

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

Rieder, Kasper 201310514	Jensen, Daniel V. 201500152	Nielsen, Mikkell 201402530
Kjeldgaard, Pernille L. PK94398	Konstmann, Mia 201500157	Kloock, Michael 201370537
Rasmussen, Tenna 201406382		

1. marts 2016

# Indhold

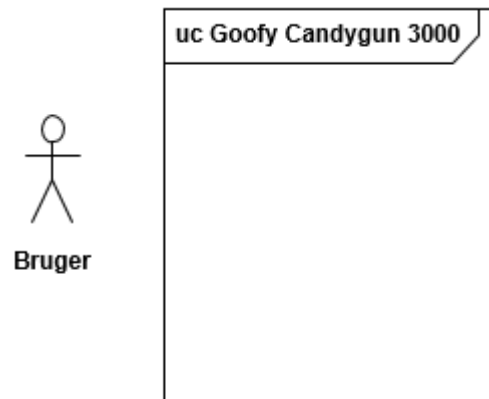
<b>Indhold</b>	<b>ii</b>
<b>Figurer</b>	<b>iii</b>
<b>1 Kravspecifikation</b>	<b>1</b>
1.1 Aktør kontekst diagram . . . . .	1
1.2 Use case diagram . . . . .	1
1.3 Aktør beskrivelse . . . . .	1
1.4 Fully dressed use case . . . . .	2
1.5 Fully dressed use case . . . . .	3
1.6 Ikke funktionelle krav . . . . .	4
<b>2 Accepttestspecifikation</b>	<b>6</b>
2.1 Use case 1 - Hovedscenarie . . . . .	6
2.2 Use case 2 - Hovedscenarie . . . . .	8
2.3 Ikke-funktionelle krav . . . . .	10
<b>3 Systemarkitektur</b>	<b>11</b>
3.1 Signalbeskrivelse . . . . .	11

## Figurer

1	Kontekst diagram for slikkanonen . . . . .	1
2	Use case diagram for slikkanonen . . . . .	1
3	Skitse af brugergrænsefladen . . . . .	5

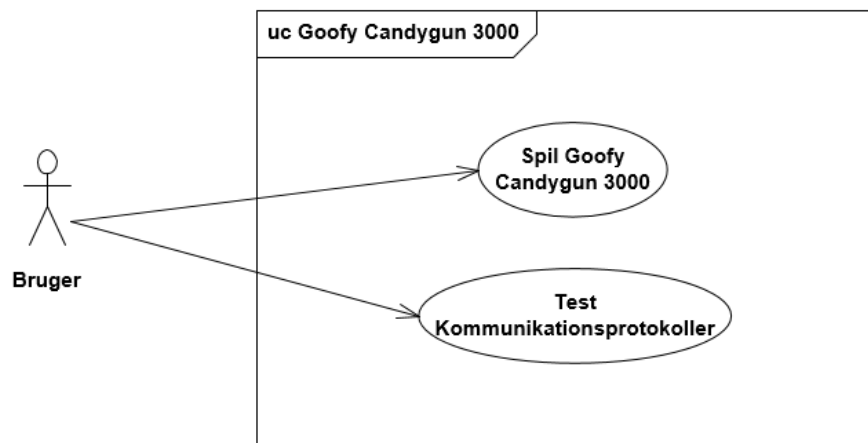
## 1 Kravspecifikation

### 1.1 Aktør kontekst diagram



Figur 1: Kontekst diagram for slikkanonen

### 1.2 Use case diagram



Figur 2: Use case diagram for slikkanonen

### 1.3 Aktør beskrivelse

I dette system er der en aktør, nemlig brugeren. Brugeren initierer systemet, ved at vælge spiltype på brugergrænsefladen. Derudover har brugeren mulighed for at stoppe spillet igennem brugergrænsefladen. Brugeren vil under spillet interagere med systemet gennem Wii-Nunchucken.

## 1.4 Fully dressed use case

<b>Navn</b>	Spil Goofy Candygun 3000
<b>Mål</b>	At spille spillet
<b>Initiering</b>	Bruger
<b>Aktører</b>	Bruger
<b>Antal samtidige forekomster</b>	Ingen
<b>Prækondition</b>	Spillet og kanonen er operationel. UC2 Test kommunikationsprotokoller er udført
<b>Postkondition</b>	Brugeren har færdiggjort spillet
<b>Hovedscenarie</b>	<ol style="list-style-type: none"> <li>1. Bruger vælger spiltype på brugergrænseflade</li> <li>2. Bruger vælger antal skud til runde</li> <li>3. Bruger fylder magasin med slik tilsvarende antal skud</li> <li>4. Bruger indstiller kanon med analogstick på Wii-nunchuck</li> <li>5. Bruger udløser kanonen med Wii-nunchucks trigger</li> <li>6. System lader et nyt skud</li> <li>7. Brugergrænseflade opdateres med spillets statistikker</li> <li>8. Punkt 4 til 7 gentages indtil skud er opbrugt <ul style="list-style-type: none"> <li>[Extension 1: Bruger vælger 2 player mode]</li> <li>[Extension 2: Bruger afslutter det igangværende spil]</li> </ul> </li> <li>9. Brugergrænseflade viser afslutningsinfo for runden</li> <li>10. Bruger afslutter runde</li> <li>11. Brugergrænseflade vender tilbage til starttilstand</li> </ol>
<b>Udvidelser/ undtagelser</b>	<p><b>[Extension 1: Brugeren vælger 2 player mode]</b></p> <ol style="list-style-type: none"> <li>1. Bruger overdrager Wii-nunchuck til den anden bruger</li> <li>2. Punkt 4 til 7 gentages indtil skud er opbrugt</li> <li>3. Use case genoptages fra punkt 8</li> </ol> <p><b>[Extension 2: Bruger afslutter det igangværende spil]</b></p> <ol style="list-style-type: none"> <li>1. Brugergrænseflade vender tilbage til starttilstand</li> <li>2. Use case afsluttes</li> </ol>

## 1.5 Fully dressed use case

<b>Navn</b>	Test kommunikationsprotokoller
<b>Mål</b>	At teste kommunikations protokoller
<b>Initiering</b>	Bruger
<b>Aktører</b>	Bruger
<b>Antal samtidige forekomster</b>	Ingen
<b>Prækondition</b>	Systemet er tændt
<b>Postkondition</b>	Systemet er gennemgået testen og resultaterne er vist
<b>Hovedscenarie</b>	<ol style="list-style-type: none"> <li>1. Bruger vælger test system på brugergrænseflade</li> <li>2. Devkit sender startbesked til PSoC0 via SPI</li> <li>3. PSoC0 sender acknowledge til Devkit via SPI [Exception 1: PSoC0 sender ikke acknowledge]</li> <li>4. Brugergrænseflade meddeler om gennemført SPI test</li> <li>5. PSoC0 sender startbesked til PSoC slaver via I2C</li> <li>6. PSoC slaver sender acknowledge til PSoC0 via I2C [Exception 2: PSoC slaver sender ikke acknowledge]</li> <li>7. PSoC0 meddeler om gennemført I2C test til Devkit via SPI</li> <li>8. Brugergrænseflade meddeler om gennemført I2C test</li> <li>9. Brugergrænseflade anmoder bruger om at trykke på knap 'Z' på Wii-nunchuck</li> <li>10. Wii-nunchuck sender besked "Knap Z trykket" til PSoC2 via I2C [Exception 3: Wii-nunchuck sender ikke "Knap Z trykket"]</li> <li>11. PSoC2 sender besked om "Knap Z trykket" til PSoC0 via I2C</li> <li>12. PSoC0 videregiver besked om "Knap Z trykket" til Devkit via SPI</li> <li>13. Brugergrænseflade meddeler om gennemført Wii-nunchuck test</li> <li>14. Brugergrænseflade meddeler at test af kommunikationsprotokoller er gennemført</li> </ol>

Udvidelser/ undtagelser	<p><b>[Exception 1: PSoC0 sender ikke acknowledge]</b></p> <ol style="list-style-type: none"> <li>1. Brugergrænseflade meddeler fejl i SPI kommunikation</li> <li>2. UC2 afsluttes</li> </ol> <p><b>[Exception 2: PSoC slaver sender ikke acknowledge]</b></p> <ol style="list-style-type: none"> <li>1. PSoC0 sender fejlmeddelse til Devkit</li> <li>2. Brugergrænseflade meddeler fejl i I2C kommunikation</li> <li>3. UC2 afsluttes</li> </ol> <p><b>[Exception 3: Wii-nunchuck sender ikke "Knap Z trykket"]</b></p> <ol style="list-style-type: none"> <li>1. PSoC2 sender fejlmeddelse til PSoC0</li> <li>2. PSoC0 videresender fejlmeddelse til Devkit</li> <li>3. Brugergrænseflade meddeler fejl i I2C kommunikation med Wii-nunchuck</li> <li>4. UC2 afsluttes</li> </ol>
-------------------------	--

## 1.6 Ikke funktionelle krav

1. Kanonen skal kunne drejes med en nøjagtighed på  $\pm 5^\circ$ 
  - 1.1. Vertikalt gælder dette for intervallet fra 0 til  $70^\circ$
  - 1.2. Horizontalt gælder dette for intervallet fra  $-45^\circ$  til  $45^\circ$
2. Kanonen skal kunne affyre projektiler med en diameter på  $1,25 \text{ cm} \pm 2 \text{ mm}$
3. Kanonen skal kunne affyre sit projektil minimum 1 meter
4. Kanonens størrelse må maksimalt være 40cm høj, bred og dyb
5. Fra aftryk på trigger til affyring må der maksimalt gå ti sekunder
6. Affyring af kanonen skal kunne afvikles minimum tre gange pr. minut
7. Figur 3 viser en skitse af hvordan den grafiskbrugergrænseflade kommer til at se ud



Figur 3: Skitse af brugergrænsefladen



## 2 Accepttestspecifikation

### 2.1 Use case 1 - Hovedscenarie

Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Vælg one-player mode.	Brugergrænsefladen viser spilside for one-player mode og anmoder om valg af antal skud.		
2	Vælg ti skud.	Brugergrænseflade anmoder om, at der fyldes ti stykker slik i magasinet.		
3	Fyld ti stykker slik i magasinet og tryk på knap for at starte spil.	Brugergrænseflade går til spilside og anmoder om, at kanon indstilles.		
4	Indstil kanon til affyring med Wii-nunchuck.	Kanon indstiller sig svarende til Wii-nunchucks placering.		
5	Udløs kanon med trigger på wii-nunchuck.	Kanon udløses.		
6	Gentag punkt 4 og 5 ti gange.	Punkt 4 og 5 gentages.		
7	Kig på brugergrænsefladen.	Brugergrænsefladen viser info om spillet.		
8	Tryk på knap for at vende tilbage til starttilstand.	Brugergrænseflade vender tilbage til startside.		

## 2.1.1 Use case 1 - Extension 1

Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Vælg two-player mode.	Brugergrænsefladen viser spilside for two-player mode og anmoder om valg af antal skud.		
2	Vælg ti skud på brugergrænseflade.	Brugergrænseflade anmoder om, at der fyldes ti stykker slik i magasinet.		
3	Fyld ti stykker slik i magasinet og tryk på knap for at starte spil.	Brugergrænseflade går til spilside og anmoder om, at kanon indstilles.		
4	Indstil kanon til affyring med Wii-nunchuck.	Kanon indstiller sig svarende til Wii-nunchucks placering.		
5	Udløs kanon med trigger på wii-nunchuck.	Kanon udløses.		
6	Giv Wii-nunchuck til den anden spiller.	Den anden spiller modtager Wii-nunchuck.		
7	Gentag punkt 4 til 6 indtil skud er opbrugt.	Punkt 4 til 6 gentages.		
8	Kig på brugergrænseflade.	Brugergrænseflade viser info om spil.		
9	Tryk på knap for at vende tilbage til starttilstand.	Brugergrænseflade vender tilbage til startside.		

**2.1.2 Use case 1 - Extension 2**

Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Vælg one-player mode.	Brugergrænsefladen viser spilside for one-player mode og anmoder om valg af antal skud.		
2	Vælg ti skud på brugergrænseflade.	Brugergrænseflade anmoder om, at der fyldes ti stykker slik i magasin.		
3	Fyld ti stykker slik i magasinet og tryk på knap for at starte spil.	Brugergrænseflade går til spilside og anmoder om, at kanon indstilles.		
4	Tryk på knap for afslutning af spil.	Brugergrænseflade vender tilbage til startside.		

**2.2 Use case 2 - Hovedscenarie**

Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Tryk start test på brugergrænseflade	Brugergrænsefladen udskriver at SPI og I2C testen er godkendt. Brugergrænsefladen anmoder bruger om tryk på Z på Wii-nunchuck		
2	Tryk Z på Wii-nunchuck	Brugergrænsefladen udskriver at Wii-testen er godkendt		

**2.2.1 Use case 2 - Exception 1**

Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Fjern SPI-kablet fra DevKittet.			
2	Tryk på start test på brugergrænseflade	Brugergrænsefladen udskriver SPI forbindelses fejlmeddelelse.		

**2.2.2 Use case 2 - Exception 2**

Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Fjern I2C-kabler fra alle I2C slaver.			
2	Tryk på start test på brugergrænseflade	Brugergrænsefladen udskriver I2C forbindelses fejlmeddelelse.		

**2.2.3 Use case 2 - Exception 3**

Step	Handling	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1	Disconnect Wii nunchuck fra systemet.			
2	Tryk på start test på brugergrænseflade			
3	Vent på timeout.	Brugergrænsefladen udskriver Wii Nunchuck forbindelses fejlmeddelelse		

### 2.3 Ikke-funktionelle krav

Krav	Test	Forventet observation/resultat	Faktisk observation/resultat	Vurdering (OK/FAIL)
1.1	Bruger styrer kanon fra "top"position til "bund"position, og måler vinkelforskellen.	Den afmålte vinkelforskel må være $70^{\circ} \pm 5^{\circ}$		
1.2	Bruger drejer kanonen fra længst til højre til længst til venstre og måler vinkelforskellen.	Den afmålte vinkelforskel ligger indenfor $70^{\circ} \pm 5^{\circ}$		
2	Et projektil på 1.25 cm i diameter $\pm 5$ mm affyres fra kanonen.	Projektilet bliver affyret		
3	Et projektil affyres, og distancen mellem kanonen og stedet hvor projektilet lander måles.	Distancen er blevet målt til at være større end 1 meter.		
4	Mål kanonens dimensioner med en lineal.	Dimensionerne overstiger ikke 40cm x 40cm x 40cm.		
5	Tryk på "triggeren" på Wii Nunchuck, og mål med et stopur hvor lang tid der går fra tryk, til kanonen bliver affyret.	Den målte tid er mindre end 10 sekunder.		
6	Kanonens affyres 3 gange, og et stopur startes ved første skud, og stoppes ved det tredje skud.	Den målte tid er mindre end 60 sekunder.		

### 3 Systemarkitektur

#### 3.1 Signalbeskrivelse

Blok-navn	Funktionsbeskrivelse	Signaler	Signalbeskrivelse
Devkit8000	Fungerer som grænseflade mellem bruger og systemet	masterSPI	Type: SPI
		touch	Type: touch
PSoC0	Master of I2C, SPI slave	slaveSPI	Type: SPI
		masterI2C	Type: I2C
Motorstyring	Doing stuff with things	motorSlave	Type: I2C
		pwm	Type: PWM
		power	Type: CMOS
Wii-nunchuck styring	stuuf	wiiSlave	Type: I2C
		nunchuckSlave	Type: I2C
PSoC1	dumdum	motorSlave	Type: I2C
Motor	movings	pwm	Type: PWM
		power	Type: CMOS
PSoC2	stuff	wiiSlave	Type: I2C
Wii-nunchuck	wiiiii	nunchuckSlave	Type: I2C
SPI	spiderman	MOSI	Type: SPI
		MISO	Type: SPI
		SCLK	Type: SPI
		SS	Type: SPI
I2C	jsjjjas	SDA	Type: I2C
		SCL	Type: I2C

##### 3.1.1 Specifikation og Analyse