

DDU Eksamensprojekt

Af Alexander Rossau, Teodor Stadler og Oliver Svanholm Kliim

Brainstorm

Underholdning, uddannelse / læring

Det kan være frustrerende at lave sit eget spil.

Det er svært/tager lang tid at lære at lave sit eget spil.

Det kan være dyrt at lave sit eget spil.

Nøgleproblem

Vi er kommet frem til nøgleproblemet: ***"Det tager lang tid at lære at lave sit eget spil."*** ud fra en brainstorm. Dette er dokumenteret ved hjælp af systematisk internetsøgning.

Problemanalyse

Årsager

Der er flere årsager til at det tager lang tid at lære at lave sit eget spil. Det kræver nemlig en masse kompetencer inden for en del områder, for at kunne lave et spil. En af de vigtigste er at lære at kode, da dette er essentielt for at få spillet til at virke, og kodning tager lang tid at lære. Det siges at det tager cirka 3-6 måneder at lære det generelle omkring kodning, hvor der i dette tidsinterval skal bruges 500-1000 timer. Dette kræver at man arbejder fokuseret i op til et halvt år, og derfor vil det ofte tage endnu længere for folk at lære at kode, da der vil opstå perioder med lav fritid eller manglede motivation.¹

Dette er dog kun med til at beskrive selve koden der sørger for spillets funktioner. Bagved det ligger der nemlig også en "game engine" og denne er meget sværere at lave. Dette er koden som spillet er bygget oven på, dvs. denne kode sørger de basale ting såsom hvordan objekter interagerer med hinanden. For at lave dette kræver det en stor mængde viden inden for matematik, især geometri, og algoritmer.²

Hvis man ikke selv vil udvikle sin egen "game engine" kan man også vælge at bruge eksisterende game engines, som er en af de mest populære game engines med over 2.5 millioner registrerede brugere³. Det tager dog også tid at lære Unity, da der er en masse funktioner til stede i programmet man skal kende til. Oven i tiden der skal bruges til at lære at kode, kan det tage op til 6 måneder før man har lært og blevet god til Unity.⁴

Alle disse ting ovenpå hinanden betyder derfor at der skal afsættes meget tid inden man kan begynde at arbejde på sit spil, hvilket gør at det kan tage meget lang tid før man har et færdigt produkt, hvis man ikke allerede har det forudsatte viden.

¹ Jalli, u.d.)

² (Blow, u.d.)

³ (arnia, u.d.)

⁴ (Samantaray, u.d.)

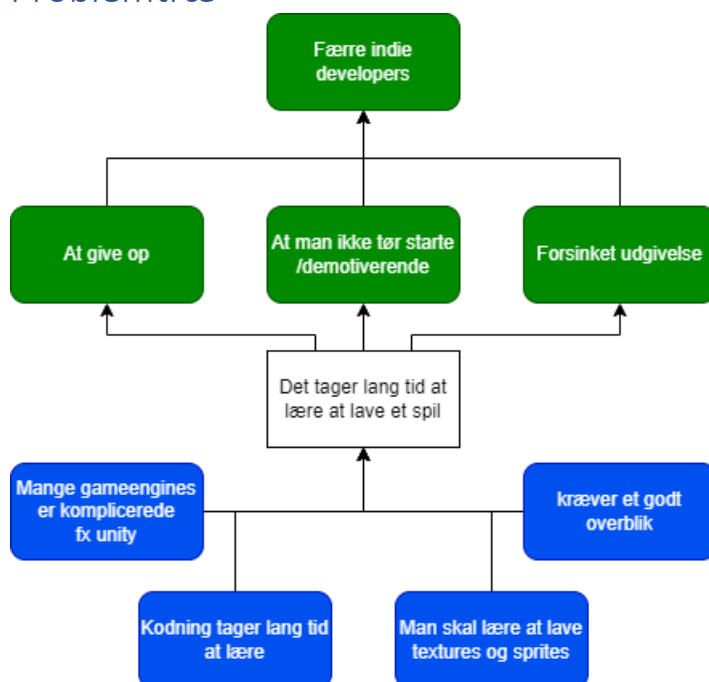
Konsekvenser

Spiludvikling er ikke en ting man bare lærer, og derefter laver præcist det spil man altid har drømt om at lave. Man vil altid blive bedre og lære noget nyt, mens man udvikler. Der er dog flere konsekvenser, som kommer af at bruge lang tid på læring. En af de mest grundlæggende, er manglende motivation, til at færdiggøre sig projekt. Hvis man stadig er relativt ny til spiludvikling, og ikke er blevet justeret til den lange proces som normalt er bag det, vil det være svært at finde motivation til faktisk at færdiggøre det. Inden man går i gang, ser man måske på en masse spil online, som ser rigtig pæne og simple ud, men så går man selv i gang, og ser at der er meget mere bag det. Den lange læringsproces skaber altså denne mur mellem hvad folk kan og vil opnå⁵.

Omfang

Hvert år bliver der udgivet cirka 6000⁶ indiespil på Steam, hvilket vil sige at der sandsynligvis er 6000+ personer årligt der har overkommet de udfordringer der er ved at udvikle sit eget videospil. Dette tal har været nogenlunde stabilt siden 2018⁷ og selv i 2020 under Corona nedlukningen hvor der ellers var mange mennesker der fandt nye hobbyer, skete der ikke nogen ændring i antallet af indiespil udgivelser på Steam. At der ikke har været en stigning i mængden af indiespil der udgives på Steam hvert år, kan være et tegn på at det er for svært at starte da man kunne forvente at se flere udgivelser i 2020 og efterfølgende år som følge af at mange mennesker pludselig havde meget mere tid på hænderne. At der ikke er en stigning, er bekymrende da dette kan betyde at der i fremtiden vil være et fald i indiespil udgivelser hvilket ville betyde at mange mennesker ville have givet op på deres drøm om spiludvikling da det er usandsynligt at der kommer til at være et fald i mængden af mennesker der er interesserede i spiludvikling.

Problemtræ



⁵ (Jennings, n.d.)

⁶ (xPaw, 2022)

⁷ (xPaw, 2022)

Årsager

Under årsager har vi placeret "Mange spiludviklingsprogrammer er komplicerede, for eksempel Unity". Dette er relevant på grund af hvordan User Interface på mange af disse programmer, er opbygget på måder hvor man først skal forstå den specielle struktur hver deres panel har.

Konsekvenser

Problemformulering

Det er et problem at mange spiludviklingsprogrammer er komplicerede at benytte fordi deres brugergrænseflade ofte tager meget lang tid at lære hvilket ovenpå det at man skal lære at programmere kan virke demotiverende for personer der ønsker at lave deres eget spil.

Produktprincip

Målgruppeanalyse

Der er forskellige typer personer som måske godt kunne tænke sig at lære at lave videospil. En indlysende en, er folk der gerne vil være spiludviklere. Det kan altså være nogle som har fundet interesse i at udvikle spil, eller selv har spillet spil, og nu gerne vil opsøge en karriere indenfor spiludvikling. Derudover kan forældre, som har børn der laver spil, også have interesse i at lære noget om hvordan det foregår. Det kunne for dem være nyttigt at lære de fremtidige jobmuligheder, for deres børn, indenfor gaming industrien. Udover bare generelt børn og deres forældre, er det også vigtigt at der er ordentlige værktøjer til lærere, for at de kan hjælpe studerende på den bedste måde til at lære spiludvikling. En sidste type person kunne være investorer som gerne vil starte et spiludviklingsfirma, eller investere i et, og derfor gerne vil forstå hvordan spiludviklingsprocessen faktisk foregår.

Konkurrentanalyse

Der er mange firmaer og individuelle mennesker der forsøger at gøre det lettere at benytte spiludviklingsværktøjer. Måden individuelle mennesker forsøger at gøre det lettere at benytte sig af spiludviklingsværktøjer er ofte ved at lave YouTube tutorials^{8 9 10} der forklarer hvordan et givent værktøj kan benyttes og hvordan man bedst kommer i gang med at lave et spil i det givne værktøj. Der er rigtig mange af disse værktøjer og de har hver i sær forskellige fokus i forhold til spiludvikling. Nogle eksempler på populære spiludviklingsværktøjer kunne være Unity, Roblox, og GameMaker.¹¹ Unity er et avanceret spiludviklingsværktøj der er i stand til at udføre mange forskellige funktioner på professionelt niveau,¹² Unity har dog den ulempe at det ikke er specielt begyndervenligt da

⁸ (Code Monkey, 2020)

⁹ (RoBuilder, 2022)

¹⁰ (Spalding, 2020)

¹¹

¹²

der er mange ting på brugerfladen hvilket gør det meget overvældende at se på hvis ikke man er vant til at bruge programmet. Desuden er det nødvendigt at kunne bruge c# til at kode sit spil i Unity, dette medvirker til at programmet er svært at bruge hvis man ikke har forudgående erfaring med programmering.¹³ Roblox er et eksempel på en spiludviklingsplatform der fokuserer på at gøre det lettere at lave et spil samt gøre det let at distribuere det og tjene penge på det spil man har lavet.¹⁴ Roblox Studio er navnet på det spiludviklingsværktøj der er tilknyttet til platformen. I Roblox Studio er der fokus på at gøre det let at lave komponenter i spilverdenen og skrive kode til dem. Roblox gør kodningen simplere end Unity ved at benytte sig af sproget Lua der er et Høj niveau sprog der er meget let at læse og intuitivt at skrive.¹⁵ Et andet eksempel på et spiludviklingsværktøj der forsøger at forsimple processen er GameMaker. GameMaker har en rimelig kompliceret brugerflade ligesom Unity, men til gengæld er det meget lettere at kode i da det benytter sig af sit eget programmeringssprog ved navn GML (GameMaker Language) der repræsenteres som en kombination af visuelle blokke og små kode strenge. Dette gør det meget let at få overblik over kode skrevet i GameMaker og betyder at det ikke er nødvendigt at skrive så meget kode som i andre spiludviklingsværktøjer.¹⁶ Ud af disse er Roblox og GameMaker vores hovedkonkurrenter der begge er spiludviklingsværktøjer der prøver at være simplere at benytte end Unity. Unity i sig selv er også en konkurrent fordi det er et spiludviklingsværktøj dog prøver det ikke at løse det samme problem som os men er tilgængelig en del af problemet, så hvis vores produkt ikke er simplere at benytte kan det ses som at være forfejlet. Til sidst er YouTube tutorials også en form for konkurrenter da de forsøger at gøre sådan at det ikke er nødvendigt at have et simplere værktøj, men derimod forsøger at gøre det lettere at bruge de mere komplicerede værktøjer.

Krav

For at lave et produkt skal vi have nogle krav til det. Vi er kommet frem til følgende krav:

Krav	Beskrivelse	Begrundelse
Nem at bruge	Det skal være nemmere at navigere og forstå User Interface i vores produkt end i andre konkurrenter såsom Unity.	Dette er vigtigt, som beskrevet i problemanalysen og problemtræet. Hvis ikke det er nemt at bruge, går det imod at løse vores problem om at det er for kompliceret at anvende og lære.
Optimeret	Programmet skal køre optimalt og ikke lagge brugerens computer. Det skal køre med mindst 30 FPS.	Hvis det kører dårligt, vil man ikke kunne bruge det, og vil derfor ikke ses som i hjælp i modsætning til andre engines.
Pænt design	Designet i programmet skal være pænt og enkelt.	Hvis det ser pænt ud, men stadig er enkelt, vil det

¹³

¹⁴

¹⁵

¹⁶

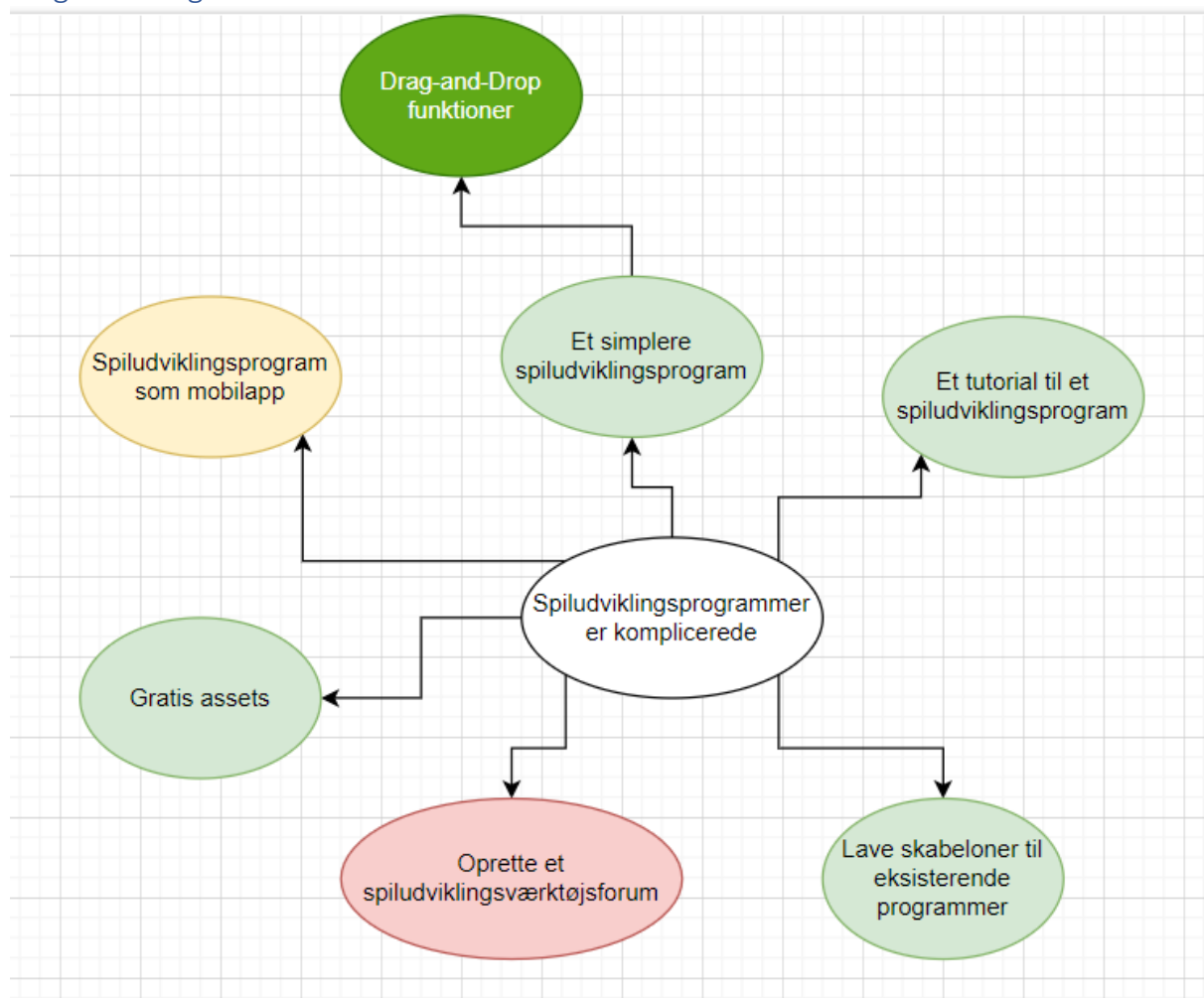
		passe til de fleste løsninger. ¹⁷
Billig	Løsningen skal være billig.	På den måde kan flest muligt kan få gavn af vores løsning der skal gøre det nemmere at lave 2D spil.
Man skal kunne lave mange unikke spil	Det skal være muligt at kunne lave en række spil der føles unikke at spille	Hvis alle de spil man kan lave føles ens så er der ikke særlig meget frihed i de spil man kan lave. Hvilket betyder at det er tættere på en levelcreator til et eksisterende spil end en gameengine.

Derefter skal kravene vægtes med en score fra 1-5. Et er mindst vigtigt og fem er mest vigtigt. Vi har givet disse vægte ud fra hvad vi mener er vigtigst for at finde den godt løsning

Krav	Vægt
Nem at bruge	5
Optimeret	4
Pænt design	3
Unikke spil	2
Billig	1

¹⁷ (Laning, 2021)

Idegenerering



Idesortering

Ikke nødvendig da der kun er fem ideer.

Løsningsforslag

Simplere Spiludviklingsprogram

En af løsningsmulighederne kunne være et simpere spiludviklingsprogram. Nuværende er de eksisterende spiludviklingsprogrammer opbygget med avancerede user interface og krav. Normalt skal man i disse programmer allerede vide meget om hvordan man programmerer og laver forskellige modeller til spil. Ved et simpere spiludviklingsprogram vil man nemt kunne lave et spil uden noget videre kode og uden modeller som man selv laver. Dette kunne for eksempel være et program hvor man bare kan vælge færdigkodet modeller fra en liste og indsætte dem i sit spil. Samtidig ville sådan en løsning skulle være nem for folk at lære at benytte, hvor folk senere hen kan opgradere til nogle mere avancerede spiludviklingsprogrammer.

Tutorial til eksisterende program

Tutorials til eksisterende programmer kan f.eks. være til Unity som har flere komplicerede funktioner. Her ville denne løsning kunne hjælpe folk som der starter ud, med at gennemgå en masse af de vigtige funktioner i programmet og illustrere dette med eksempler. Dette vil

have effekten at brugerne skulle bruge mindre tid på at lære programmet og derfor opfylde vores problemformulering. Disse tutorials vil være videoer som folk kan følge med i, og se gennem flere gange hvis der er noget som blev overset den første gang.

Skabeloner til eksisterende program

Denne løsning går ud på at lave nogle skabeloner for forskellige typer af spil til et eksisterende spiludviklingsprogram, sådan at folk let kan komme i gang med at lave den type spil de ønsker. En skabelon til en platformer kunne for eksempel indeholde en simpel bane med en spiller og en fjende nogle platforme og et mål. Eksempler på spiltyper som kunne være repræsenterede er: Platformer, shooter, beat em up, og racing.

Gratis Assets til eksisterende program

Dette løsningsforslag går ud på at lave nogle dele til et eksisterende program som andre ville kunne downloade og bruge. Disse dele ville have til formål at gøre det lettere for folk at komme i gang med at udvikle deres spil sådan at de ikke behøver at lave en masse ting som er grundlæggende for mange spil. Eksempler på dele som denne løsning kunne indeholde er: en spiller, en fjende, et checkpoint, et mål, og diverse terræn dele. Delene kunne eventuelt være en del af nogle pakker som folk kan downloade afhængigt af hvilken type spil de vil lave.

Spiludviklingsprogram som mobilapp

Denne løsning går ud på at udvikle en mobilapp til et spiludviklingsprogram, det kan enten være en app til et eksisterende program eller et helt nyt program. Formålet med løsningen er at gøre det muligt for folk at udvikle spil lige gyldigt hvor de er sådan at det er lettere at øve sig. Man ville for eksempel kunne øve sig på at udvikle spil i bussen på vej til skole.

PV-Skema

krav	vægtning	Simplere program		Tutorial		Skabeloner		Assets		Mobil app	
		point	v. point	point	v. point	point	v. point	point	v. point	point	v. point
Nem at bruge	5	9	45	9	45	9	45	9	45	3	15
Optimeret	4	9	36	9	36	3	12	3	12	9	36
Pænt design	3	9	27	3	9	3	9	3	9	1	3
Unikke spil	2	3	6	1	2	1	2	3	6	9	18
Billig	1	9	9	9	9	3	3	3	3	1	1
i alt		39	123	31	101	19	71	21	75	23	73

Vi vælger derfor at gå videre med idéen "Simplere program", da denne scorede højest i vores PV-skema.

Produktudformning

Beskrivelse af produktet

Et spiludviklingsprogram til 2d spil, hvor der er fokus på at have en simpel brugergrænseflade sådan at brugeren ikke bliver overvældet.

Kravspecifikationer

- Skal tage mindre end 6 måneder at lære. (mindre end Unity)
- Skal ikke lagge computere med følgende specs og bedre (5000 kr):
processor: i5-1035G1,
cpu: 1-1.19 GHz,
16 GB RAM,
64-bit OS.
- Skal være overskueligt at finde rundt i, og ikke overvældende med funktioner når det åbnes. (kan testes via pilot test senere)

Tekniske specifikationer

For at fremstille vores produkt skal vi bruge nogle tekniske værktøjer, som hjemmesider og software. Dette er følgende som vi vil anvende:

- Figma - Til design af User Interface
- Visual Studio Code - Til at udvikle vores kode i
- GitHub - Til at gemme vores kode online
- Processing i Java - Til at udvikle selve koden og programmet

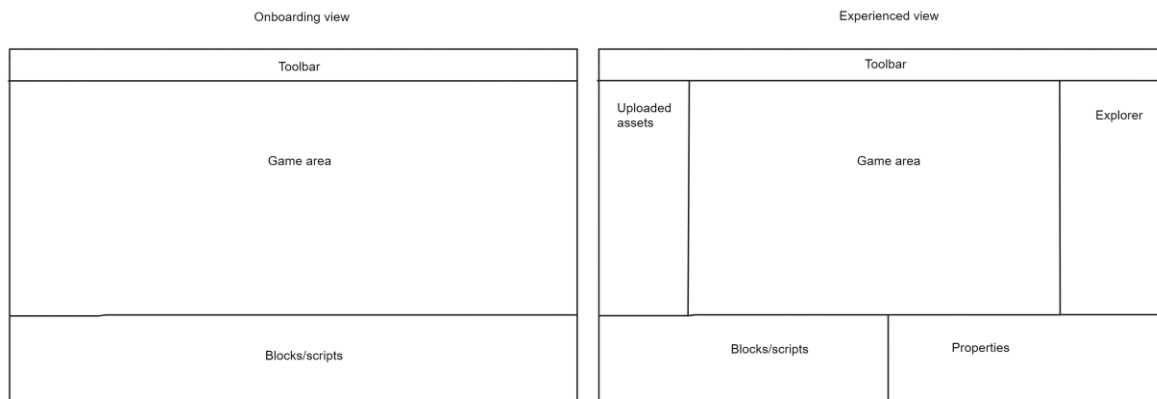
Disse værktøjer vil blive brugt til at fremstille et funktionelt spiludviklingsprogram.

Design specifikationer

Feature liste:

- Database hvor man kan opbevare/uploade levels
- Man skal have muligheden for at have dem lokalt
- Bygge og dele spillet uden at dele source koden (Ligesom Roblox med spil liste)
- Menuliste over spil projekter
- Kunne binde levels sammen
- Skifte mellem levels i én fil
- Drag and drop med forskellige typer components som statiske (for eksempel blokke) og dynamiske (som scripts man kan vælge fra en menu som åbner efter man klikker på de statiske components)
 - Firkant, trekant, ... (basic shapes)
 - Specielle pickups
 - Sværd, pistoler, etc.
- Skift mellem redigering og testing mode
- Support for sine egne teksturer
- Tilpasbar slutnings-/vinde/tabe betingelser
- Spriteark animationer
- Evt. eget script sprog (KUN hvis extra tid)

Sketch af vores spiludviklingsprogram UI-opbygning



Skitse af løsning

Onboarding view er det man ser når man åbner et nyt projekt og det har til formål at gøre sådan at folk der er nye ikke føler sig overvældet.

Experienced view dukker langsomt op når man placerer flere objekter eller interagerer med objekter. Når man vælger det første objekt dukker properties op med en infoboks der forklarer hvad den går ud på. Når man har placeret 10-15 objekter, dukker explorer vinduet op med en infoboks der forklarer hvad det gør. Uploadet assets menuen dukker op samtidig med explorer og har også en info boks. Det vil selvfølgelig være muligt manuelt at fremkalde menuerne inden da, og kollapse dem igen.

Tidsplan

Dag 1 22/2/23

Projektbeskrivelse

in list to do

Members

O

TS

+

Notifications

Watching

▼

Due date

5 Mar at 23:00

▼

Add to card

Members

Labels

Checklist

Dates

Attachment

Cover

Custom Fields

Power-Ups

+ Add Power-Ups

Automation

+ Add button

Actions

→ Move

Copy

Description

Add a more detailed description...

indhold

57%

Hide checked items

Delete

☒ Nøgleproblem

☒ Problemanalyse-mini

☒ Problemtree

☒ Problemformulering

☐ Konkurrent og målgruppeanalyse

☐ Krav

☐ Ide generering og sortering

Add an item

Dag 2 27/2/23

backlog

problemanalyse

+ Add a card

to do

+ Add a card

doing

+ Add a card

done

Projektbeskrivelse

5 Mar 7/7

+ Add a card

scrapped

+ Add a card

Dag 3 1/3/23

backlog

problemanalyse

+ Add a card

to do

+ Add a card

doing

+ Add a card

done

(9) Projektbeskrivelse [7]

5 Mar 7/7

+ Add a card

scrapped

+ Add a card

Dag 4 3/3/23

backlog

problemanalyse

+ Add a card

to do

+ Add a card

doing

+ Add a card

done

(9) Projektbeskrivelse [7]

5 Mar

7/7

0

15

+ Add a card

scrapped

+ Add a card

