

TIC TAC TOEMINATOR

SEMESTERPROJEKTGRUPPE: 8

Kravspecifikation

Studienummer:

1# 201609213
2# 201608892
3# 201607598
4# 201605348
5# 201605961
6# 201608798
7# 201609922
8# 201600313

Deltager:

Jakob Rune Skov
Jesper Mundbjerg Madsen
Mathias Kirkeby
Morten Dahl Nielsen
Morten Rahr Nielsen
Simon Fabricius Nielsen
Tobias Saaby Steffensen
Rune Bjørn Lassen

Vejleder:

Peter Høgh Mikkelsen

22.12.17

Indholdsfortegnelse

Kravspecifikation	3
1 Systembeskrivelse	3
1.1 Skitse af brugergrænseflade	4
1.2 Skitse af systemets fysiske design	5
2 MoSCoW	6
3 Funktionelle krav	7
3.1 Aktør beskrivelse	7
3.2 Use Case Diagram	8
3.3 Use Case Beskrivelse	8
3.4 Fully dressed Use Case beskrivelser	9
3.4.1 Use Case 1: Initier Spil	9
3.4.2 Use Case 2: Spil "Finite Mode"	10
3.4.3 Use Case 3: Spil "Infinity Mode"	11
4 Ikke-funktionelle krav	12
5 Udviklingsværktøjer	13
Referencer	14

Figurer

1 Udkast til systemets GUI	4
2 Skitse af armens fysiske design	5
3 Oversigt over pladelayout	5
4 Aktør-kontekst diagram	7
5 Use Case diagram	8

Tabeller

1 Aktør beskrivelse for aktøren bruger	7
2 Use Case beskrivelse UC1: Initier Spil	9
3 Use Case beskrivelse UC2: Spil "Finite Mode"	10
4 Use Case beskrivelse UC3: Spil "Infinity Mode"	11
5 Udviklingsværktøjer brugt til fremstilling af TTT	13

Ordforklaring

Forkortelse:	Forklaring:
AC	Actor Context Diagram
APSoC	A = Arm, PSoC-styring til arm
Arm	Robotarm
CAD	Computer Aided Design
EXC	Exception
EXT	Extension
FM	Finite Mode, spillemode til pladen fuld
GUI	Graphical User Interface
GUIrpi	RPi, hvorpå GUIprogram ligger
IDE	Integrated Development Environment
IM	Infinity Mode, spillemode med tre brikker til hver spiller
KB	Kryds og bolle
Master	En RPi som styrer systemet.
MoSCoW	Must have, Should have, Could have, and Won't have
PPSoC	P = Plade, PSoC-styring til spilleplade
PSoC 5LP	Programmable System-on-Chip
RPi	Raspberry Pi Zero W, single-board computer device
TTT	Tic Tac Toeminator
UC	Use Case Diagram
Wi-Fi	Trådløst netværk

Kravspekifikation

I dette dokument vil der blive gennemgået en systembeskrivelse af TTT. Systembeskrivelsen indebærer et udkast til GUI'en, samt en beskrivelse af de funktionelle og ikke-funktionelle krav. Systemets funktionelle krav er beskrevet ved tre opstillede Use Case's, som gennemgår TTT's funktionalitet.

1 Systembeskrivelse

TTT er et system, der har til formål at kunne underholde brugeren ved at spille det velkendte spil KB. Systemet er lavet til at kunne spille mod hvem som helst, både nye og rutinerede spillere. Spillet er designet med to forskellige game modes og tre forskellige sværhedsgrader, så alle spillere bliver udfordret på netop deres niveau. Brugergrænsefladen er designet intuitivt, som gør det nemt og overskueligt at indstille spillet efter brugerens behov.

Systemet kommer til at bestå af følgende dele:

- Robotarm, som styres af motorer, står for placeringen af systemets brikker
- Spilleplade, med indbyggede sensorer, til at registrere brugerens træk
- Gribemekanisme, der skal håndtere systemets spillebrikker
- Brugergrænseflade, hvorpå brugeren kan bestemme spillets indstillinger
- Strømforsyning, som forsyner systemet fra én stikkontakt
- Kasse, som danner base for de førnævnte mekaniske dele af systemet samt gemme bagvedliggende elektronik

Følgende elektriske og mekaniske dele vil blive styret af to PSoC 5LP og to RPi.

Det første brugeren møder er TTT's brugergrænseflade, som er en touchskærm. Touchskærmen styres af den ene Raspberry Pi (RPi), der behandler in- og outputs fra brugergrænsefladen. Denne RPi vil fremover blive benævnt som GUIrpi. GUIrpi'en kommunikerer gennem en Wi-Fi forbindelse med den anden RPi. Denne RPi vil herefter blive benævnt som Master og denne fungerer som systemets hjerne. Den beregner computerens træk og alt kommunikation vil centrerer herom. Masteren kommunikerer ligeledes med systemets to PSoC'er, som står for styringen af sensorerne i spillepladen, gribemekanismen og motorerne i robotarmen. PSoC'en, som står for styringen af sensorerne kaldes for PPSoC, og PSoC'en som styrer motorerne og gribemekanismen kaldes for APSoC. Robotarmen er udstyret med tre motorer, og disse styrer armens mulige bevægelser.

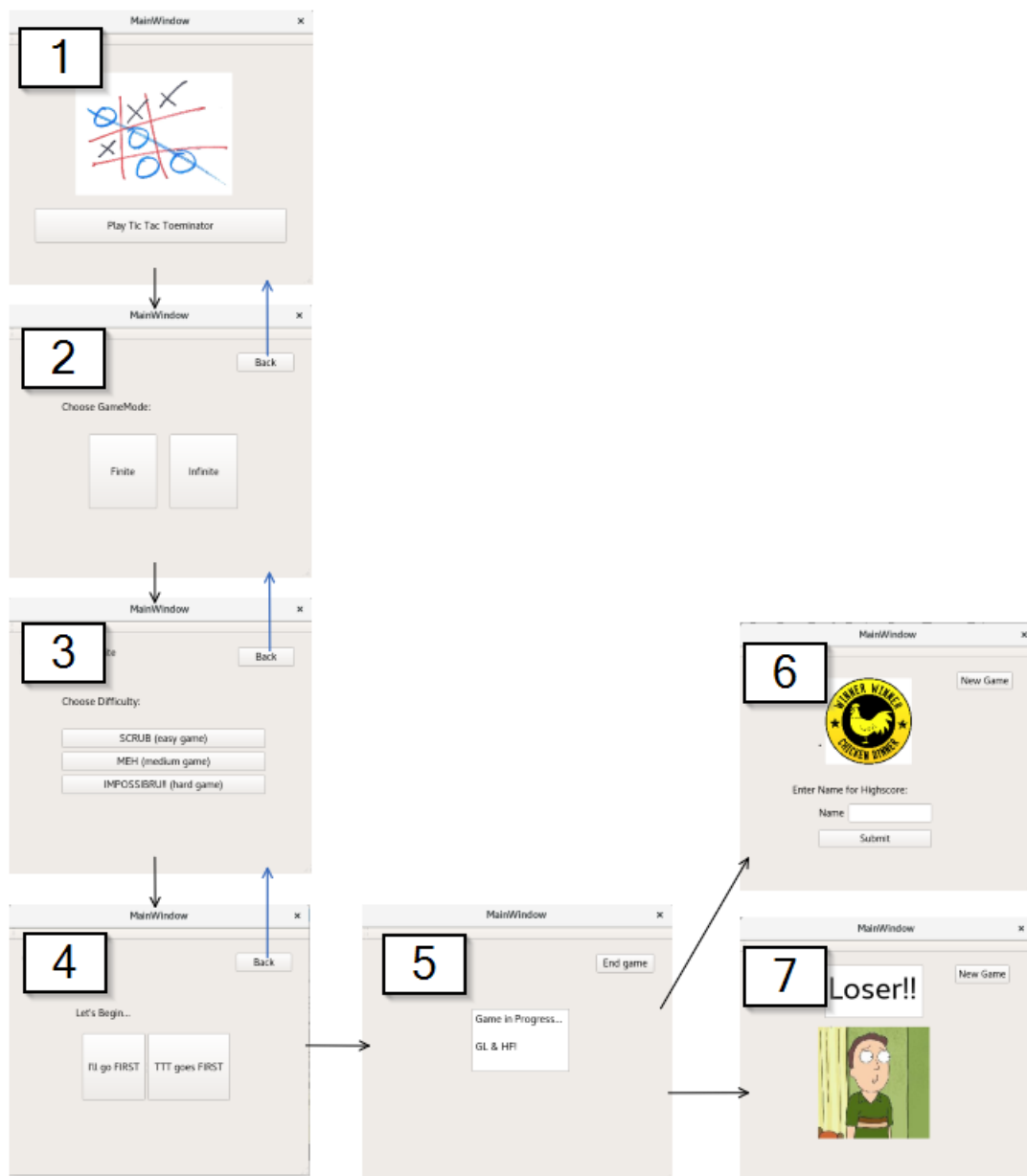
Strømforsyningen til systemet vil være gemt væk inden i kassen, så spillet ser indbydende og simpelt ud for brugeren.

Brugergrænsefladen bruges til at modtage data fra brugeren, som Masteren skal bruge til at starte spillet f.eks. valg af sværhedsgrad og spilstarter. De inputs, som systemet modtager fra brugeren, skal sendes fra GUIrpi'en til Masteren på en gang.

Systemet udvikles i udgangspunktet som en prototype for at demonstrere den overordnede funktionalitet, i det projektets rammer ikke rummer ressourcerne til at udvikle et endeligt kommercielt produkt.

1.1 Skitse af brugergrænseflade

Herunder vises en skitse af GUI'ens opbygning, se Figur 1. GUI'en består af syv vinduer, hvor de fire vinduer i første kolonne, skal indsamle og gemme info omkring brugerens valg. Først mødes brugeren af et startvindue, hvor brugeren skal trykke "Play Tic Tac Toeminator" for at starte spillet. Herefter møder brugeren vindue 2, hvor brugeren har mulighed for at vælge den ønskede game mode. Efterfølgende har brugeren mulighed for at vælge mellem de tre sværhedsgrader, Easy, Medium og Hard, i vindue 3. I vindue 4, i venstre kolonne, træffer brugeren det sidste valg, om hvorvidt han/hun ønsker at starte, eller om TTT skal starte. Spillet starter, og vinduet 5 vises med "Game in progress". De sidste to vinduer (vindue 6 og 7) i tredje kolonne skifter brugergrænsefladen til, alt efter om brugeren vinder, eller taber spillet mod TTT.

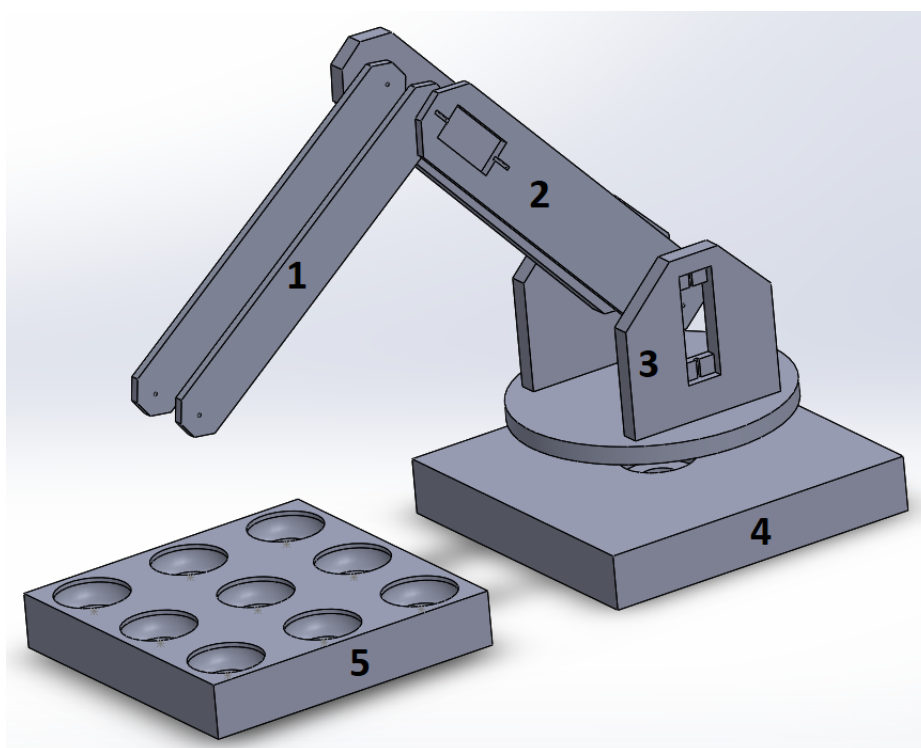


Figur 1: Udkast til systemets GUI

1.2 Skitse af systemets fysiske design

På Figur 2 ses en skitse af systemets fysiske design. Delene på skitsen er designet i CAD-værktøjet SolidWorks og kan findes i projektets bilag. Systemets delelementer navngives som følger:

1. Underarm
2. Overarm
3. Base
4. Anker
5. Spilleplade



Figur 2: Skitse af armens fysiske design

For at holde styr på spillepladens 9 pladser vil hver plads blive nummereret fra 0-8, som ses på Figur 3. Referencer til de forskellige pladser på pladen vil rette sig efter dette layout. Her vil robotarmen være placeret på samme side som pladserne 6, 7 og 8.

0	1	2
3	4	5
6	7	8

Figur 3: Oversigt over pladelayout

2 MoSCoW

For at fastslå kravene til systemet, er der lavet en overordnet MoSCoW analyse [1]. En MoSCoW analyse er et redskab til at opstille og give et konkret overblik over projektets krav i en prioriteret rækkefølge. Kravene bliver opdelt i de fire kategorier: *Must have*, *Should have*, *Could have* og *Won't have (this time)*.

Must have:

- Systemet **skal** have en brugergrænseflade, f.eks. til at indstille sværhedsgrad.
- Systemet **skal** kunne placere brikker på en spilleplade.
- Systemets software **skal** kunne beregne trækkene til at kunne spille KB.
- Systemets spilleplade **skal** involvere min. 1 sensor.

Should have:

- Systemet **bør** kunne spille KB i FM.
- Systemets brugergrænseflade og udformning **bør** kunne benyttes af en udefrakommende/ikke instrueret person.
- Systemets brugergrænseflade **bør** give muligheden for at vælge hvem der starter – eller kaste en virtuel mønt for at bestemme, hvem der starter.
- Systemets spændingsforsyning **bør** have en udformning så al forsyning gemmes væk bag ét stik, der uden videre kan sættes i stikkontakten.

Could have:

- Systemet **kan** spille KB i IM.
- Systemet **kan** være i stand til at fjerne brikkerne fra spillepladen efter endt spil.
- Systemet **kan** have et "skak-ur", der kan beslutte en vinder i IM.
- Systemets spændingsforsyning **kan** være en egenudviklet transformer og ensretter.

Won't have (this time):

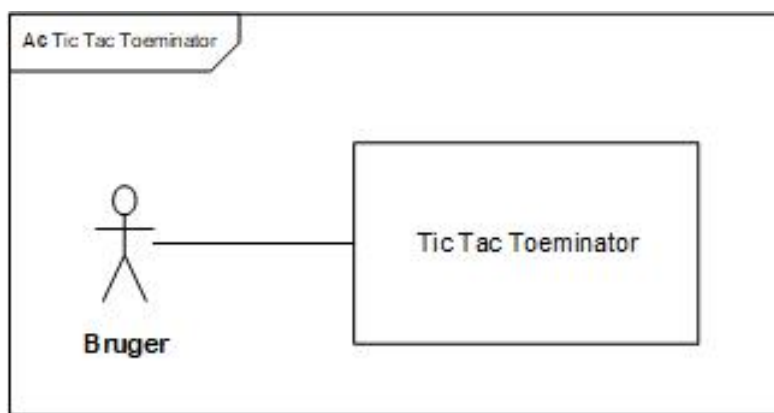
- Systemet **vil ikke** kunne kende forskel på kryds og bolle, blot hvem der placerer brikken (bruger eller system).
- Systemet **vil ikke** kunne spille andre brætspil end KB.
- Systemet **vil ikke** kunne udtrykke sig i fagter overfor modspilleren imens der spilles.

3 Funktionelle krav

I dette afsnit vil de funktionelle krav til systemet blive beskrevet. For at beskrive de funktionelle krav, er der først opstillet et Aktør-kontekst Diagram for at identificere systemets aktører. Herefter kan systemets funktionalitet beskrives ved et Use Case diagram, som viser systemets Use Cases's. For hver Use Case er der lavet fully dressed Use Case beskrivelser, som går ned i detaljer og beskriver systemets funktionalitet.

3.1 Aktør beskrivelse

Til at beskrive systemet er der blevet opstillet et Aktør-kontekst Diagram. På Figur 4 vises den ene aktør der interagerer med systemet, brugeren. På dette niveau vises TTT som en Blackbox, som brugeren kan interagere med.



Figur 4: Aktør-kontekst diagram

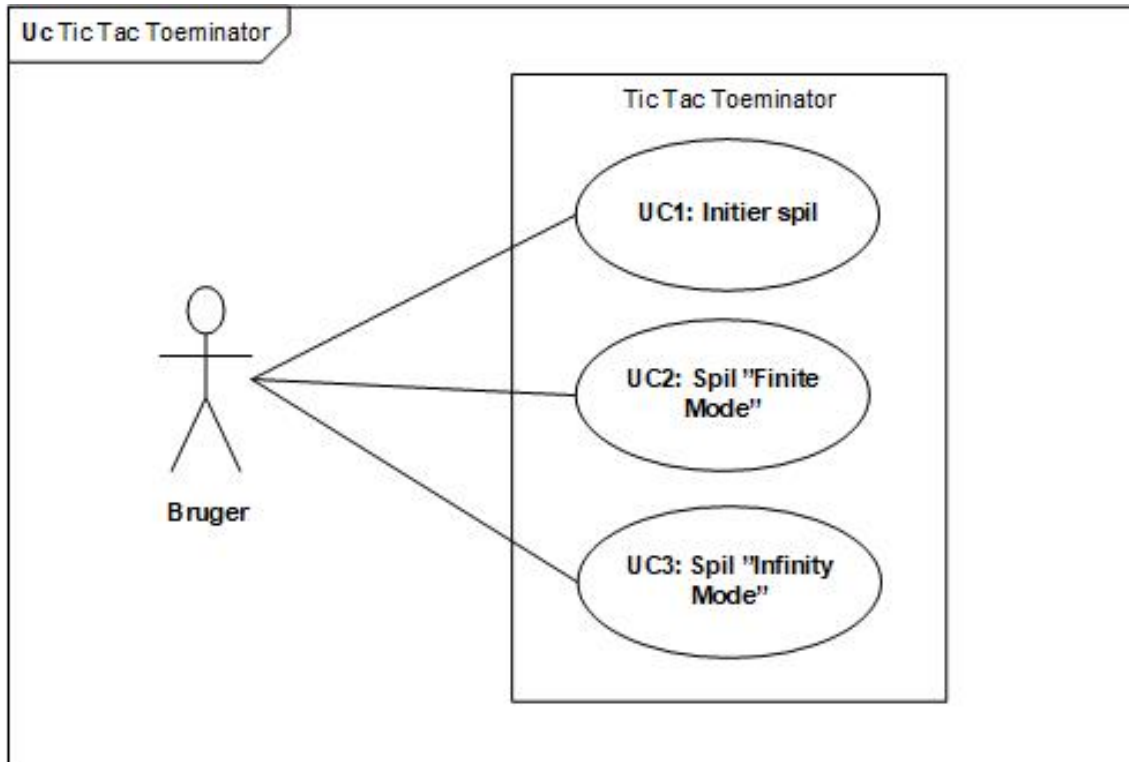
Herunder følger en beskrivelse af brugeren.

Aktør navn:	Bruger
Alternativ reference:	Spiller
Type:	Primær
Beskrivelse:	Brugeren kan interagere med systemet enten gennem brugergrænsefladen, hvorigennem spillemode, sværhedsgrad og spilstarter skal indstilles, eller ved at flytte på en spillebrik.

Tabel 1: Aktør beskrivelse for aktøren bruger

3.2 Use Case Diagram

På Figur 5 ses Use Case diagrammet for systemet. Der er i alt tre Use Case's, som hver især beskriver en funktionalitet systemet tilbyder brugeren. Her vises TTT som en Whitebox, da det her er muligt at se funktionaliteten inde i blokken.



Figur 5: Use Case diagram

3.3 Use Case Beskrivelse

UC1 Initier Spil: For at starte et spil skal systemet initieres. Det indbefatter at brikkerne skal placeres korrekt og, at brugeren har valgt de ønskede indstillinger for spillet.

UC2 Spil "Finite Mode": Brugeren har valgt at spille i "Finite Mode", hvilket betyder, at der spilles indtil der er opnået tre på stribe eller til pladen er fyldt ud. Brugerens resultat (Vundet, Tabt eller Uafgjort) vil blive vist på GUI'en.

UC3 Spil "Infinity Mode": Brugeren har valgt at spille i "Infinity Mode", hvilket betyder, at spillet fortsætter indtil tre på stribe er opnået. Brugerens resultat (Vundet eller Tabt) vil igen blive vist på GUI'en.

3.4 Fully dressed Use Case beskrivelser

I dette afsnit vil der blive gennemgået fully dressed Use Case beskrivelser for alle tre Use Case's på Figur 5.

3.4.1 Use Case 1: Initier Spil

Navn:	Initier Spil
Mål:	At initiere det ønskede spil, ved at vælge spillemode, sværhedsgrad og spilstarter
Initiering:	Bruger
Aktører:	Primær: Bruger
Prækondition:	Systemet er funktionsdygtigt og spillepladen er tom
Postkondition:	Systemet er initieret og klar til at spille mod en bruger
Hovedscenarie:	<ol style="list-style-type: none"> 1. Bruger tænder for systemet 2. Bruger vælger spillemode 3. Bruger vælger sværhedsgrad 4. Bruger vælger spilstarter 5. Bruger vælger "Start game" 6. Systemet vinker til brugeren
Udvidelser/ Undtagelser:	

Tabel 2: Use Case beskrivelse UC1: Initier Spil

Data variationsliste:

Spillemode: "Test Finite", "Finite", "Test Infinite", og "Infinite"

Sværhedsgrad: Easy, Medium og Hard

Spilstarter: Systemet eller brugeren

3.4.2 Use Case 2: Spil "Finite Mode"

Navn:	Spil "Finite Mode"
Mål:	At brugeren har gennemført et spil i "Finite Mode"
Initiering:	Bruger
Aktører:	Primær: Bruger
Prækondition:	Systemet er tilsluttet strøm. "Finite Mode" samt sværhedsgrad og spilstarter er valgt
Postkondition:	Brugeren har gennemført et spil i FM
Hovedscenarie:	<ol style="list-style-type: none"> 1. Bruger placerer sit træk [EXT1: Systemet er spilstarter og dette er første træk] 2. System registrerer brugerens træk 3. System beregner sit træk ud fra sværhedsgrad [EXC1: System registrerer tre på stribe (Bruger)] [EXC2: Systemet registrerer, at pladen er fyldt] 4. System placerer sit træk 5. Gå til punkt 1. [EXC2: Systemet registrerer, at pladen er fyldt] [EXC3: System registrerer tre på stribe (TTT)]
Udvidelser/ Undtagelser:	<p>[EXT1: Systemet er spilstarter og dette er første træk]</p> <ol style="list-style-type: none"> 1. Gå til punkt 3 <p>[EXC1: System registrerer tre på stribe (Bruger)]</p> <ol style="list-style-type: none"> 1. Systemet viser vindarskærm for FM <p>[EXC2: Systemet registrerer, at pladen er fyldt]</p> <ol style="list-style-type: none"> 1. Systemet viser uafgjortskærm <p>[EXC3: Systemet registrerer tre på stribe (TTT)]</p> <ol style="list-style-type: none"> 1. Systemet viser taberskærm for FM

Tabel 3: Use Case beskrivelse UC2: Spil "Finite Mode"

3.4.3 Use Case 3: Spil "Infinity Mode"

Navn:	Spil "Infinity Mode"
Mål:	At brugeren har gennemført et spil i "Infinity Mode"
Initiering:	Bruger
Aktører:	Primær: Bruger
Prækondition:	Systemet er tilsluttet strøm. "Infinity Mode" samt sværhedsgrad og spilstarter er valgt
Postkondition:	Brugeren har gennemført et spil i IM
Hovedscenarie:	<ol style="list-style-type: none"> 1. System beregner sit træk ud fra sværhedsgrad [EXT1: Brugeren er spilstarter og dette er første træk] 2. System placerer sit træk 3. Bruger placerer sit træk [EXT2: Seks brikker er registreret på spillepladen(Bruger)] 4. System registrerer brugerens træk 5. System beregner sit træk ud fra sværhedsgrad [EXT3: Seks brikker er registreret på spillepladen(TTT)] 6. Gå til punkt 2 7. System fjerner og placerer en brik 8. Bruger fjerner en brik [EXC1/EXC2: System har vundet på Easy/Medium/Hard] 9. Bruger placerer brik på tom plads 10. System registrerer brugerens træk 11. Gå til punkt 5. [EXC3: Bruger sat sin brik samme sted som han tog den fra] [EXC4: Bruger har vundet]
Udvidelser/ Undtagelser:	<p>[EXT1: Brugeren er spilstarter og dette er første træk]</p> <ol style="list-style-type: none"> 1. Gå til punkt 3 <p>[EXT2: Seks brikker er registreret på spillepladen (Bruker)]</p> <ol style="list-style-type: none"> 1. Gå til punkt 9 <p>[EXT3: Seks brikker er registreret på spillepladen(TTT)]</p> <ol style="list-style-type: none"> 1. Gå til punkt 7. <p>[EXC1: System har vundet på Easy/Medium]</p> <ol style="list-style-type: none"> 1. Systemet viser taberskærm for IM <p>[EXC2: System har vundet på Hard]</p> <ol style="list-style-type: none"> 1. Systemet viser taberskærm for IM 2. Bruger indtaster navn 3. System gemmer navn og score på topscorerlisten <p>[EXC3: Bruger sætter sin brik samme sted som han tog den fra]</p> <ol style="list-style-type: none"> 1. Systemet viser fejlbesked og viser "Der er opstået en kritisk situation" <p>[EXC4: Bruger har vundet]</p> <ol style="list-style-type: none"> 1. System viser vindarskærm for IM 2. Bruger indtaster navn 3. System gemmer navn og score på topscorerlisten

Tabel 4: Use Case beskrivelse UC3: Spil "Infinity Mode"

4 Ikke-funktionelle krav

I dette afsnit vil de ikke-funktionelle krav til systemet være beskrevet. De ikke-funktionelle krav er delt op i kategorierne Fysiske dimensioner, GUI og Andet.

De fysiske mål skrives i denne rækkefølge (medmindre andet er angivet): Bredde x Længde x Højde, og alle fysiske dimensioner er angivet i mm. Alle \pm mål er for hver af Bredde x Længde x Højde værdierne, ikke for det samlede areal/volume.

Fysiske dimensioner:

1. Spillepladens dimensioner skal være: 170 x 170 x 40 (± 10)
2. Kassens dimensioner skal være: 400 x 500 x 200 (± 25)
3. Spillebrikkernes dimensioner: 15 i diameter (± 5)
4. Robotarmens overarm skal have dimensionerne: 50 x 220 x 70 (± 10)
5. Robotarmens underarm skal have dimensionerne: 30 x 220 x 40 (± 10)

GUI

1. Reaktionstiden mellem hvert vindue skal være under 1 sekund
2. Opdatering af topscorer på databasen skal ske på under 1 sekund
3. Varigheden for initiering af spillet skal være maksimum 20 sekunder
4. Opløsningen af touchskærmen skal minimum være 400x240

Andet

1. Reaktionstiden for beregning af træk skal være under 2 sekund
2. Robotarmen skal kunne placere en brik på under 10 sekunder
3. Spillebrikkerne skal veje mellem 5 og 20 gram
4. Det samlede system må maksimalt veje 10 kilogram

5 Udviklingsværktøjer

I forbindelse med fremstillingen af TTT systemet, er der blevet brugt en del forskellige udviklingsværktøjer - disse værktøjer er her opstillet og kort beskrevet. (IDE = Integrated Development Environment)

Værktøj	Beskrivelse
Microsoft Visual Studio 2017	IDE til udvikling af C/C++ kode til algoritme
ShareL ^A T _E X	IDE til L ^A T _E Xskrivning
Microsoft Visio	IDE til diverse diagrammer
Multisim	IDE til kredsløbsdesign
Mathcad Prime 3.1	Matematik software
Sublime Text Editor	IDE til udvikling af netværk og database
DB Browser	Databasehåndteringsprogram
Qt Creator	IDE til udvikling af GUI
VMware	Sandbox platform
GIT	IDE til versionstyring
Redmine	Projekthåndteringsprogram
Nano	Command line tekstredigeringsprogram på RPi
GeoGebra	IDE til skitsering af geometri og algebra
PSoC Creator 4.1	IDE til udvikling af PSoC software
SolidWorks	CAD værktøj til udvikling af 3D modeller
Cura	Software til slicing af 3D modeller før print

Tabel 5: Udviklingsværktøjer brugt til fremstilling af TTT

Referencer

- [1] Wikipedia. *MoSCoW method*. 2016.
URL: <https://en.wikipedia.org/wiki/MoSCoW_method>. [8 december 2017]