

# Et computerspil til matematikundervisning i folkeskolen

---



af

Andreas Damgaard Olsen

---

B.Sc.-2013-22

Institut for Matematik og Computer Science

Danmarks Tekniske Universitet

Matematiktorvet

Building 303 B

2800 Kgs. Lyngby

Tlf: 45 25 30 31

## Resume

Denne rapport handler om udviklingen af et spil til børn i alderen 7-12 år. Spillet skal lærefolkeskoleelever matematik, og der fokuseres især på gangestykker. Da et gangestykke kan repræsenteres på flere måder, såsom et areal, vil dette blive brugt i spillet. For at gøre spillet spændende, skal eleven kunne interagere med spillet, på den måde at det følte som andre spil, som børn spiller på internettet. Børn skal lære selv at finde en måde at udregne matematik, og dette kan gøres med et spil, da spil lægger op til at man prøver sig frem.

Konkurrence er en stor del af børns interesse i spil, og de skal derfor opnå resultater, som kan sammenlignes med deres klassekammerater. Dette gøres ved at implementere kommunikation med en database, som gemmer elevens information. Yderligere skal denne information kunne tilgå læren, så information kan bruges som evaluering af en elev.

Da de fleste skoler i dag har computere og internet til rådighed, lægges spillet online, så eleverne ikke behøves at installere noget på deres computere. Spillet kan findes på hjemmesiden: [matskole.compute.dtu.dk/bscf134](https://matskole.compute.dtu.dk/bscf134). Da matematik indgår i spillet, vil det være udmærket materiale, som kan bruges i skolen, til oplæring i af matematik i matematiktimerne. Med tiden kan flere spil blive lavet, og man kan samle det hele under en hjemmeside, hvor det ikke kun er matematik, men for eksempel også dansk og engelsk.

Da det er børn der skal spille spillet, skal det være så let tilgængeligt som muligt. Derfor er simple implementeringer brugt til udvikling af spillet, såsom JavaScript og html, som ikke kræver plug-in til at kunne køre i de fleste browsere.

Ved at få spillet testet hos en folkeskole, har det vist sig at spillet har stor fordel i forhold til videreudvikling. Andre regnestykker er nemt at implementerer, og spillet kan derfor også bruges til plus, minus og division. Ud fra testene og elevernes svar, er spillet blevet taget godt imod, da det netop er simpelt, når man ved hvordan det spilles. Da der ikke sker så meget på skærmen ad gangen, er det let overskueligt. Det fik nogenlunde god karakter, og med forbedringer i forhold til børnenes ønsker, kan spillet gå hen og blive en god løsning til læring af matematik i skolen.

## Abstract

This report is about the development of a game for children aged 7-12 years. The game should teach school students math, and focus in particular on multiplication. As a multiplication may be represented in several ways, such as an area, it will be used in the game. To make the game more playable, the student should be able to interact with the game in a way that look like other games on the internet. Children must learn by doing, to calculate the math in the game.

Competition is a big part of children's interest in the games and they must therefore obtain results that can be compared to their classmates. This is done by implementing a server-client based game with a database which stores student information. Further, this information should be visible to the teacher, such that the information can be used as evaluation of the student.

Since most schools today have computers and internet available, the game should be playable online, which means that students don't need to install anything on their computers. The game can be found on the webpage: [matskole.compute.dtu.dk/bscf134](http://matskole.compute.dtu.dk/bscf134). Since mathematics is used in the game, it will be good material that can be used in school for training mathematics in math lessons. Over time, more games will be made, and you can bring it all together under one website where it is not only mathematics, but also teachings in languages.

Since the target group is children who will play the game, it should be as easily accessible as possible. Therefore, simple implementations are used for development of the game, such as JavaScript and HTML that does not require plug-ins, to run in most browsers.

By testing the game at a primary school, the game has great advantage for further development. Other calculations are easy to implement, and the game could also be used for addition, subtraction and division. Based on tests and student responses, the game has been well received, as it is the simple when you know how to play it. Since the game is pretty simple, it is easily manageable. It got fairly good graded, and with improvements to the children's wishes for further improvement, the game can turn out to be a good solution for the learning of mathematics in elementary school.

## Forord

Denne afhandling er udarbejdet ved Institut for Matematik og Computer Science, Danmarks Tekniske Universitet Danmark i en delvis opfyldelse af kravene for at erhverve Bachelorgrad i Software. Projektet startede: 07-02-2013 og denne afhandling blev afleveret: 28-06-2013. I afhandlingen behandles forskellige aspekter af software, og det forventes derfor at læseren har de fulgt, eller kendskab til de anbefalede kurser på bacheloren, software.

## Tak

Jeg vil gerne takke mine to vejledere, Jeppe Revall Frisvad (lektor på DTU Compute) og Niels Jørgen Christensen (Lektor på DTU Compute), for at vejlede mig igennem dette projekt. De har givet mig information og litteratur til at jeg kunne lave projektet.

Derudover vil jeg gerne takke Erik Ottar Jesen (Lærer på folkeskolen: Fuglevangsvej 5., Frederiksberg), som har bidraget med viden omkring hvordan børn lærer, og hvordan spillet skulle laves. Samtidig har han gjort det muligt at mit projekt kunne blive testet.

Jeg vil også gerne takke Michael Rose (Lektor på DTU Compute), som har givet introduktion om hvordan man designer et spil.

Til sidst vil jeg gerne takke eleverne på skolen (Fuglevangsvej 5., Frederiksberg), som har testet spillet. Det har været til megen stor hjælp til at gøre spillet bedre, og givet anledning til forbedringer.

## Indhold

Resume .....	3
Abstract .....	4
Forord .....	5
Tak .....	6
Introduktion.....	9
Analyse .....	10
Matematikken .....	10
Hvad gør et spil godt?.....	14
Hjemmesiden.....	17
HTML.....	17
Databasen.....	18
PHP .....	18
Python.....	19
Ruby.....	19
ASP.NET .....	19
Spil .....	20
Javascript .....	20
Design .....	22
Spillet .....	22
Inspiration.....	22
Matematisk færdighed .....	23
Beskrivelse .....	23
Spilfunktioner .....	24
Use cases .....	25
Univers.....	29
Implementering .....	30
Login system .....	30
Index .....	30
Check .....	30
Game .....	31
Update .....	31
Saved .....	31

Logout.....	32
Data ordbog.....	32
Spillet .....	33
Animate .....	34
Draw .....	35
DrawTanks .....	36
DrawCalc.....	39
DrawGrid.....	39
DrawGridArea.....	39
DrawBarrier .....	40
WriteScoreLife .....	40
DrawRocket .....	40
drawExplosion .....	40
Resultater og test .....	42
Diskussion .....	47
Forbedringer .....	47
Data ordbog over forbedret database .....	48
Forbedret Use cases .....	50
Konklusion .....	53
Referencer .....	54
Bilag .....	55
Hvordan man spiller spillet? .....	55




## Introduktion

Børn i dagens Danmark spiller meget computer, Iphone, Playstation og andre former for spillemaskiner. Nogle børn bliver desværre afhængige af at spille, og det kan gå ud over lektierne. Nu hvor børn er så vant til spillemaskiner, hvorfor så ikke bruge dem i undervisningen? Hvad nu hvis man kunne koble de to ting sammen, så børnene både kunne spille computer og stadig lærer matematik. Hvis man nu kunne lave et spil som havde elementer af matematik, som måske kunne overtage noget af undervisningen i folkeskolen, og ville endda nå længere ud end normal undervisning. Elever som ikke følger med i timen, behøves ikke være dårlige til matematik. Det kan være måden de lærer på som er anderledes end andres. Ifølge en skotsk undersøgelse i 2008, var spillet "Brain Training from Dr. Kawashima" bedre til at lære børn matematik end den oprindelige form for undervisning. Spil kan bruges til at engagere eleverne mere, og give dem interessen som de ikke fik ved matematikhæfter og forelæsning<sup>1</sup>.

Der findes mange spil, som allerede bliver brugt i folkeskoler til at lære matematik. Men de fleste går ud på det samme som matematikhæfterne indeholder. Det handler om at regne en masse regnestykker, og eleven lærer på den måde hvordan man regner. Dette er desværre ikke alle som lærer på den måde, og det er bedre hvis de ser en sammenhæng i regnestykkerne.

Gangestykker i matematik ses ofte som 2 tal ganget sammen, og resultatet giver et nyt tal. Men et gangestykke kan også ses som et areal, og det er måske ikke alle elever som forstår det.


$$\begin{array}{ccc} \text{||} & & \text{||} \\ \text{6} & = & \text{3x2} \end{array}$$

Den grundlæggende viden i matematikken mangler, og der bliver nødt til udarbejdes en måde hvorpå eleven kan se sammenhængen. De fleste spil på internettet fokusere kun på den ene måde at vise det på, men viser ikke sammenhængen mellem dem.

Der er derfor brug for et spil, som viser den sammenhæng. På den måde lader man eleven selv bestemme hvordan de vil forstå et gangestykke.

---

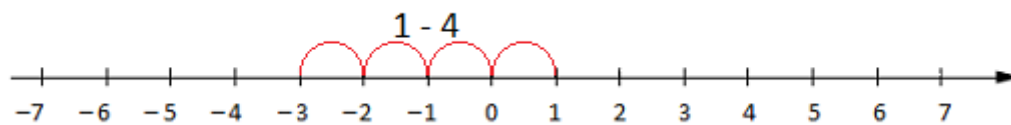
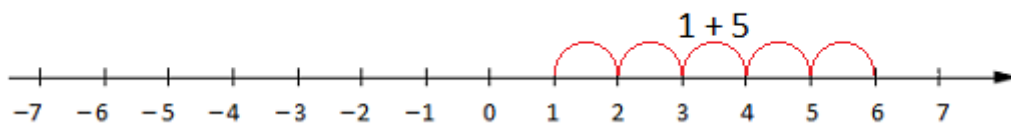
<sup>1</sup> <http://videnskab.dk/kultur-samfund/computerspil-bedre-end-matematiklaerer>

## Analyse

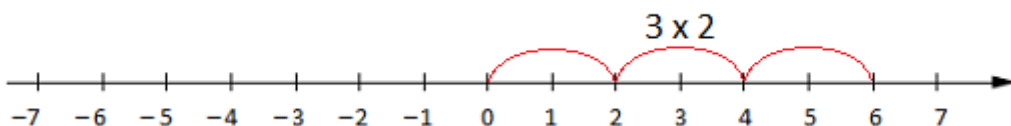
### Matematikken

Det vigtigste element i spillet er at matematik indgår. Børn i alderen 7-12 år skal ikke lærer ved at regne en masse regnestykker, men de skal derimod se en sammenhæng mellem regnestykkerne. 3 ting som er svære for eleverne at forstå er: tallinjen, positionssystemet og gange som areal. Tallinjen bruges til at se sammenhængen mellem at regne almindelige regnestykker såsom addition, multiplikation og subtraktion.

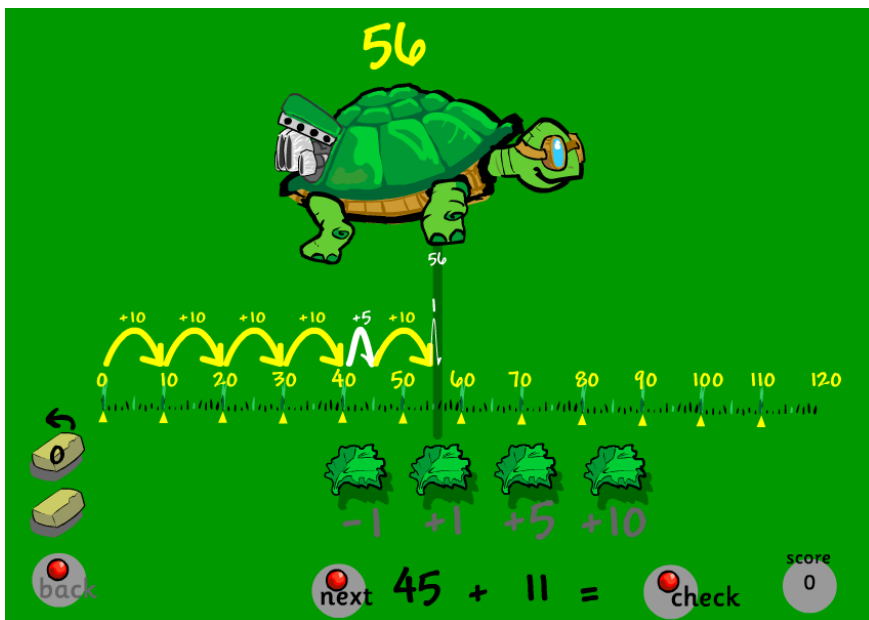
Addition og subtraktion skal eleven se som en lige bevægelse igennem tallinjen:



Mens multiplikation skal ses som spring på tallinjen:

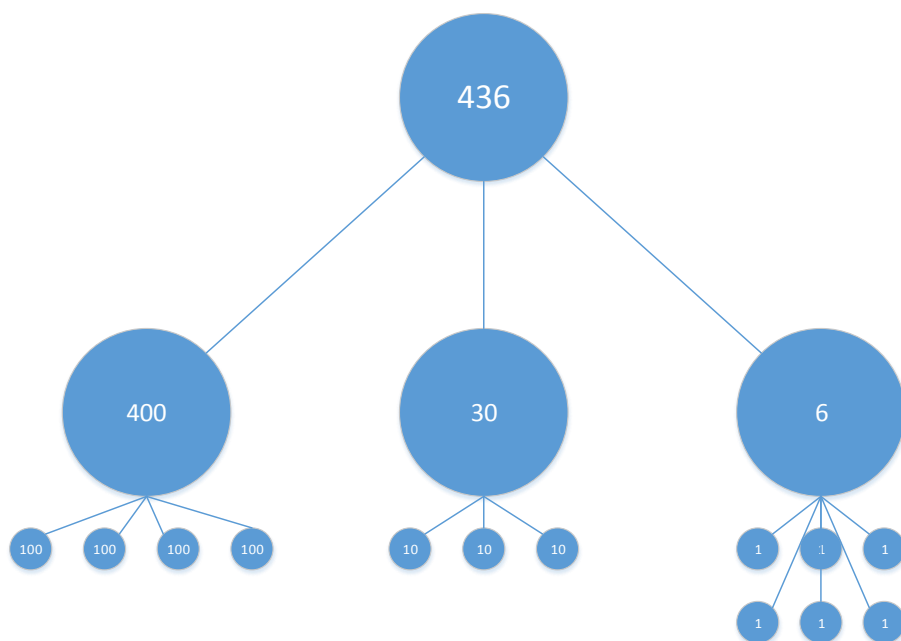


På internettet findes mange spil som lærer matematik til elever i de tidlige klasser, og et af de mange spil er dette tallinjespil, hvor en skildpadde flyver igennem tallinjen <sup>2</sup>. Dens position er tallet som ønskes udregnet, og eleven kan lægge til eller trække fra:

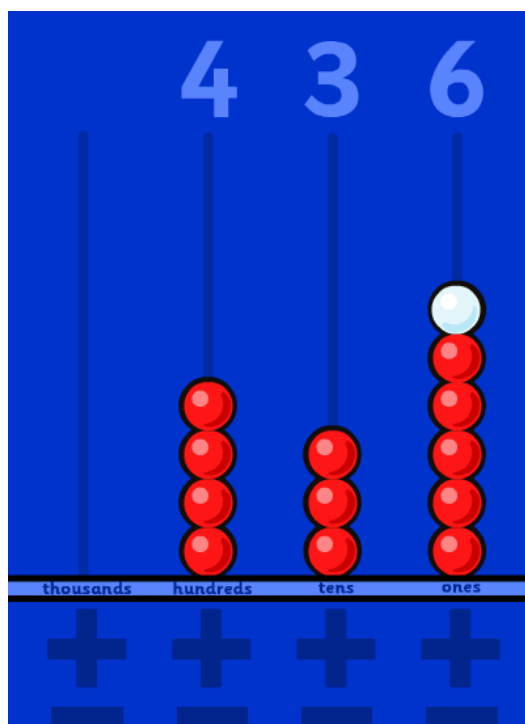


Positionssystemet skal hjælpe eleven med at forstå større tal som en samling af tal. Mange møntspil fokuserer på dette. Hvis man skal betale 436 kroner, kan man betale med 436 1-kroner. Men det er langt bedre at betale 4 hundredkronesedler, 3 tiere og 6 1-kroner. Eleven skal altså kunne dele store tal op til mindre overskuelige tal.

<sup>2</sup> <http://www.ictgames.com/technowithflock.html>



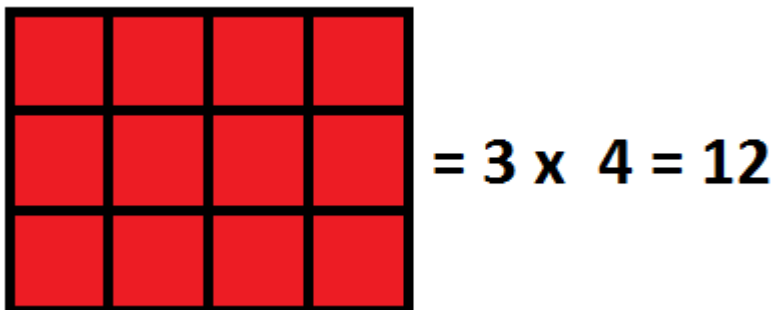
Et Eksempel på et spil som fokusere på positionssystemet er dette spil <sup>3</sup>:



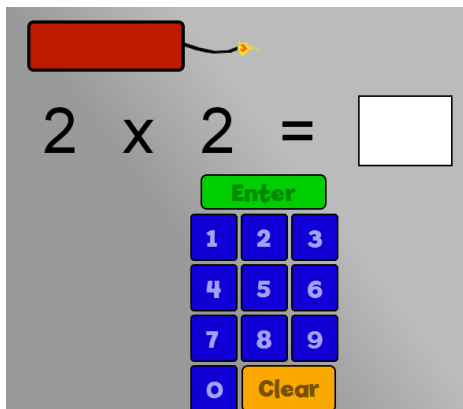
Ved at placere kugler på kuglerammen, kan man tælle op. Jo længere til venstre man befinder sig på rammen, jo mere svarer kuglerne til. Dette spil minder meget om kuglerammen som de fleste voksne er vokset op med, før man fik lommeregneren.

<sup>3</sup> <http://www.ictgames.com/abacusInteger.html>

Eleverne skal lære at gangestykker kan defineres på flere måder. Det kan for eksempel vises som et tal, et regnestykke eller som et areal.



Der findes igen mange spil på internettet som lærer om at regne gangestykker ud. Det gælder her om at regne fra en af de tre former til den anden<sup>4</sup>.

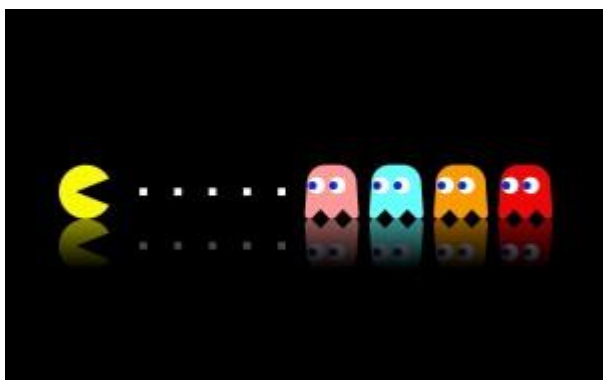


<sup>4</sup> <http://www.multiplication.com/games/play/dynamite-multiplication>

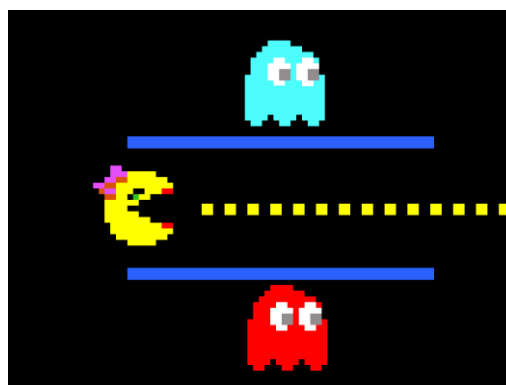
## Hvad gør et spil godt?

For at få børn til at spille et spil som lærer dem matematik, skal spillet være sjovt. Det skal ikke bare være en matematikbog som er lavet digital. Derfor skal spillet have spilelementer i sig, som gør spillet godt og sjovt at spille. Alle individer har hver deres interesser inde for spil. Nogle kan lide action og skydespil, så som Counter Strike hvor man skal reagere hurtigt. Mens andre vil spille stille og rolige spil som giver en tid til at tænke over sine handlinger, såsom Sims. Derfor er det svært at lave et spil som når ud til alle. Dog kan man lave et spil som opfylder grundlæggende regler, som gør et spil godt og spændende at spille.

Spillet skal være noget nyt, som brugeren ikke har prøvet eller set før. Man ville aldrig få succes inde for et spil, hvis man laver noget som allerede findes. De fleste kender spillet Pacman, som har været en stor succes i spillehaller rundt omkring i verdenen. Men der er også lavet en mere feminin version af det, som hedder Pacgirl<sup>5</sup>, som går ud på det samme som pacman, bare med en feminin side. Dette spil er det samme som det originale spil, men har ikke fået ligeså stor succes, da der ikke er udviklet noget nyt i spillet.



Figur: Det kendte spil Pacman



Figur: Det knap så kendte Pacgirl

Et meget vigtigt element i spillet er at der er sjovt at spille igen. Hvis man hver gang spiller et spil, og det føltes som om at spille det samme spil igen og igen, bliver det kedeligt i længden. Derfor skal brugeren have mulighed for at have indflydelse på spillets gang. En godt eksempel er de fleste rollespil som findes. Man bygger sin karakter op som man har lyst, og det er højst usandsynligt, at næste gang man spiller spillet, at resultatet bliver det samme. Hvis der sker ting i spillet som brugeren ikke forventer, vil spillet gøres mere interessant at spille, da det på den måde ikke føles som det samme spil som man har spillet før, da man ikke kan forudse hvad der sker.

Et spil er ikke sjovt mere når det går dårligt for en. De fleste mennesker har prøvet at spille et brætspil, med en som nogle vil kalde en dårlig taber. Så snart personen ser sit nederlag, kastes brætspillet op i luften,

<sup>5</sup> [http://uk.divertis.com/jeu/game1687\\_Pacgirl](http://uk.divertis.com/jeu/game1687_Pacgirl)

og spillet er slut for alle. Derfor skal et spil helst altid har en mulighed for alle at vinde, selv i slutningen af spillet. Et af de elementer som gør det kendte spil Counter Strike sjovt at spille, er at når man dør i spillet, går der ikke lang tid til man genopstår igen. Hver bane er et nyt spil, og alle har næsten den samme chance for at vinde.

Et spil skal være sammenhængende. Når spilleren spiller spillet, skal det give mening de ting som sker. Hvis man søger på nettet efter actionspil, og My Little Pony er et søgeresultat, giver det ingen mening. Man skal sørge for man vælger et tema til et spil, og holder sig til dette. Hvis man laver en forhistorisk spil, må der ikke indgå rumskibe, da dette vil ødelægge ens forventning til spillet.

For at kunne spille et spil, skal man kende reglerne bag. Hvis der er for mange regler, bliver det for uoverskueligt. Man skal kunne gøre mange ting, men man skal kunne gøre det nemt. De fleste spil i dag, bruger piletasterne eller wasd systemet, samt musen, til at styre en person rundt med. Det nytter ikke noget, hvis man for at spille spillet skal bruge halvdelen af keyboardet, da man vil glemme hvad de forskellige taster gør. Men derimod skal det heller ikke være for nemt. Hvis et bilspil handler om at køre ligeud, og man ikke bruger andet end op-piletasten, er det ingen udfordring, og spillet bliver for kedeligt.

Alle disse elementer skal tages med i overvejelserne ved produktionen af et spil. At disse elementer bliver overholdt, betyder ikke man får succes, men det kan guide en til et godt spil.



Figur: Wordfeud<sup>6</sup>



Firgur: Scrabble til android<sup>7</sup>

Hvorfor at det kendte spil Wordfeud har fået stor succes, men mange andre kopier ikke har, kommer an på brugerne. Vi er alle forskellige, og der er ikke nogen decideret guide til succes ved et spil.

<sup>6</sup> <http://www.feudmaster.dk/artikler/sla-et-wordfeud-snydeprogram/>

<sup>7</sup> <http://www.onlineandroidtips.com/apps/scrabble.html>



## Hjemmesiden

Da spillet skal være nemt tilgængeligt for eleverne, vil spillet blive kørt over en hjemmeside. På den måde kan de spille det i skolen eller derhjemme. Der findes forskellige måder at lave en hjemmeside. En af de nemmeste måder, men ikke så åbne metoder, er at lave en hjemmeside online. Der findes utallige hjemmesider, for eksempel [www.webs.com](http://www.webs.com) eller [www.wix.com](http://www.wix.com), hvor man ikke behøves nogen form for baggrundsviden, for at kunne lave en hjemmeside. Med et "drag and drop" system, kan man med snilde opsætte en hjemmeside på meget kort tid. Dog har de fleste, det krav om at der skal være reklamer fra forskellige sponsorer. Og samtidig har man ikke mulighed for at personliggøre layoutet ned til den mindste detalje.

## HTML

De fleste webudviklere bruger sproget HTML til at lave en hjemmeside<sup>8</sup>. HTML eller Hypertext Markup Language er et sprog som bruges til at fremstille en hjemmeside. HTML tager sig af udseendet og layoutet på en hjemmeside, og interaktionen med en hjemmesides udseende. Det indebærer også knapper og links til andre hjemmesider. Man loader for eksempel et billedet, og man kan så bruge HTML til at gøre et billede til et link til en anden hjemmeside. På den måde kan det med fortrinsvis bruges til reklamer på hjemmesider. Eksempel på simpel htmlkode:

```
<title>Hjemmesidens titel</title>
</head>
<div style="color:#0000FF">
  <h3>Dette er heading</h3>
</div>
<body>
Sig noget!
<form name="form1" method="post" action="">
  <input type="submit" name="b1" id="b1" value="Knap">
</form>
En tabel:
<table width="200" border="1">
  <tr>
    <td>Hej1</td>
    <td>Hej2</td>
    <td>Hej3</td>
  </tr>
</table>
<p><a href="www.matskole.compute.dtu.dk/bscf134" title="Det bedste spil i
verden">matskole.compute.dtu.dk/bsc134</a></p>
</body>
</html>
```

---

<sup>8</sup> <http://da.wikipedia.org/wiki/HTML>

Som vil vise dette:

## Dette er heading

Sig noget!

En tabel:

Hej1	Hej2	Hej3
------	------	------

[matskole.compute.dtu.dk/bsc134](http://matskole.compute.dtu.dk/bsc134)

HTML er simpel at bruge, og det er kun en spørgsmål om tid og fantasi som sætter grænser for udseendet på en hjemmeside. Ligesom online websider som gør det nemmere at lave hjemmesider, findes også programmer, som henvender sig til lidt mere erfarne brugere.

## Databasen

Da spillet skal laves til folkeskoler og børn, skal de have muligheden for at logge ind og holde styr på deres progression. En primitiv mulighed var at gemme information i en fil, men det er langt mere sikkert og brugbart at gemme information på en database. En database er nem at tilgå, og den kan udvides, hvis ny information skal logges. Ved at holde styr på administratorrettigheder, er det muligt, for eksempel lærerne, at oprette nye brugere.

## PHP

PHP, Hypertext Preprocessor, er et programmeringssprog som anvendes til kommunikation mellem klienten og serveren. Det er oplagt værktøj til webdesign, da det kan samarbejde med HTML. På den måde kan man nemt få adgang til databasen fra sin hjemmeside.

For	Imod
Open Source, Nemt at få hjælp på internettet	Dårlig problemhåndtering
Fleksibel, PHP kan nemt kommunikere med andre datatyper	Langsom i forhold til andre sprog ved større mængder data
Meget populær	

9

<sup>9</sup> <http://www.wisegeek.com/what-are-the-pros-and-cons-of-php.htm>

## Python

Python er et platformsafhængigt programmeringssprog som bliver brugt over hele verden.

For	Imod
Simpelt programmeringssprog, dog imponerende	Ikke samme syntax som de fleste programmeringssprog
Open Source	Har ikke meget kendskab til det
Køre hurtigt, da det ikke kræver megen kode.	Svært at lære i forhold til andre sprog

10

## Ruby

Ruby er et simpelt programmeringssprog som anvendes til websides.

For	Imod
Simpelt programmeringssprog, dog imponerende	Svært at lære i forhold til andre sprog
Fleksibelt	Har ikke meget kendskab til det

11

## ASP.NET

ASP.NET bruges til udvikling af webapplikationer, hjemmesider og webservices som bruges af Microsoft. Sproget som mest benyttes er C# eller Visual Basic<sup>12</sup>.

For	Imod
Fleksibelt	Er ikke open-source
Objekt-orienteret	Svært at lære i forhold til andre sprog
	Har ikke meget kendskab til det
	Kun til Windows platform
	Kræver mere data på serveren

13

<sup>10</sup> <http://stackoverflow.com/questions/4812600/what-are-the-pros-cons-of-teaching-javascript-vs-python-as-a-first-programming>

<sup>11</sup> <http://nerdfortress.com/2008/05/13/ruby-on-rails-pros-cons-lets-get-real/>

<sup>12</sup> <http://da.wikipedia.org/wiki/ASP.NET>

## Spil

Spillet vil højst sandsynligt køre i et canvas. Med der er flere måder at vise canvasset på. Man skal have mulighed for at interagere med spillet, så derfor skal der indgå listenere i spillet.

## Javascript

Javascript er et af de simpleste sprog til funktionsudvikling på en hjemmeside. Det bruges for eksempel til udregninger på siden, som ikke behøver adgang til en database, men som kan udregnes ud fra brugerens valg. HTML kan ikke lave udregninger, og derfor har man brug for javascript. Det giver også brugeren mulighed for at interagere med hjemmesiden, og bliver ofte brugt til for eksempel online spil<sup>14</sup>.

For	Imod
Open Source	Da udregningerne sker på klientsiden, vil det være muligt at hacke informationen, som skal sendes.
Hurtigt, da udregninger sker på klientside.	
Simpelt, nemt at lære	
Arbejder godt sammen med andre sprog, og kan nemt indsættes i en hjemmeside.	
Da udregninger udregnes på klientsiden, vil serveren ikke blive overbelastet.	
Er det mest anvendelige sprog til interaktion på hjemmesider <sup>15</sup> .	

<sup>16</sup>

---

<sup>13</sup> <http://answers.yahoo.com/question/index?qid=20080820120329AADDnbw>

<sup>14</sup> [http://referencedesigner.com/tutorials/js/js\\_1.php](http://referencedesigner.com/tutorials/js/js_1.php)

<sup>15</sup> [http://en.wikipedia.org/wiki/Programming\\_languages\\_used\\_in\\_most\\_popular\\_websites](http://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites)

<sup>16</sup> <http://www.mediacollege.com/internet/javascript/pros-cons.html>

## WebGL

WebGL er en softwaregrænseflade, som gør det muligt at vise 3d og 2d elementer på en hjemmeside. Det bliver understøttet af de fleste webbrowsere, og bliver derfor ofte brugt til repræsentation af 3d grafik. Da man arbejder i 3d, gør WebGL det også muligt at integrere fysik i sit spil. Ting vil kunne få tyngdekraft på og biler vil kunne accelerere og køre rundt. Samtidig har man kontrol over alt form for kamera synspunkt, lys og skygger, som kan gøre spillet mere realistisk<sup>17</sup>.

For	Imod
Hurtigt hvis man skal lave større programmer, det vil sige hvis der skal tegnes mange ting.	Kan være svært at lære
Fysik integration	Kræver plugin på diverse browsere

18

## Canvas

Udover at bruge en softwaregrænseflade, kan man bruge man bruge canvas2D til at tegne spillet. Canvas skal ses som et lærred. Det vil sige at hvis man tegner noget, vil dette vises indtil man tegner noget nyt. Hvis man vil have animation ind i billedet, bliver man nødt til at vise billedet, slette et objekt, tegne objektet et nyt sted, og derefter vise billedet igen. Det minder lidt om, når man laver en stop-motion video med modellervoks<sup>19</sup>.

For	Imod
Nemt at kode	Ved større programmer skal billedet tegne mange ting, og det vil blive langsomt
Virker i alle browsere med HTML5	

20

<sup>17</sup> <http://en.wikipedia.org/wiki/WebGL>

<sup>18</sup> <http://curvedcatgames.wordpress.com/2011/08/27/html5-canvas-vs-webgl/>

<sup>19</sup> <http://html5doctor.com/an-introduction-to-the-canvas-2d-api/>

<sup>20</sup> <http://curvedcatgames.wordpress.com/2011/08/27/html5-canvas-vs-webgl/>

## Design

## Spillet

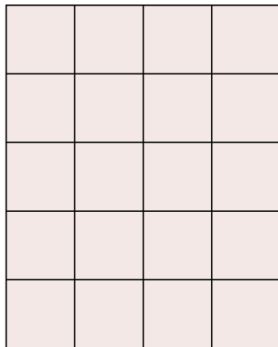
### Inspiration

Spillet tager udgangspunkt i spillet Laste Biler<sup>21</sup>:

## Laste Biler

### Matematisk færdighed

Her arbejdes med repræsentationen af gange som et areal.



$$4 * 5$$

### Beskrivelse

Spillet foregår på en pakkecentral af en art. En pålæsningsplads til lastbiler. Spillerens opgave er at være mellemlid mellem lastbiler og pakker og skal sørge for at de rigtige pakker kommer de rigtige steder hen. Pakker kan være repræsenteret ved arealer eller gangestykker. Ligeledes kan lastbilernes lasteevne være repræsenteret ved areal eller gangestykker.

Konceptet er at man har en ordreliste med forskellige kunder der skal have forskellige antal pakker (evt. også karter med et antal pakker i). Lastbiler ankommer løbende med forskellig kapacitet. Man kan også have mulighed for at bestille en lastbil med kapacitet givet ved et gangestykke. Det vigtige er at man arbejder med multiplikation på forskellige måder. På højere levels kan man også inddrage ubekendte.

---

<sup>21</sup> workshop (torsdag d. 7. februar 2013) af Erik Ottar Jensen

En form for bonus kan være et godstog der kommer ind en gang imellem hvor der er rigtigt meget der skal læsses.

## Univers

Pakkecentral. Lasbiler kører til og fra tomme eller fulde og skal lastes eller aflastes.

## Matematisk færdighed

Der arbejdes med repræsentation af gange som areal.

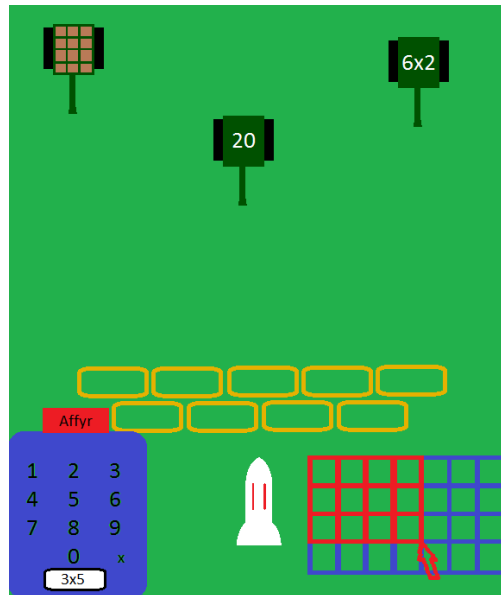
## Beskrivelse

Spillet foregår under en krig mellem spilleren og en fjende, styret af computeren. Fjenden sender tanks mod spillerens base, og det er op til spilleren at forsvare basen. Fjendens tanks er præsenteret ved et areal som kommer kørende på hjul, vist som areal, gangestykke eller tal. Jo større areal tanken har, jo langsommere kører den. Det vil sige at man har længere tid til at regne de store regnestykker, mens de små regnestykker som ikke kræver så lang tid, vil køre hurtigt. For at forsvare basen, er det op til spilleren at sende raketter af sted mod tanksene, som svarer til det rigtige resultat. For at det ikke er for nemt, kan et areal ikke blive besejret med et areal, men må regnes ved gangestykke eller tal.

	Gangestykke	Areal	Tal
Gangestykke	x	o	o
Areal	o	x	o
Tal	o	o	x

Tabel: Hvilke repræsentationer som kan besejre hvem. Et x betyder at den ikke kan besejres.

Spilleren kan definere tal eller gangestykke ved at bruge en lommeregner, hvor det ønskede resultat skrives ind. Areal bliver defineret ved at man kan tegne arealet på et grid.



Figur: Ide til spillet, hvor flere metoder er mulige til at definere et regnestykke på.

Hvis man indtaster et forkert resultat, og sender raketten af sted, vil tanken som rammes af raketten ikke destrueres. Hvis man ikke destruerer køretøjerne inden de når han til spillerens base, vil de køre ind i spillerens base, hvorved man mister liv. Når man når 0 dør man og spillet er slut.

Hvis man sender en raket af sted med det rigtige resultat, og rammer den rigtige tank, vil tanken destrueres, og man får point i forhold til resultatet. Det vil sige jo større regnestykke, jo flere point.

### Spilfunktioner

*Træk:* Spilleren kan trække arealer på et grid, som vil tegne et areal. Dette gøres ved at holde musen over et grid, og så sende raketten afsted.

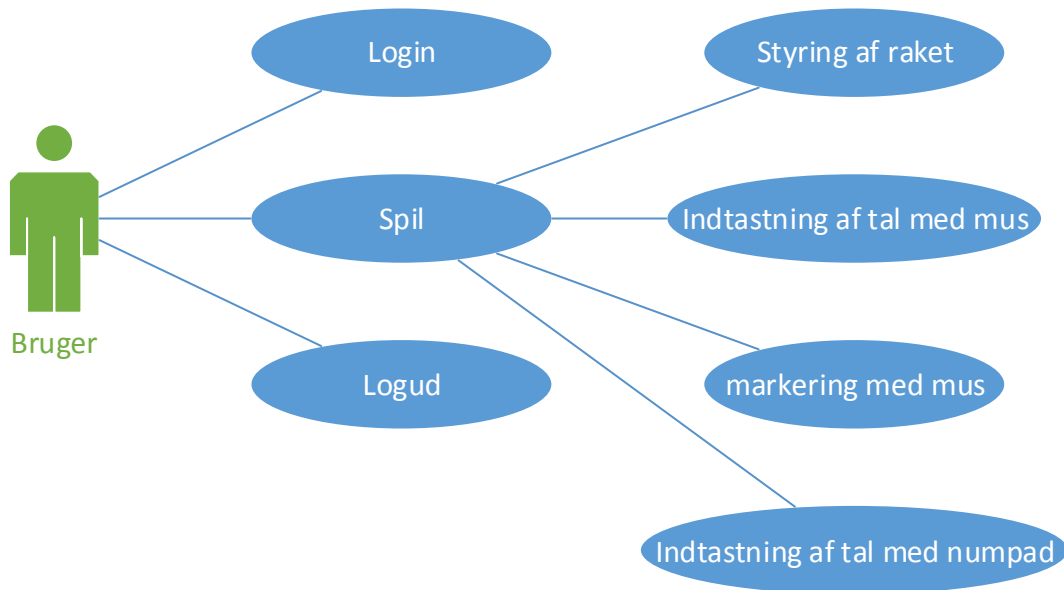
*Indtastning af tal både med mus eller tastatur:* Spilleren kan trykke på de viste tal og få det ønskede areal frem. Enten som regnestykker:  $4 \times 5$  eller tallet 20.

*Affyr knappen:* Når spilleren har indtastet det ønskede resultat, eller markeret det ønskede areal, kan man trykke på space, som vil affyre raketten.

*Styring af raket:* Raketten skal kunne styres af spilleren med piletasterne både før og efter at raketten er sendt af sted.



## Use cases



Lige nu er det muligt for et allerede oprettet bruger at logge på, og spille spillet. Scoren vil blive gemt, men det er ikke muligt at se scoren, med mindre man har adgang til serveren.

**Use case:** Login

**Handling:** Logge ind på hjemmesiden

**Aktør:** Bruger

**Hoved scenarie:** Brugeren kan logge ind med brugernavn og kodeord

- Brugernavn indtastes på brugernavn feltet
- Kodeord indtastes på kodeord feltet
- Der trykkes på ENTER eller knappen indsend
- Brugeren sendes videre til spillet

**Alternativt scenarie:** Brugeren kan ikke logge ind

- Brugernavn indtastes på brugernavn feltet
- Kodeord indtastes på kodeord feltet
- Der trykkes på ENTER eller knappen indsend
- Brugeren sendes ikke videre til spillet, som skyldes forkert brugernavn eller kodeord

**Use case:** Logud

**Handling:** Logge ud af hjemmesiden

**Aktør:** Bruger

**Hoved scenarie:** Brugeren kan logge

- Brugeren trykker på logud
- Brugeren sendes tilbage til startside

**Use case:** Spil – styring af raket

**Handling:** Styring af raket

**Aktør:** Bruger

**Hoved scenarie:** Brugeren kan styre raketten med piletasterne, og affyre på SPACE

- Brugeren trykker på højre eller venstre piletast
- Raketten bevæger sig henholdsvis til højre eller venstre
- Der trykkes på SPACE
- Raketten bevæger sig fremad på skærmen

**Alternativt scenarie:** Brugeren kan ikke bevæge raketten til venstre eller højre

- Brugeren trykker på højre eller venstre piletast
- Raketten bevæger sig ikke, som kan skyldes at raketten er ude ved grænsen af skærmen

**Use case:** Spil – Indtastning af tal med mus

**Handling:** Indtastning af tal med mus

**Aktør:** Bruger

**Hoved scenarie:** Brugeren kan indtaste tal på lommeregneren ved at bruge musen

- Brugeren trykker med musen på en af knapperne på lommeregneren
- Det ønskede tal vises i lommeregneren
- Flere knapper trykkes
- Flere tal vises i lommeregneren

**Alternativt scenarie:** Brugeren kan ikke indtaste tal på lommeregneren ved at bruge musen

- Brugeren trykker med musen på en af knapperne på lommeregneren
- Det ønskede tal vises ikke i lommeregneren, dette kan skyldes:
  - o Hvis det er første tal, kan det ikke være gange
  - o Hvis der er mere end 5 tal, er der ikke plads til flere
  - o Hvis der er 4 tal, kan man ikke trykke gange

**Alternativt scenarie:** Brugeren kan indtaste tal, men de vises ikke permanent på lommeregneren

- Brugeren trykker med musen på en af knapperne på lommeregneren
- Det ønskede tal vises ikke i lommeregneren
- Det ønskede tal forsvinder fra lommeregneren, dette kan skyldes:
  - o Raketten bliver affyret og rammer noget
  - o Musen holdes ind over gitterområdet

**Handling:** Indtastning af tal med tastatur

**Aktør:** Bruger

**Hoved scenarie:** Brugeren kan indtaste tal på lommeregneren ved at bruge numpad

- Brugeren trykker på det ønskede tal
- Det ønskede tal vises i lommeregneren
- Flere knapper trykkes
- Flere tal vises i lommeregneren

**Alternativt scenarie:** Brugeren kan ikke indtaste tal på lommeregneren ved at bruge numpad

- Brugeren trykker med det ønskede tal
- Det ønskede tal vises ikke i lommeregneren, dette kan skyldes:
  - o Hvis det er første tal, kan det ikke være gange
  - o Hvis der er mere end 5 tal, er der ikke plads til flere
  - o Hvis der er 4 tal, kan man ikke trykke gange
  - o Hvis brugeren ikke har num-pad på sin computer

**Alternativt scenarie:** Brugeren kan indtaste tal, men de vises ikke permanent på lommeregneren

- Brugeren trykker på det ønskede tal
- Det ønskede tal vises i lommeregneren
- Det ønskede tal forsvinder fra lommeregneren, dette kan skyldes:
  - o Raketten bliver affyret og rammer noget
  - o Musen holdes ind over gitterområdet
  - o Hvis der er mere end 5 tal, er der ikke plads til flere
  - o Hvis der er 4 tal, kan man ikke trykke gange

**Use case:** Spil – markering med mus

**Handling:** Markering af areal med mus

**Aktør:** Bruger

**Hoved scenarie:** Brugeren kan med musen markere det ønskede areal i griddet

- Brugeren holder musen ind over griddet

Området markeres

### **Univers**

Kommer an på tidsalderen. En form for ørken eller slagmark. Tanks kommer kørende fra toppen og ned.

## Implementering

### Login system

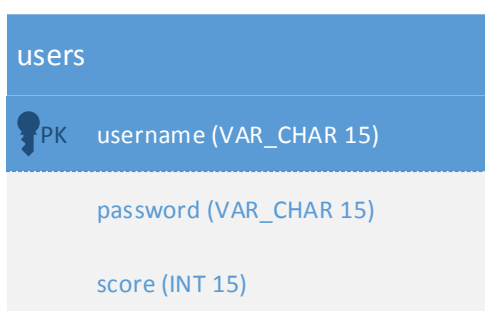
Som sprog til brug af kommunikation mellem klient og server bruges php. Dette skyldes at datamængderne som skal sendes ikke er store, og der derfor ikke er behov for at vælge et andet sprog. Php er simpelt at lære, og er populært blandt mange andre hjemmeside, som har brug for kommunikation mellem klienten og serveren.


### Index

Når man indtaster spillets hjemmeside, er den første side man møder index.php. Denne side giver mulighed for at logge ind. Øverst findes en bjælke tegnet, som kun er til gavn for det visuelle. Siden består af en form med to inputs, hvori brugeren kan indtaste sit brugernavn og password. Metoden er en post, så når man trykker enter eller klikker på knappen indsend vil brugernavn og password blive tilgængeligt via \$\_POST. Ved submit vil siden gå videre til siden check.php. Udover login består siden af et canvas hvori et billede vises af hvordan man styrer spillet.

### Check

Check.php starter med at forbinde til databasen. Da metoden post blev brugt i index, kan brugernavn og password hentes som \$myusername og \$mypassword. For at undgå injektion angreb på databasen, benyttes mysql\_real\_escape\_string, som sørger for at det er rigtig information som bliver sendt af sted. Når forbindelsen er oppe at køre, laves en query som indeholder antallet af brugere med angivne brugernavn og password fra databasens tabel over brugere.















users	
 PK	username (VAR_CHAR 15)
	password (VAR_CHAR 15)
	score (INT 15)

Figur: Tabel over brugere i databasen.

Hvis antallet er lig en, betyder det at der findes en bruger, og man kan blive sendt videre til game.php, samtidig med at brugeren bliver registreret i sessionen. Hvis der ikke findes en bruger, bliver man dirigerede tilbage til index siden, hvor man kan indtaste et nyt brugernavn og password.

Eksempel på tabel:

	username	password	score
<input type="checkbox"/>  Ret  Kopi  Slet user	123		59
<input type="checkbox"/>  Ret  Kopi  Slet test	4444		0
<input type="checkbox"/>  Ret  Kopi  Slet test2	2222		0
<input type="checkbox"/>  Ret  Kopi  Slet test3	3333		0

## Game

Game.php er siden som indeholder hoveddelen, altså spillet. Hvis brugeren er registreret i sessionen, vil spillet starte. Hvis ikke vil man komme tilbage til index siden, da det ikke skal være muligt at logge ud og stadig have mulighed for at spille spillet.

Game.php indeholder to forms: En logout knap som bruges til at logge ud, og en update som gemmer ens opnåede score i spillet. Denne form benytter ligesom index, post metoden, som gør det muligt at hente resultatet til senere anvendelse.

Udover de to former, indeholder siden et canvas, hvori spillet kører. Positionen af canvasset defineres og en baggrund bliver sat til spillet. Spillet er opbygget i JavaScript, som indhentes i filen game.js, som vil blive diskuteret senere i afsnittet.

## Update

Update.php startet med at forbinde til databasen. Da metoden post blev brugt i game.php, kan resultatet hentes som \$result. Igen for at undgå injektion angreb på databasen, benyttes mysql\_real\_escape\_string.

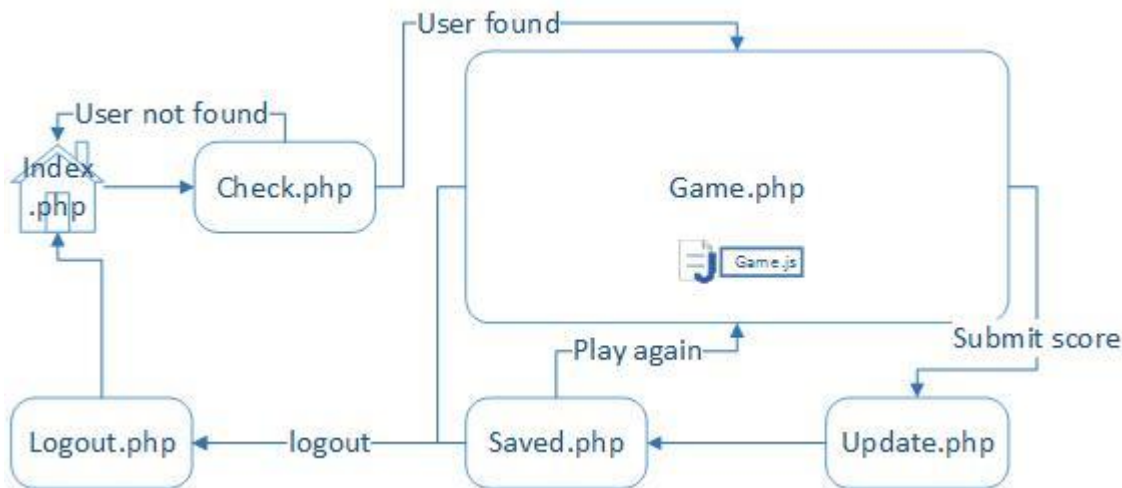
Når forbindelsen er oppe at køre, opdateres brugerens score ved at benytte sessionens registrerede brugernavn samt resultatet som kunne hentes fra tidligere. Resultatet lægges til den nuværende score, og man sendes automatisk videre til siden saved.php.

## Saved

Denne side anvendes efter at en score er blevet registreret i databasen. En tekst vises, hvorpå man bliver oplyst om at scoren er blevet gemt, og man får valgmuligheden om at starte et nyt spil eller logge ud. Det vil sige at der findes to former på denne side. Hvor den ene viderestiller brugeren til logout.php og den anden viderestiller brugeren til game.php.

## Logout

Logout.php tager sig at destruere den brugte session. Det vil sige at den registrerede bruger ikke længere vil være i stand til at genindlæse de anvendte sider. Kun hvis brugeren logger på igen med det korrekte brugernavn og password. Når session er destrueret sendes brugeren videre til index.php, og en ny bruger kan logge sig på.



Figur: Diagram over databaselogin.

## Data ordbog

**Users:** Holder styr på brugerne som kan logge ind og spille spillet.

**Username:** Navnet som brugeren skal benytte sig af for at logge ind. Dette er unikt, og kan derfor kun bruges til én elev. Hvis der er flere som hedder Jesper, må de vælge navne som for eksempel Jesper123.

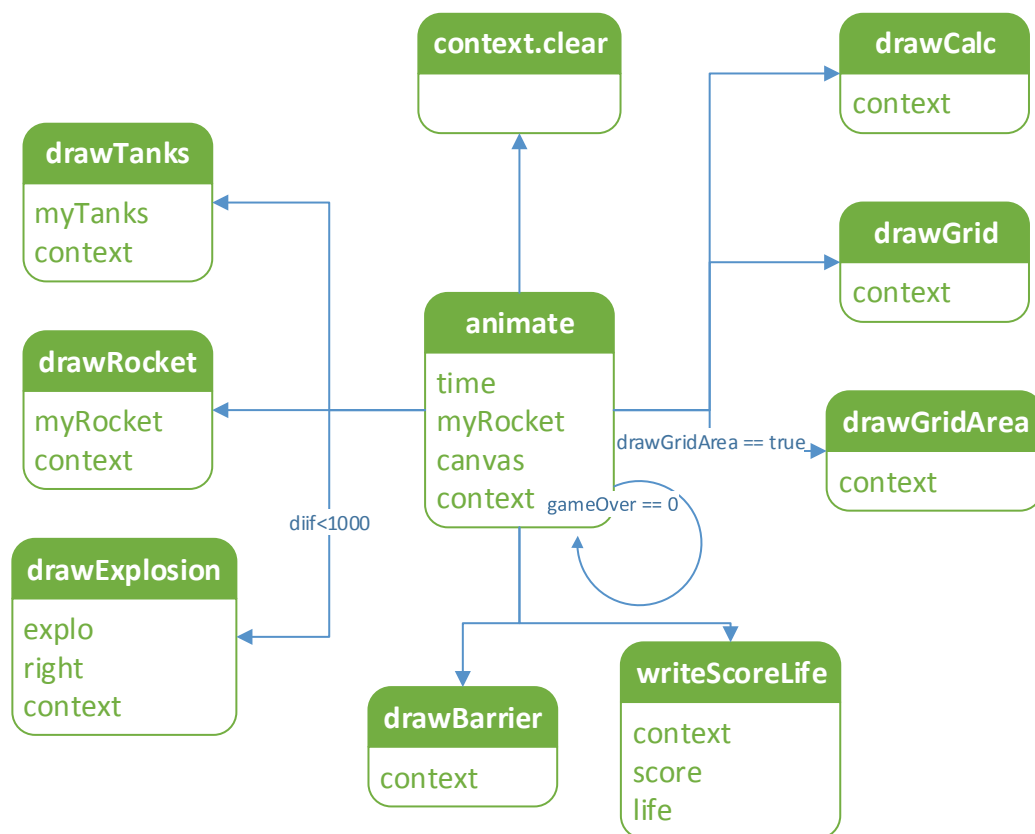
**Password:** Til hvert brugernavn hører der en kode til. Dette kan være en kombination af tal eller bogstaver som kun eleven kender til. På den måde vil ingen andre kunne logge ind på deres konto.



## Spillet

Spillet er implementeret i canvas2d. Dette er valgt, da der ikke skal tegnes mange operationer ad gangen. Endvidere da spillet skal ses fra oven, er 3d ikke nødvendigt, så derfor er der ikke behov for at implementere 3d. En af de vigtigste grunde til at canvas2d er valgt, er at nogle folkeskoler i dag, har en streng politik omkring at installere nye browsere og plug-ins på skolens computere. Canvas2d kan køre i de fleste opdaterede browsere, mens nogle softwaregrænseflader kræver et plug-in. Da målgruppen er børn, er det vigtigt at det er så nemt for dem at komme i gang med at spille, og hvis de først skal til at installere nyt software, bliver det besværligt, hvis de ikke selv kan finde ud af det.

Spillet består af to hoveddele. Der er området hvor tanksne kører, samt hvor raketten flyver (battlefield). Og så er der området som behandler lommeregner og grid (Udregning). Diverse tegnefunktioner tegner hvordan billedet ser ud lige nu, og hovedfunktionen animate holder styr på hvad der er sket siden sidst hvad angår området battlefield. Udover disse funktioner, findes et antal listenere som sørger for hvad brugeren trykker på i udregning området.



Figur: Forhold mellem funktionerne i spillet.

## Animate

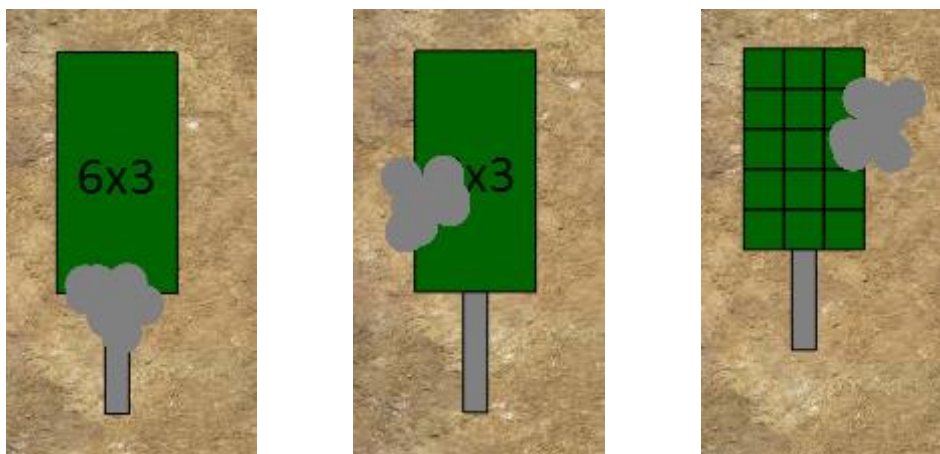
Animate er hovedfunktionen som kører spillet. Den kalder de forskellige tegnefunktioner som bruges igennem spillet, og det registrere hvad der er sket siden sidste visning af billedet. Der bruges tre forskellige listenere i funktionen animate.

Den ene listener er en mouseMove, som holder styr på hvornår musen er inde for grid området, da dette område kun skal markeres når musen holdes inde over. Når missilet sendes af sted, er det det markerede område som er registreret som resultatet som bruges til at kontrollere om en tank skal destrueres eller ej. Så længe musen er registreret over griddet, vil resultatet være gemt som typen "area". Hvis musen flyttes fra området, vil resultatet blive sat til nul, og man kan vælge en ny metode til at ødelægge tanksne.

En anden listener er en mouseDown, som holder styr på hvilke knapper der trykkes på, på lommeregneren. Hver gang der klikkes med musen registreres koordinaterne, men de bliver kun anvendt hvis de er inde for lommeregnerens knappeområder. På den måde vil der ikke ske noget, når man klikker uden for lommeregneren.

Den sidste listener som bliver anvendt er keyDown listenere. Denne bliver brugt to steder. Den ene anvendes til at gøre det nemmere for brugeren at indtaste tal. I stedet for at bruge musen til at klikke på lommeregneren, har brugeren også den mulighed at taste tallene ind på computerens numpad, hvis man har sådan en. Tallene fra 0-9 bruges til tallene, mens gange tasten og backspace bruges henholdsvis som gange og clear. Den anden anvendes til styring af raketten. Raketten kan, både før og efter affyring, styres til siderne med piletasterne højre og venstre. Hver gang de forskellige piletaster trykkes nede, vil raketten rykke sig til siderne langs x-aksen. Når space trykkes nede, sætte en boolean til sand, som fortæller at raketten skal bevæge sig fremad med en bestemt hastighed på y-aksen.

Da animate holder styr på hvordan næste billede vil se ud ved næste visning, ved den også om en raketten vil ramme en tank. Dette klares ved hele tiden at kontrollere om raketten er inde for en tanks område. Da tanksene har forskellige størrelse kontrollere raketten's x og y koordinat med tanksen sidder, samt den bund, da det ikke er muligt at ramme en tank bagfra:



Figur: Kollision mellem raket og tanks.

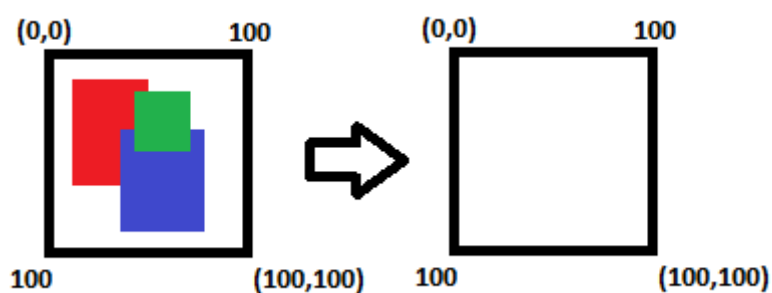
Da tanksene ikke kan rykke sig til siderne, kontrollere først om raketten er inde for tanksen bredde, hvis den er det, kontrollere herefter om den også er inde for den højde, ved at tage en tanks x og y koordinat, som er defineret i tanken, og lægge dens højde til.

### Draw

Som nævnt tidligere virker canvas som stop-motion. For hver gang billedet bliver tegnet, skal det forrige slettes. Dette gøres ved at kalde `clear.rect`, som fjerner alt inde for det angivne areal.

Før `clear.rect(0,0,100,100)`:

Efter `clear.rect(0,0,100,100)`:



Der gøres opmærksom på at baggrunden ikke fjernes, hvis denne er angivet i HTML, så der er altså kun de elementer som er ved henvendelse af draw funktionerne i javascriptet. Da baggrunden ikke skal ændres i spillet, er denne sat i HTML dokumentet.

`Clear.rect` bliver kaldt lige før de individuelle draw funktioner bliver kaldt. På den måde har man et nyt lærred at tegne på. Hvis `clear` ikke blev kaldt ville gamle tegninger stadig være vist, og dette ville danne et spor efter animationerne. Et eksempel er vist herunder:

Med clear:



Uden clear:



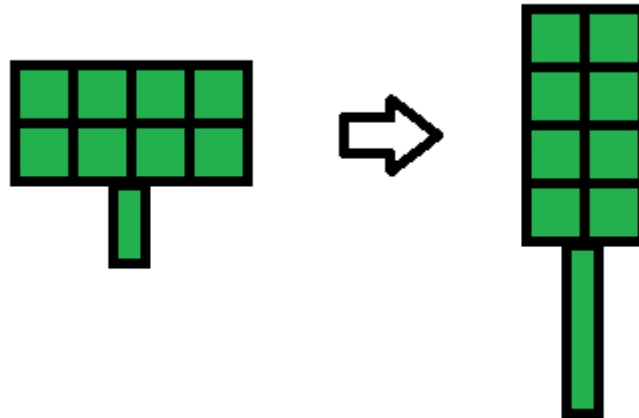
Der bruges 7 draw funktioner som tager sig af indholdet i spillet. Derudover bruges også en funktion til at vise spillerens score og liv.

### DrawTanks

Denne funktion tegner tanksene I spillet. En tank er defineret ved et array bestående af 6 værdier. De 6 værdier:

1. Bruges til at vide om tanken er i live. Hvis værdien er sand, betyder det at tanken stadig er i live, og at den vil blive tegnet i spillet. Hvis værdien er falsk, betyder det at tanken er destrueret, og en ny tank vil blive dannet med nye værdier.
2. Definerer tankens ene tal. Alle tanksene består af et regnestykke. Regnestykket består så af 2 tal som ganges sammen. Dette tal bestemmer højden på tanken, så der vil kunne ses en sammenhæng mellem areal og regnestykke.

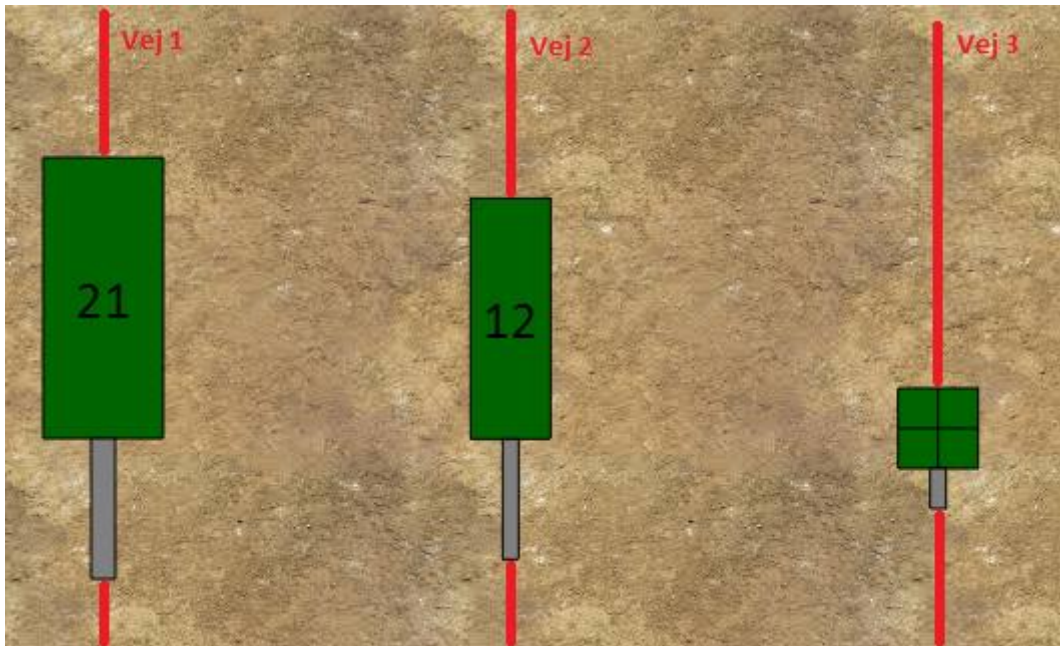
3. Definerer tankens andet tal. Dette tal bestemmer så bredden på tanken. Da de to tal dannes tilfældigt, kontrolleres de to tal op mod hinanden, og det største tal sættes som højden, da dette ser bedre ud rent visuelt.



Figur: Tankens højde og bredde bytte rundt, hvis bredden er større end højden

De to tal vælges mellem værdien 2 og variabelen  $\max N$ . Tallet 2 er mindste værdier, da det ikke giver mening at skulle udregne regnestykker såsom  $1 \times 7$ .  $\max N$ , defineres fra start af, og dette tal er hvad man maksimalt kan møde. Hvis tallet er 5, vil man ikke få gangestykker som indeholder mere end 5, det vil sige tal som  $4 \times 2$  og  $5 \times 3$  er mulige, men ikke  $7 \times 2$ .

4. Denne værdi er tankens type. Ligesom de to tal, laves dette tilfældigt. Typen kan være et areal: "area", regnestykke: "equation" eller tal: "number". På den måde vil spilleren tilfældigt komme igennem alle former for præsentationer af et gangestykke.
5. Tankens x værdi. Denne værdi bestemmer x positionen af tanken. Da tanken kun bevæger sig i en retning, nemlig langs y-aksen, vil denne værdi være det samme gennem hele spillet. Den ændres altså ikke på noget tidspunkt. Der vil hele tiden være 3 tanks i spil. Når en tank destrueres dannes en ny, i samme vej som den destruerede tank. X-værdien er derfor bestemt ud fra antal tanks, hvilken vej tanken er i, og bredden på battlefieldet.



6. Tankens y værdi. Denne værdi bestemmer y positionen af tanken. Da tanken bevæger sig langs y-aksen, opdateres denne værdi efterhånden som spillet spilles, i forhold til tankens størrelse. Større tanks bevæger sig langsommere end små tanks.

Ved at have gemt disse 6 værdier vil tegningen af tanksene nemt kunne lade sig gøre. Alle tanksene tegnes først ved at deres omrids laves. Dette gøres med `context.rect(x,y,h,w)`, som tegner en rektangel med startposition i koordinatet  $(x,y)$  og med henholdsvis højde og bredde som  $h$  og  $w$ .  $x$  og  $y$  er gemt som 5. og 6. værdi i tanken, og højden og bredden hentes ved at tage værdi 2 og 3, som er tankens tal, og ganger disse med størrelsen af et felt. Hvis typen af tanken er "number" eller "equation", består tanken kun af dette rektangel, da resten af tanken så består af en tekst, som bliver vist som for eksempel "6" eller "2x3". Hvis tanken derimod er af typen "area", skal ingen tekst vises, men linjer tegnes som danner et gitter. Dette gøres ved at bruge `context.moveTo` og `context.lineTo`. `moveTo` benyttes i starten for at definere linjens start. Og `lineTo` bruges til at definere linjens slutpunkt. Igen ved at bruge tankens værdier, kan man nemt med 2 for loops lave gitteret: en til de vandrette linjer, og en til de horisontale linjer.

## DrawCalc

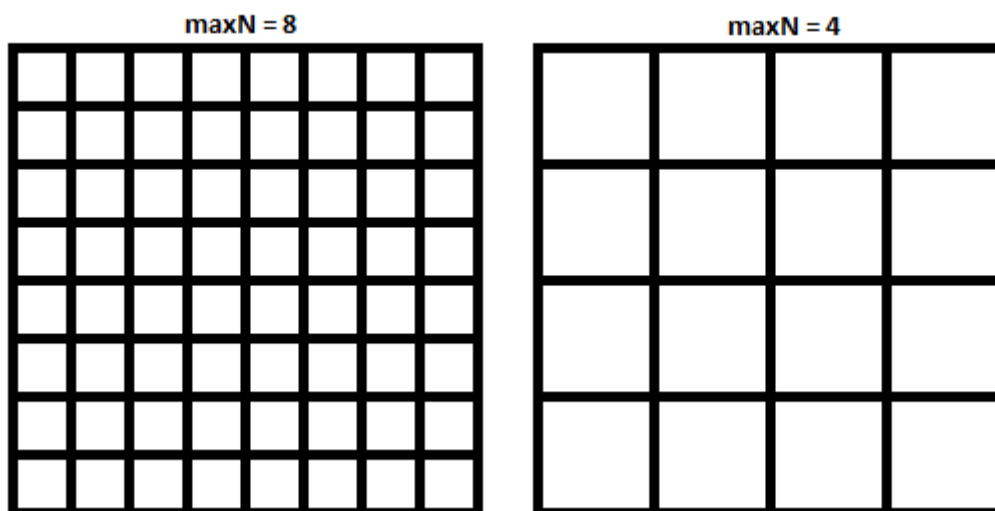
Lommeregneren består af 2 ting. Skærmen hvor det indtastede resultat vises, og så knapperne som bruges til at taste med. Skærmen er et rektangel, som med tekst viser hvad brugeren har indtastet.

Af knapper findes 0-9 samt en gangeknop og en clearknop. Talknapperne bruges til at taste det ønskede tal ind, gangeknappen bruges hvis man vil bruge et gangestykke, og clearknappen bruges til at starte forfra med et nyt resultat.

Alle 12 knapper tegnes ved at bruge 2 for loops, en til højden og en til bredden. Hver gang laves et rektangel med samme størrelse, men for hvert loop ændres positionen. For knapperne 1-9 tegnes teksten nemt ved at bruge værdierne som loopet kører over. For de øvrige knapper må teksten dannes enkeltvis.

## DrawGrid

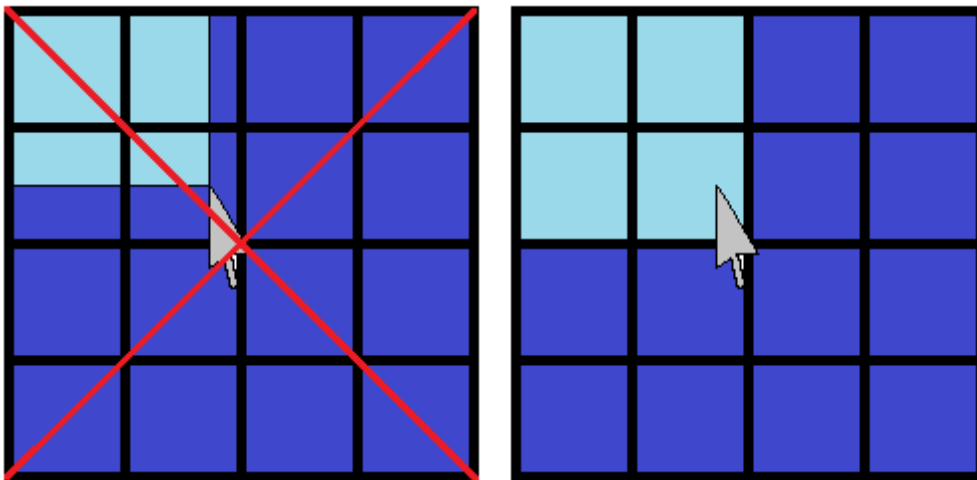
Området som bruges til at markere et område tegner først omridset af griddet. Dens, position, højde og bredde er fastsat fra start. Linjerne tegnes som tidligere forklaret ved brug af moveTo og.lineTo, samt 2 for loops. Afstanden mellem linjerne bliver sat i forhold til maxN. På den måde vil hele griddet blive brugt, og det ikke kun er en del af det.



Figur: Uanset hvad maxN er, vil griddet fylde det samme i spillet.

## DrawGridArea

DrawGridArea kaldes kun hvis musen holdes over griddet. I bund og grund er det et rektangel som bliver tegnet. Men for at få musen til at holde sig fast i skæringerne, bruges Math.ceil. På den måde vil musens koordinater rundes op, og fanges af den næste skæring.



Figur: Det markerede område vil fanges af næste skæring, så man ikke får decimaltal

### DrawBarrier

Barrieren som repræsenterer spillerens forsvar, tegnes med `context.rect`. Positionen er den samme, men højden ændres i forhold til hvor meget liv man har tilbage. Ved 100 liv har man den maksimale højde barriere, mens ved mindre liv, mindskes højden indtil den når 0, og spillet er slut.

### WriteScoreLife

Denne function sørger hele tiden for at spilleren score og liv bliver vist på skærmen. Ved at have to variabler: `life` og `score`, hentes disse hele tiden, så teksten er opdateret

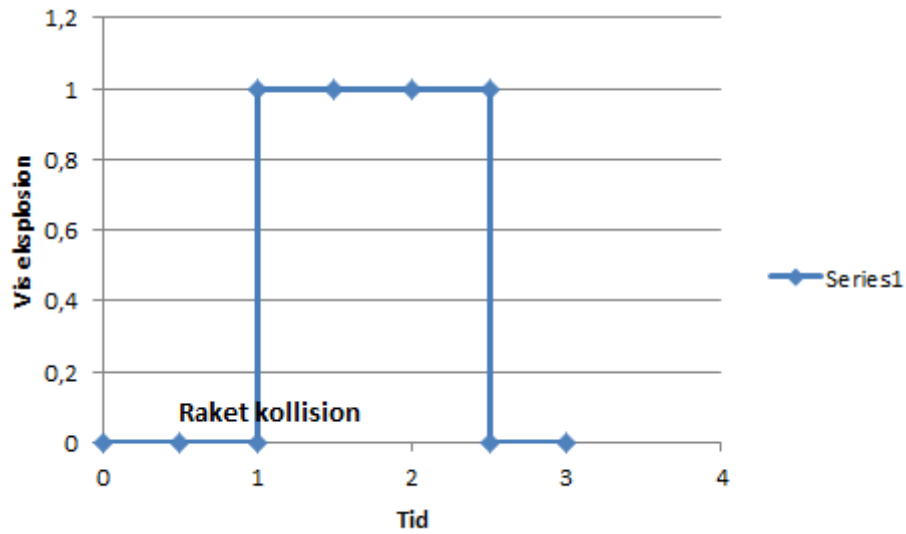
### DrawRocket

Denne funktion tegner raketten. Da raketten er, i modsætning til tanksene, en figur som ikke skifter form gennem spillet, bruges her et billede i stedet for at tegne rektangler. Selve metoden som henter billedet er ikke så svær, dette gøres nemt ved `context.drawImage(image,x,y)`. Men udover billedet tegnes mundingsild fra raketten, ved at bruge `context.arc`. Denne funktion tegner en cirkel. Ved at bruge et for loop, tegnes en masse cirkler tilfældige steder, som hermed giver en ildeffekt, men kun hvis raketten er skudt af sted.

### drawExplosion

Hver gang at raketten rammer noget, en tank eller enden af battlefieldet, sættes en timer. Denne timer sørger for at `drawExplosion` bliver kaldt. Hvis timeren er sat indenfor det seneste sekund, vil en eksplosion tegnes ved det sidst gemte eksplosionssted.





Figur: Hvordan eksplosion bliver vist i forhold til tiden efter kollision

Ligesom mundingsilden tegnes eksplosionen med for loop samt context.arc. For at vise om raketten har ramt rigtigt, forkert eller slet ikke ramt noget, tegnes eksplosionen med forskellige farver. Hvis farven er orange, har man ramt rigtigt. Hvis farven er grå, rammer man ikke noget, eller også er svaret forkert.

## Resultater og test

Hvad kan gøre spillet bedre?

3.B	2.C	4.C
En vejledning i spillet	Flottere grafik	Vejledning
Brug af tastatur til talindskrivning	Vejledning	Flere forskellige regnestykker
Valg af sværhedsgrad	Mere realistiske tanks	Hvis der skete flere ting i spillet
Valg af regneform (+ - * /)	Andet end gangestykker	Sværhedsgrad
Flottere grafik (tanks som ligner tanks)	Sværhedsgrad	Bedre grafik
Der skal ske noget nyt når man klarer sig godt		Multiplayer
Spillet skal blive sværere med tiden		Mini baner
Forskelligt terræn		Tastatur
		Flere tanks
		Hurtigere tanks
		Andet end gangestykker
		"At man selv var i tanken og man kigger ud gennem vinduet. For at skyde andre tanks skal man regne for at skyde"

## Hvorfor spiller du computerspil?

3.B	2.C	4.C
Man bliver inspireret til at tegne forskellige ting	Det er spændende	Det er sjovt
Det er sjovt	Det er sjovt	Jeg keder mig
Jeg keder mig	Fordi mine forældre siger det	For at spille med venner
		For at tiden går hurtigere
		Man skal tænke sig om

## Hvad var sjovt ved spillet?

3.B	2.C	4.C
At man skulle tænke	At man kunne skyde	Man blev snydt til at lave matematik
At man blev stresset og det var svært	At man skulle prøve sig frem	Man skulle arbejde sammen
At man skulle skynde sig	At man skulle noget nyt	Man skulle skynde sig
At man skulle skyde raketter		Flere muligheder at regne på
At man skulle bruge matematik		Man lærte noget
At tanksene kørte hurtigt		At det handlede om tanks
		Areal boksen
		Hvis man regnede forkert blev man stresset
		Alt
		Forskellige hastigheder
		Spillet var udfordrende

## Hvad var svært ved spillet?

3.B	2.C	4.C
At man skulle være hurtig	Regnestykkerne	De små regnestykker kørte hurtigt
De store tanks	At det var stressende	Svære regnestykker
At man skulle være koncentreret		Stressende
At finde ud af hvad man skulle		At forstå spillet
At man skulle regne		At man skulle skynde sig
		Når man svarede forkert, skulle man skynde sig
		At bruge musen

## Hvad var nemt ved spillet?

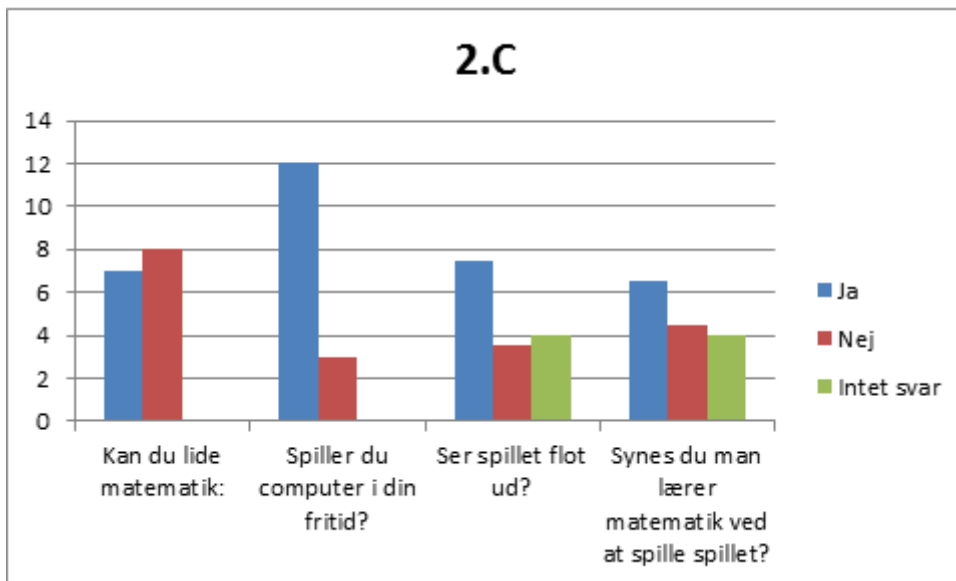
3.B	2.C	4.C
De små tanks	De små tanks	De små regnestykker
At tanksene var langsomme	Svære regnestykker	At spille spillet
At skyde	At skyde	

En ting som går igen blandt mange svarene, hvad angår og det var nemt eller svært, er at regnestykkerne var for svære eller for nemme. For nogle er det for svært, mens andre syntes det var for nemt. Dette kan ændres ved at implementere sværhedsgrad i spillet. På den måde vil spillet selv kunne justeres til den enkelte elev. Og alle aldersgrupper(fra 7-12 år) vil så kunne spille spillet.

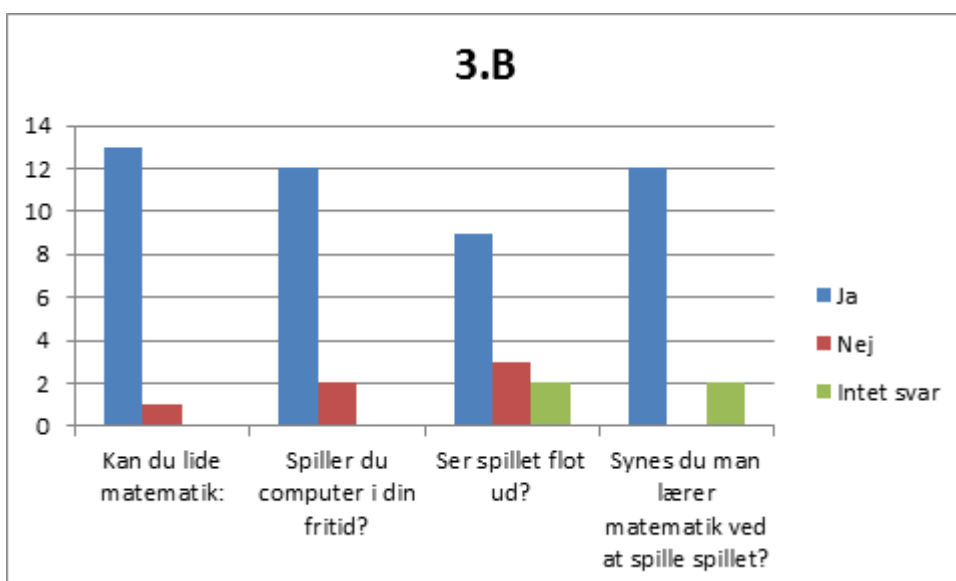
## Score:

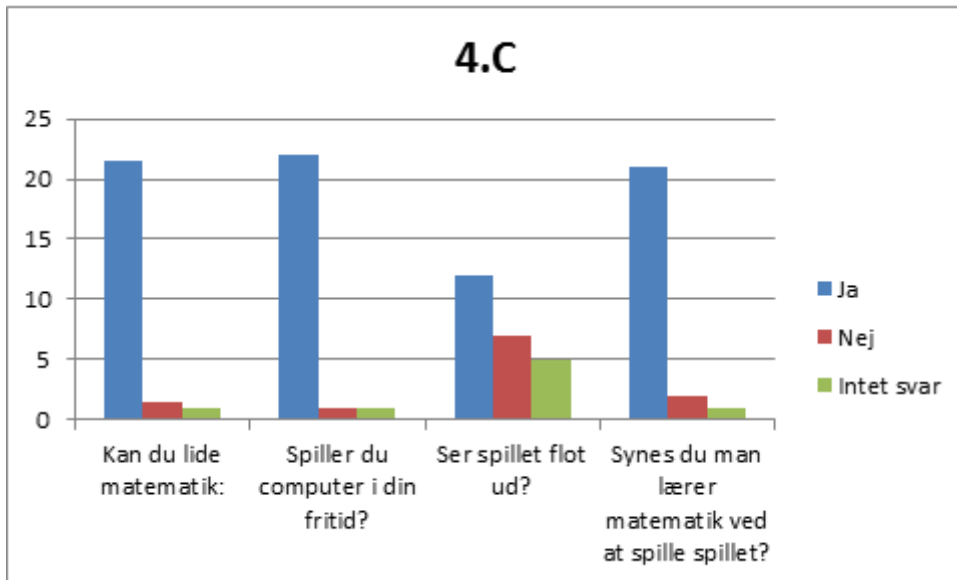
	3.B	2.C	4.C	Samlet
Antal	11	10	22	43
Score	84	61	166,5	311,5
Gennemsnit	7,636364	6,1	7,568182	7,244186047

Selvom man ikke kan sige meget ud fra at teste spillet i tre klasser, så ser det ud som at der er en tendens til spillet er tilfredsstillende. Man kan forestille sig at 2.C, har for svært ved spillet, men dette skyldes at der var samme sværhedsgrad for alle klasserne. Dette kan individualiseres, ved at sværhedsgraden ændres igennem spillet. Og på den måde vil spillet kunne gøres bedre. Både 3.B og 4.C har givet spillet nogenlunde god karakter, og dette kan skyldes de forstår spillet bedre, og har nemmere ved det.



På diagrammet over spørgsmålene, ses det samtidig at 2.C ikke er så glad for matematik i forhold til de to andre klasser. Dette kan også have betydning for hvad de syntes om spillet.





Både 3.B og 4.C har givet bedre karakter til spillet, end 2.C, og de er også betydelig gladere for matematik. Samtidig synes de også at man lærer matematik af spillet, som også kan have betydning for hvad man synes om spillet.

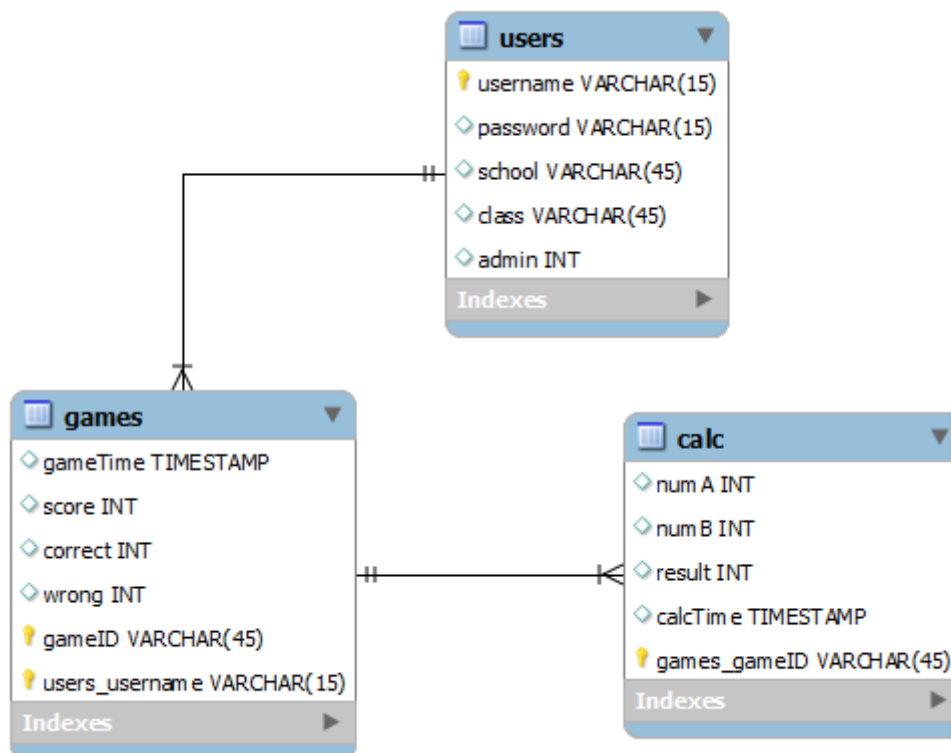
Alle klasserne er dog enige om at spillet godt kunne se bedre ud, men dette er også en af de største mangler i spillet.

## Diskussion

Spillet som er blevet lavet gennem dette projekt, er langt fra færdigt. Der vil altid være plads til forbedringer. Ved at teste spillet hos en folkeskole og bede eleverne svare på hvad der kunne gøres bedre ved spillet, er en masse ideer kommet frem.

## Forbedringer

- En vejledning er tilføjet hjemmesiden som et billede om hvilke taster man skal bruge, men hvordan spillet fungere skal helst være op til spilleren. Eleven skal selv finde ud af at et areal ikke kan destrueres med et areal. Man lærer hvad man gør, og ikke hvad man får fortalt.
  - o For at gøre det mere interessant kunne vejledningen blive vist som en tegneserie, eller video. Ved at for eksempel få en soldat ind i billedet som fortæller en historie og viser hvordan man styrer, kan man måske fange elevens interesse nemmere.
- Tastaturet er blevet integreret, så man nu kan bruge num-paden, hvis man har sådan en, til at indtaste tallene. Det er muligt at implementere sådan at man også kan bruge tastaturets tal knapper, det vil sige ikke num-pad. Men så skal eleven holde øje med en større del på tastaturet, og det kan gøre det sværere for eleven.
- En tydelig forbedring ved spillet, er grafikken, som også mange elever har nævnt. Hvis man stadig vil bruge canvas2D, vil pænere grafik kræve flere operationer, og det ville gøre spillet langsommere. Ved at benytte sig af en softwaregrænseflade, ville grafikken forbedres betydeligt, og stadig kunne følge med.
- En anden tydelig forbedring er en mere kompliceret database. I stedet for kun at gemme score, kunne man lave flere tabeller, som også kan gemme tid, antal rigtige og forkerte. På den måde kan man holde styr på hvert enkelt spil, og se elevens forbedringer. Et eksempel kunne være:



Figur: Opdatering af database

## Data ordbog over forbedret database

**Users:** Holder styr på brugerne som kan logge ind og spille spillet.

**Username:** Navnet som brugeren skal benytte sig af for at logge ind. Dette er unikt, og kan derfor kun bruges til én elev. Hvis der er flere som hedder Jesper, må de vælge navne som for eksempel Jesper123.

**Password:** Til hvert brugernavn hører der en kode til. Dette kan være en kombination af tal eller bogstaver som kun eleven kender til. På den måde vil ingen andre kunne logge ind på deres konto.

**School:** Definere hvilken skole brugeren hører til.

**Class:** Definere hvilket klassetrin brugeren går på. Denne værdi, sammen med school, gør det nemmere for læreren at finde ud af hvordan den enkelte klarer sig. Eller sammenlagt hvordan den samlede klasse klarer sig.

**Admin:** Gør det muligt at give rettigheder til brugeren. Læreren skal have rettigheder til at gå ind og se på elevernes præstationer. Han skal samtidig være i stand til at registrere nye brugere.



**Games:** Brugeren kan have mange spil registreret. Games holder styr på disse spil.

**gameTime:** Tidspunktet for hvornår eleven færdiggjorde et spil. Dette gør det muligt at se hvordan brugeren har udviklet sig igennem forløbet.

**Score:** Resultatet som eleven har opnået igennem spillet.

**Correct:** Antal rigtige svar i det givne spil.

**Wrong:** Antal forkerte svar i det givne spil.

**gameID:** En bruger kan spille mange spil. Hvert spil har en værdi som identificerer det givne spil.

**Calc:** Hvert spil indeholder et antal regnestykker bestående af gangestykker som brugeren skal løse.

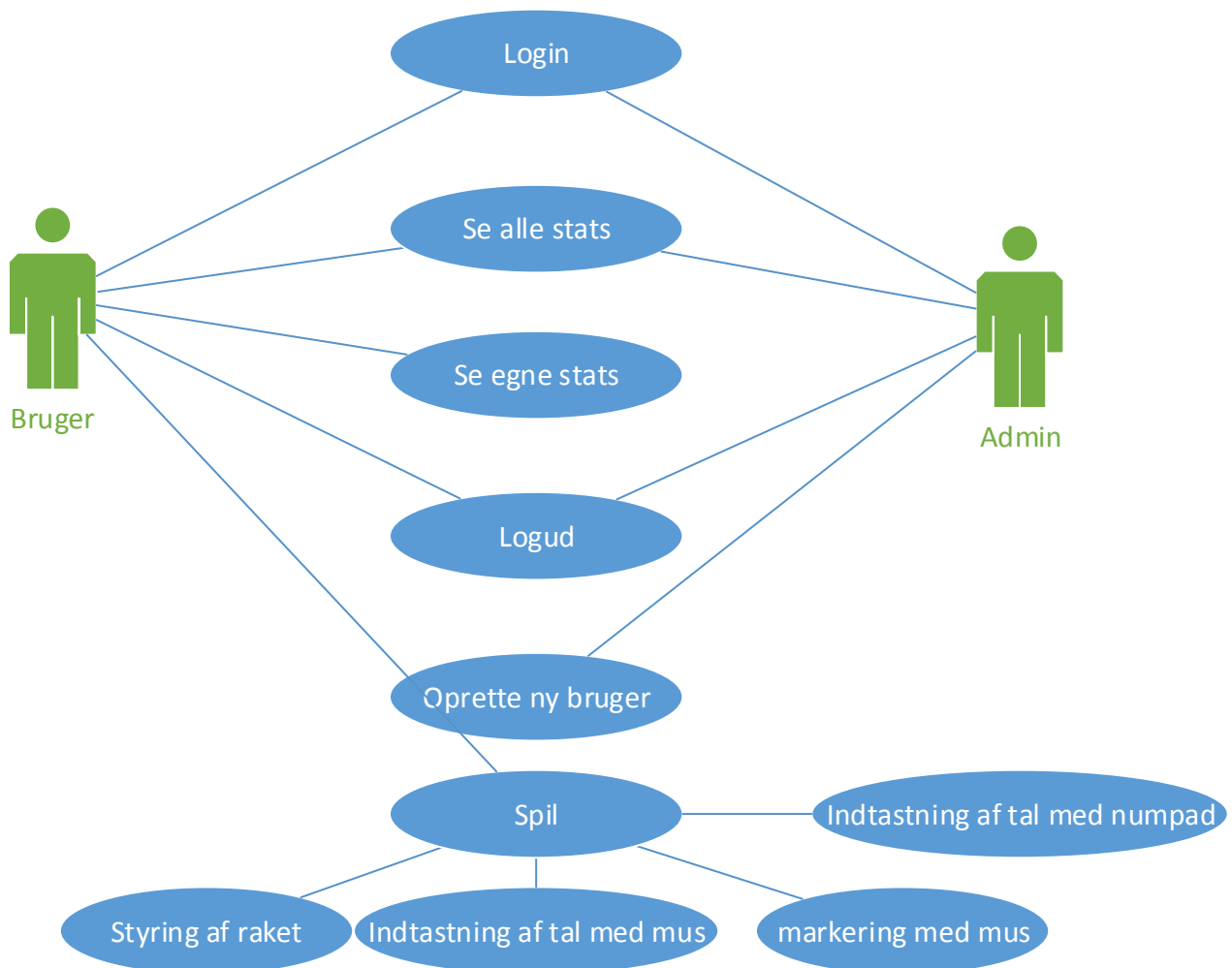
**numA:** Det først tal i gangstykket.

**numB:** Det først tal i gangstykket.

**Result:** Resultatet som eleven har prøvet at løse gangestykket med. På den kan læreren holde styr på hvilke tal eleven er gode til, eller hvilke tal eleven har svært ved.

**CalcTime:** Tidspunktet for hvornår eleven udregnede et regnestykke. Dette gør det muligt at se hvordan brugeren har udviklet sig igennem spillet.

## Forbedret Use cases



Ved forbedring af php filerne, kan ovennævnte use cases virkeliggøres, som indsætter rettigheder til en bruger, så man bliver admin. Admin skal kunne oprette nye brugere, samt gå ind og se brugerens resultater igennem spillene.

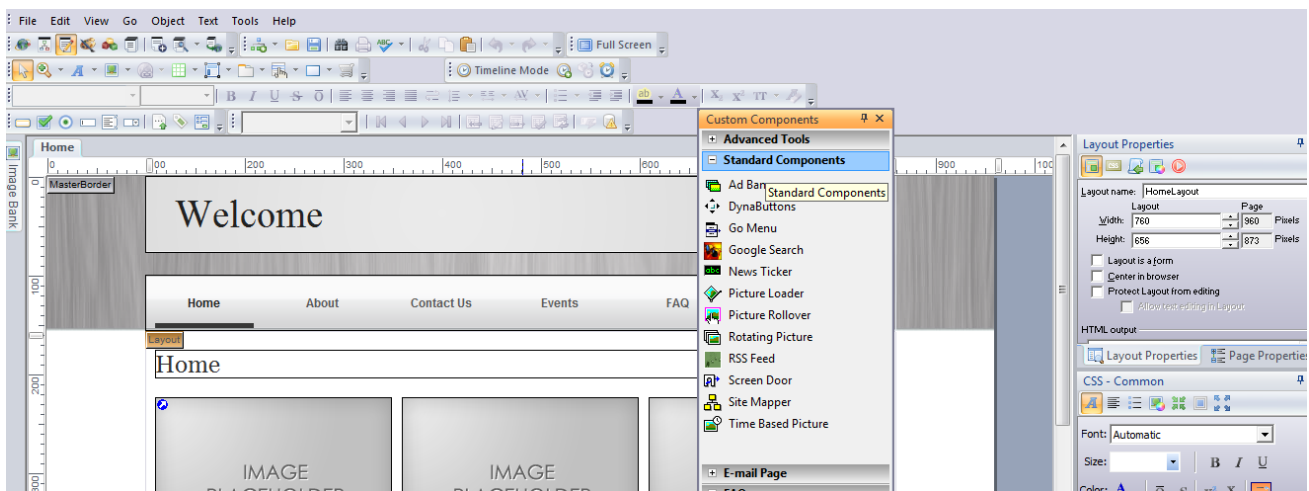
Derudover skal brugeren kunne følge deres egen score, så de kan se hvordan de klarer sig.

- Hvis man samtidig giver rettigheder til brugere, er det muligt for en lærer at oprette flere brugere, eller se hvordan en elev klarer sig.
- Sværhedsgraden kan ændres på den måde at tanksene kører hurtigere og hurtigere. Hver gang en tank destrueres med en raket, kan variabelen speed øges med en værdi. Man kan også implementere, sådan at regnestykkerne bliver sværere og sværere. Man starter måske med 2 og 3 tabellen, men som spillet kører, møder man også højere tabeller. Hvis man i databasen, også

gemmer hvilke værdier eleven har svært ved, kunne man sørge for at eleven får flere af den slags regnestykker.

- Spillet har fokuseret på gangestykker, men med lommeregneren, kan man simpelt integrere plus, minus og division. Programmet skal så vide at der er tale om en form end gange.
- En af de ting som kan gøre et spil sjove at spille er, at hvis man klarer sig godt belønnes man. Man kunne indføre en bonus, så for hver gang man svarer rigtigt, får man bonuspoint. Hvis man bliver ved med at svarer rigtigt, får man mere bonus. Hvis man så svarer forkert eller en tank når frem til barrieren, mister man sin bonus, og den nulstilles.
  - o Samtidig kunne der være en knap hvor man mister noget score men til gengæld kan man ødelægge en tank efter eget valg. På den måde sidder en elev ikke med et regnestykke som man bare ikke kan komme videre med.
- Spillet er meget ens formet på den måde at det er den samme grafik, som man kigger på hele tiden. Ved at man har baner, som skifter grafik, både terræn og maskiner, kan det samme spille føles som en ny oplevelse. Man kunne for eksempel være i rummet, hvor det var rumskibe som man skulle destruere.
- Børn har nemmere ved at spille et spil, hvis de kan konkurrere mod hinanden. Derfor ville det være smart hvis man fra databasen hentede score, tid mm ned til en hjemmeside, som så viste resultaterne ud for elevernes navne. På den måde kan man være på top ti, ikke kun blandt ens klassekammerater, men også blandt andre skoler.
- Lyde kan gøre spillet sjovere. Hvis man indsætter eksplosionslyde når raketten rammer noget, eller tastelyde ved lommeregneren, gøres spillet mere interessant. Baggrundslyd ville også være en god ide, som måske skulle passe til temaet.

- Tanksene kunne være i stand til at skyde. Enten kun for det visuelle skyld, eller også skulle man sørge for at raketten ikke blev ramt af skuddene.
- Udover at gøre spillet pænere, kan hjemmesiden også gøres pænere. Hvis man ikke vil gøre det manuelt, findes programmer som gør det lettere. Nogle eksempler er Adobe Dreamweaver og Netobjects Fusion, som er "drag and drop" programmer.



Figur: Netobjects Fusion: Webdesigner program til nem udvikling af hjemmesider

Selvom spillet er meget simpelt, indeholder det de elementer som det kræves for at lære matematik. Det indeholder samtidig en vis interaktion, som gør spilleren en betydelig del af spillet. Et spil behøves netop ikke at være stort og flot, for at være godt. Simple spil kan være ligeså gode som andre krævende spil.

## Konklusion

Efter at have fundet ud af hvordan børn skal lære matematik i dagens Danmark, er et spil blevet implementeret, som giver eleven mulighed for at se sammenhængen mellem gangestykker, tal og areal. Spillet er implementeret i JavaScript, og benytter sig ikke af nogen form for krævende software, så det er nemt tilgængeligt for elever. En hjemmeside, implementeret i HTML bruges som visning af spillet, samt login, men da spillet har været i centrum, er der tydelige forbedringer af hjemmesidens layout. Hjemmesiden arbejder sammen med en database, ved at bruge PHP.

Efter at spillet er blevet testet på en folkeskole af en 2.klasse, 3. klasse og en 4. klasse, har spillet vist sig at kunne bruges som oplæringsmateriale i folkeskolen. Eleverne mente af de lærte matematik af spillet, og med tilpassende sværhedsgrad, vil spillet kunne nå ud til flere klassetrin i folkeskolen.

Grafikken kan altid gøres bedre, men samtidig har det vist sig i spillets verden at selv de simpleste spil, har fået succes.

## Referencer

Hjemmesider:

<http://videnskab.dk/kultur-samfund/computerspil-bedre-end-matematiklaerer>

<http://www.ictgames.com/technowithflock.html>

<http://www.ictgames.com/abacusInteger.html>

<http://www.multiplication.com/games/play/dynamite-multiplication>

[http://uk.divertis.com/jeu/game1687\\_Pacgirl](http://uk.divertis.com/jeu/game1687_Pacgirl)

<http://www.feudmaster.dk/artikler/sla-et-wordfeud-snydeprogram/>

<http://www.onlineandroidtips.com/apps/scrabble.html>

<http://da.wikipedia.org/wiki/HTML> 23-06-2013

<http://www.wisegeek.com/what-are-the-pros-and-cons-of-php.htm>

<http://stackoverflow.com/questions/4812600/what-are-the-pros-cons-of-teaching-javascript-vs-python-as-a-first-programming>

<http://nerdfortress.com/2008/05/13/ruby-on-rails-pros-cons-lets-get-real/>

<http://da.wikipedia.org/wiki/ASP.NET> 23-06-2013

<http://answers.yahoo.com/question/index?qid=20080820120329AADdNbw>

[http://referencedesigner.com/tutorials/js/js\\_1.php](http://referencedesigner.com/tutorials/js/js_1.php)

[http://en.wikipedia.org/wiki/Programming\\_languages\\_used\\_in\\_most\\_popular\\_websites](http://en.wikipedia.org/wiki/Programming_languages_used_in_most_popular_websites) 24-06-2013

<http://www.mediacollege.com/internet/javascript/pros-cons.html>

<http://en.wikipedia.org/wiki/WebGL> 24-06-2013

<http://curvedcatgames.wordpress.com/2011/08/27/html5-canvas-vs-webgl/>

<http://html5doctor.com/an-introduction-to-the-canvas-2d-api/>

<http://curvedcatgames.wordpress.com/2011/08/27/html5-canvas-vs-webgl/>

## Bilag

### Hvordan man spiller spillet?

Spillet ligger på linket:

[matskole.compute.dtu.dk/bscf134](https://matskole.compute.dtu.dk/bscf134)

Man kan prøve spillet ved at benytte sig af brugernavnet: user og kodeordet: 123

Når man indtaster adressen, kommer man frem til log-in siden:



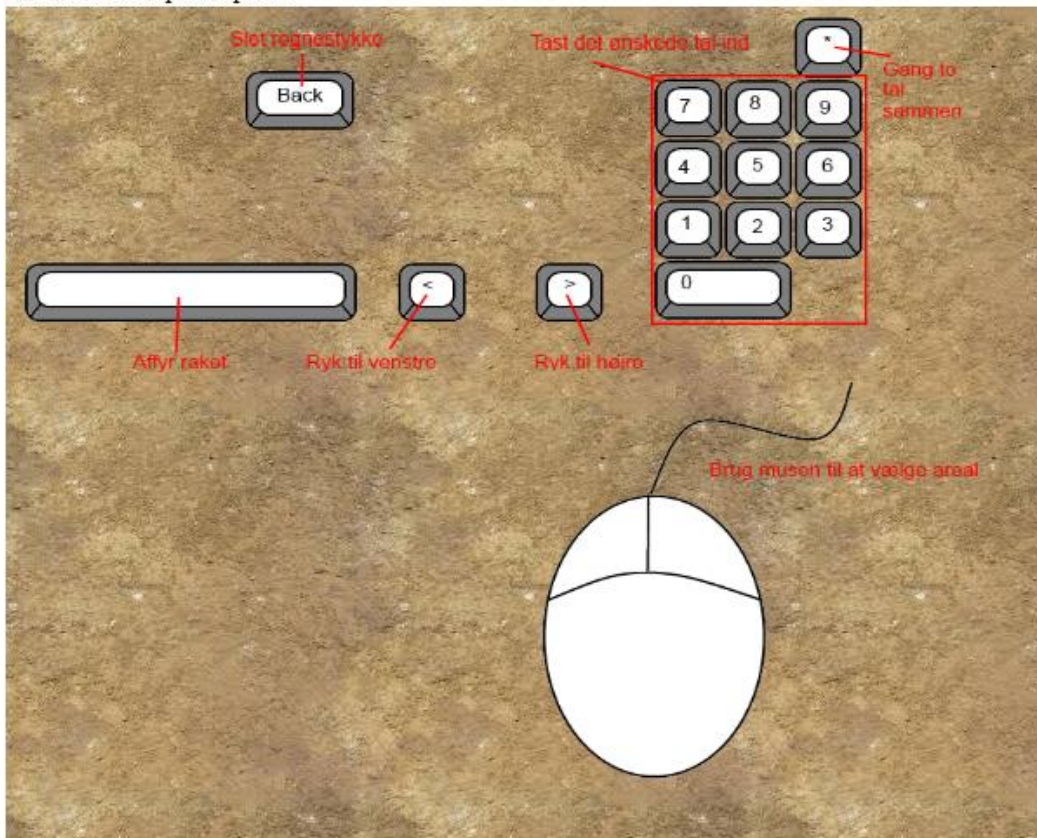
Brugernavn:

Kodeord:

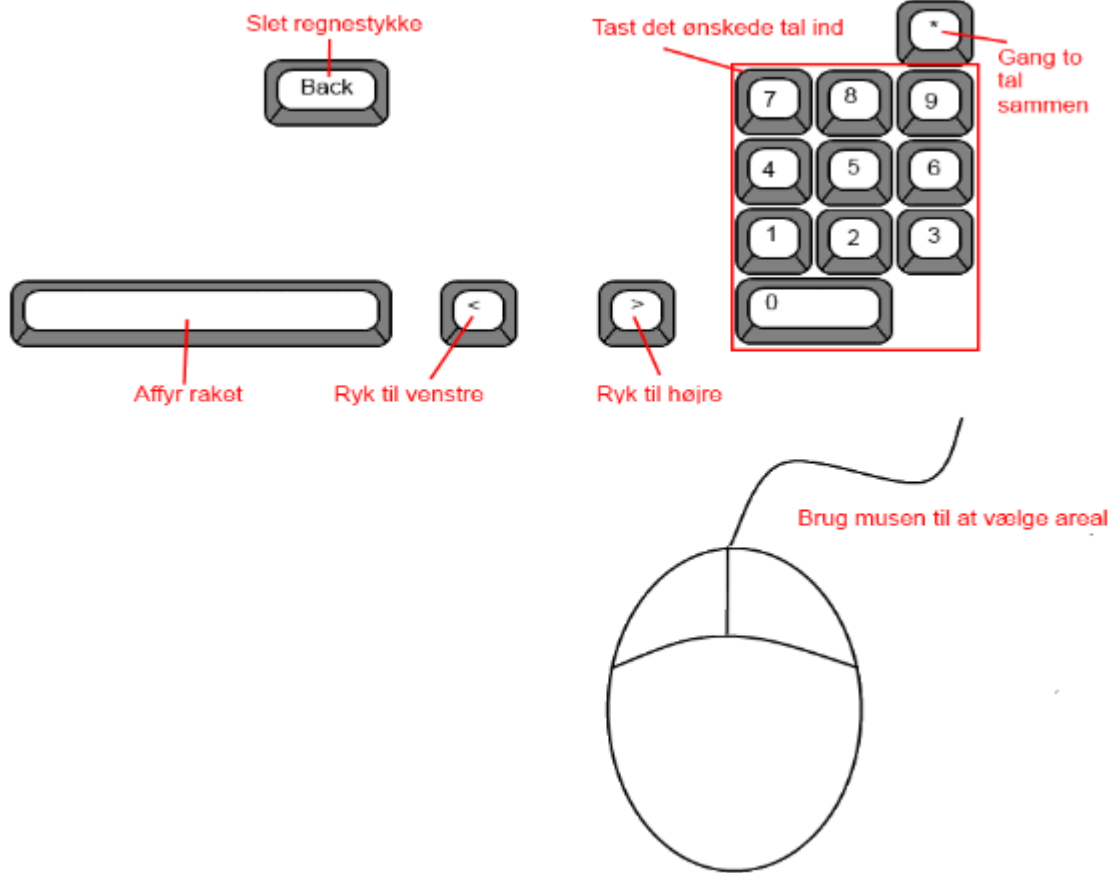
Indsend

Ens brugernavn og kodeord indtastes, efterfulgt af ENTER, og hvis man findes i databasen, vil man blive sendt videre til spillet.

Hvordan man spiller spillet:



Spillet styres med tasterne som vist:





Hvis man ikke er bruger, er det ikke muligt at spille spillet. Ved korrekt brugernavn og kodeord sendes man til siden:

Logout

Man kan til enhver tid logge ud ved at trykke på denne knap. Der gøres opmærksom på at scoren ikke vil blive gemt hvis man logger ud.

5x3

4

Tanksene som kommer kørende skal destrueres med det rigtige regnestykke.

0

1 2 3  
4 5 6  
7 8 9  
0 X C

Lommeregneren bruges til at indtaste det ønskede regnestykke. Både tal eller gangestykker er mulig.

Barrieren som markerer livet. Hvis man når 0, vil spillet være færdigt.

Man kan definere et areal, ved at holde musen over det ønskede areal.

Life = 100

Score = 0

Rakette

Raketten affyres med SPACE. Den kan både før og efter affyring, styres med piletasterne.

Score:  Indsend

Det er også muligt at bruge numpad til at indtaste tallene, samt GANGE. Samtidig kan man slette alt indtastet, ved at trykke BACKSPACE.

Når spillet er slut, vil denne besked blive vist, med den opnåede score:



Man kan nu kun komme videre ved at trykke logud, eller indsend score. Ved logud, vil man blive sendt tilbage til startside.

Ved indsend, sendes man til siden:



Din score blev gemt, hvad vil du nu?

Game

Logout

Man har nu to valgmuligheder:

1. Game, som starter et nyt spil. Man er altså stadig logget ind, og man kan spille videre.
2. Logout, sender brugeren videre til startside. For at spille igen, skal man logge ind.