligt at skabe overblik over de nødvendige funktioner, som produktet skal have for at opfylde projektformuleringen. Som modellerings sprog til diagrammer er der anvendt UML og SysML, som netop har fastlagte standarder, der gør de forskellige diagrammer lettere at sætte sig ind i, og som dermed bidrager til at give en god forståelse for opbygningen og designet af produktet.

### 8.1 SysML

### 8.2 Software Design Principper

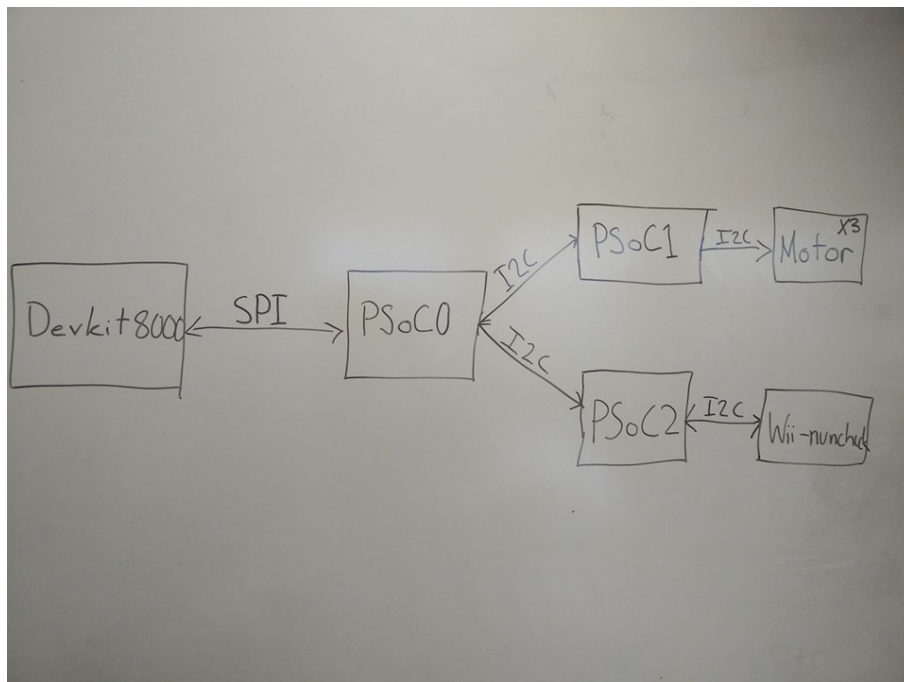
**\*\*Cohesion, coupling\*\***

### 8.3 Designmetode

**\*\*Afgivelser fra traditionelt vandfaldsmodel\*\***

## 9 Systembeskrivelse

Goofy Candygun 3000 er et underholdningssystem, som kan styres efter brugerønsker. Slikkanonen fungerer ved, at en bruger starter spillet på brugergrænsefladen på devkittet, derefter er spillet i gang. For at ændre sigteretningen for kanonen anvendes en Wii-nunchuck. Wii-nunchuck sender analogstikkets koordinater, via I2C kommunikation, til PSoC2. PSoC2 tolker dette data og sender kommandoen til PSoC0. PSoC0 videregiver denne kommando til PSoC1. PSoC1 tolker denne kommando og udsender et PWM signal, som bevæger motorene efter den sendte kommando. Når der affyres et projektil sendes data om et knap tryk på Wii-nunchuck fra PSoC2 til PSoC1 via PSoC0. På figur 2 ses den overordnede struktur af systemet.



Figur 2: Illustration af Goofy Candy 3000 overordnet struktur

HUSK BILLEDE OG FORKLARING AF ENDELIG UDSEENDE.

## 10 Systemarkitektur

### 10.1 Signalbeskrivelse

Blok-navn	Funktionsbeskrivelse	Signaler	Kommentar	3
Mål		Klarhed om projekt idé, samt de vigtigste krav bla bla bla bla	blabla	blabla
Længde		1 uge	3 uger	3 uger
Disciplin	Artefakt	Inception 1	...	...
<b>Projekt Formulering</b>	hey hey hey hey	Udled projektformulering. Anvend MosCoW til at bla bla bla		
<b>Specifikation</b>	Kravspecifikation & Accepttest	...	...	...
<b>Arkitektur</b>	Systemarkitektur	Ingen	bla bla	bla bla
<b>Design</b>	Design dokumentation	Ingen	bla bla	bla bla
<b>Implementering og modul test</b>	HW/SW/Mekanik	Ingen	bla bla	bla bla
<b>Projekt styring</b>	Projekt plan	bla bla	bla bla	bla bla

#### 10.1.1 Specifikation og Analyse



## **11 Design og Implementering**

### **11.1 Valg og Begrundelse**

## **12 Udviklingsværktøjer**

### **12.1 PSoC**

### **12.2 DevKit 8000**

## **13 Resultater og Diskussion**

### **13.1 Perspektivering**

### **13.2 Perspektivering til semesterets kurser**

### **13.3 Ingeniørfaglige Styrker og Svagheder**

**14 Fremtidigt Arbejde**

## **15 Fejl og Mangler**

**16 Referencer**

**Litteratur**