Technical University of Denmark

Mads Lundt, s103439

Spil til undervisning

Et computerspil til matematikundervisning i folkeskolen

Software teknologi projekt Juni 2013

DTU Compute
Institut for Matematik og Computer Science

Mads Lundt, s103439

Spil til undervisning

Et computerspil til matematikundervisning i folkeskolen

Software teknologi projekt Juni 2013

Spil til undervisning,Et computerspil til matematikundervisning i folkeskolen

Rapporten er udarbejdet af
Mads Lundt, s103439

Vejledere
Jeppe Revall Frisvad
Niels Jørgen Christensen

Udgivelsesdato:	Juni, 2013
Klasse:	1 (public)
Udgave:	First

| Hjemmeside: | matskole.compute.dtu.dk/bscf132 |

Institut for Matematik og Computer Science
Danmarks Tekniske Universitet
Matematiktorvet, bygning 303 B
2800 Kgs. Lyngby
Danmark

www.compute.dtu.dk
Tel: (+45) 45 25 30 31
Fax: (+45) 45 88 13 99
E-mail: compute@compute.dtu.dk

Resumé

In today's elementary school pupils are having problems relating math in to their daily life. The pupils are looking for patterns when they get a math problem and if they can't find the pattern they can't solve the problem. If they don't figure this out they have problems solving math problems when they get older and are going to get some tough years in elementary school. To help pupils in elementary school a computer game is needed. The game is going to challenge the pupils into breaking these patterns so they can solve the problems without matching with their own pattern. The pupils' interest for the game should be in focus so the pupils are excited to play with math.

This project shows how the math game has been developed from analyzing pupils, game development to the finishing product. The math game are focusing on the number line. Some numbers are falling from above and should be placed on the number line. The game has been made to be played on a web page so a web page should be made. The web page has been made with login so the pupils have their own account with their highscore. The teachers then have the option to get a view of the pupils and how they are doing. This is to help the pupils as the teacher can get an overview of how the pupils are doing and if they need further challenges or not.

The game has been tested in two school classes with younger pupils to see if it has any math game potential.

Indhold

Figurer

Tabeller

Listings

1

Introduktion

I dag har mange børn i folkeskolerne problemer med at reflektere matematikken til det virkelig liv. Dette gør det svært for dem, at klare simple udfordringer, som at regne matematiske opgaver i dagligdagen. Problemet er, at børn har det med at se et bestemt mønstre og hvis de ikke kan genkende et mønster, så får de problemer med at regne den ud. For at hjælpe børnene og få dem til at kunne reflektere matematikken til den virkelige verden, kan dette læres ved hjælp af computerspil. Disse computerspil findes allerede på internettet, men det er svært at finde nogen som børnene kan lære af og udvikle sig med. For det andet er det vigtigt at gøre spillet spændende, så børnene får lyst til at blive ved med, at spille dem.

Problemet for dette projekt er at få analyseret unge elevers interesse i matematikken. Det er vigtigt at få lavet et spil som eleverne synes om og får deres interesse.

Dette er interessant da unge elever har det med at have et mønster som de husker på og hvis der sker noget nyt, så har de det med ikke at kunne fuldføre opgaven. Det er derfor vigtigt at træne dem til at bruge matematikken i andre tilfælde.

Det er svært at analysere elevers indgang til matematikken da de alle sammen er så forskellige. Samtidig med det, så skal der også gøres noget spændende ud af spillet, så eleverne får lysten til at spille det.

Der findes allerede spil derude til elever, men mange af dem formår ikke at opretholde elevernes interesse og så går det tabt. De fleste spil er begrænset hvor meget eleverne kan lære og så går udviklingen i stå.

Projektet fokuserer på at opretholde spændingen i spillet, samt fleksibilitet med spil og elever.

2

Kravspecifikation

Krav til produktet er listet før udvikling af det, da det er vigtigt at der er nogle rammer for produktet. Produktet ønskes som en hjemmeside som skal stå for adminstrationen og forbinde brugerne til spillet. Spille skal altså spilles gennem hjemmesiden ved at logge ind på hjemmesiden. Produktet kan opdeles i tre dele spillet, hjemmeside og database.

Spillet

- Virker i de fleste moderne internet browsers.
- Flydende grafik med animationer uden at det hakker.
- Login-system til hver bruger for at gemme spilhistorikker med mere.
- Mulighed for flere baner.
- Fleksibelt og mulighed for at videreudvikle funktionaliteter.

Hjemmeside

- Pænt og simpelt design.
- Responsive design så det passer på alle opløsninger, selv små enheder som mobiltelefoner.
- Forskellig adgang efter om man er lære eller elev.

Hjemmesiden skal samtidig indeholde udover spillet.

- Velkomstside (forside).
- Vælg bane man vil spille.
- Se spilhistorikker.
- Vurdering af spil.
- Oprettelse af nye baner.

Database

- Stor plads til indhold af brugerdata og tidligere spil.

3 Analyse

Kan det overhoved betale sig at benytte spil til at forbedre elevernes evner i matematikken?

Det er vist i en undersøgelse, at elever, ved hjælp af computerspil, kan lære mere af at spille spil end tidligere antaget. Dette skyldes blandt andet at spil får eleverne til at engagere sig mere i det og da det er vist at elever lærer mere ved at være engageret, så er spil en rigtig god mulighed for dem. Med dette sagt så betyder det ikke at børn kun skal spille spil og stoppe med undervisningen. Det er vigtigt med noget afveksling for eleverne og der kan spil gå hen og blive et vigtigt redskab i læren om matematikken[1]. Spilundervisningen, se reference [3], har også skrevet om nogle gode grunde til at inddrage computerspil i undervisningen, på baggrund af lærere, forskere og elever. De nævner blandt andet at det som beskrevet før øger elevernes interesse og derved øger elevernes motivation til faget. De elever der har det svært bogligt kan få en stor fordel ved computerspil da det øger de enkelte elevers selvværd da de går hen og bliver gode til faget. Spillene kan også have den fordel, at det er muligt at ændre sværhedsgrad for de forskellige elever og derved tilpasse den til de enkelte.

Ifølge Erik, se reference [1], så laver elever et mønster for at regne et stykke. Hvis de så ikke kan overføre dette mønster til et nyt stykke, så har de svært ved at løse den. Det er derfor vigtigt at eleverne bliver sat ud for nogle udfordringer og bryder dette mønster for så at lære at gøre det på nye måder.

3.1 Spilkoncepter

Spillet skal egne sig som undervisningsmateriale og henvende sig til unge elever. Spilkonceptet skal få elevernes opmærksomhed ved at relatere til noget de holder af og kender til. Det er vigtigt at spilkonceptet udfordrer eleverne og gør at de bliver engagerede i det. Nogle af de spilkoncepter der er udtænkt er og diskuteret med en folkeskolelære som har kendskab til unge elevers interesser og evner [1].

- Tallinjen
- Arealberegning
- Positionssystem

[1] Se reference [2]

Tallinjen foregår ved at der er faldende talbobler med plus og minus tal, som skal placeres korrekt på tallinjen. Det skal være muligt at justere på tallinjen så den kan blive højere eller lavere, så flere tal kan komme ned på tallinjen. I løbet af spillet skal der komme nogle bonustal som eleverne skal ramme for at få nogle bonusser. Dette er til for at spillet ikke skal blive ensformig og giver nogle udfordringer til eleverne.

Arealberegning foregår ved at man skal regne på eller tegne firkanter. Gangestykker kan også forekomme som skal tegnes som firkanter ved at man skal fylde nogle lastbiler op ved hjælp af arealberegning og regnestykker.

Positionssystem foregår ved at nogle faldende talbobler skal sættes ned i beholder af 1'ere, 10'ere, 100'ere og så videre. Det går så ud på at alle beholderne skal fyldes helt op for at klare banen. Talboblerne kan splittes op så de passer ned i flere beholdere.

3.2 Tidligere spil

Der findes i forvejen spil på markedet som børn kan finde interessant. Alle disse spil er dog ikke altid egnet til undervisning og der skal en dybere søgning efter spil til undervisning igang. Ved at sortere i disse spil er der fundet nogle spil som egner sig til undervisning.

- Pinatafever - http://www.mangahigh.com/en_gb/games/pinatafever
- Fruit shoot number line - http://www.sheppardsoftware.com/mathgames/earlymath/fruit_shoot_NumberLine.htm

Pinatafever som er et spil med tallinjen. Spillet går ud på at man skal samle nogle ting op på tallinjen og for at gøre dette så skal man regne sig frem til de forskellige tal på tallinjen. Spillet ser flot ud og kan godt gå hen og blive spændende for eleverne, men er udviklet i Flash (se afsnit 3.3).

Fruit shoot number line er igen et spil med tallinjen, men anderledes end Pinatafever. Spillet går ud på at skyde nogle stykker frugter ned som flyver rundt på skærmen. Frugterne har nogle regnestykker skrevet på dem og det skal matche med tallinjen. Tallinjen bevæger sig efterhånden som man skyder frugterne ned, men igen så er spillet udviklet i Flash. Spillet er meget ensformig i længden og giver ikke brugerne mulighed for at prøve nye ting af.

Disse spil er begge egnet til elever, men kan være svært for lære og elever at se hvordan de klarer sig gennem spillene. Samtidig er begge disse spil udviklet i Flash, som kan gå hen og blive et problem.

3.3 Spiludvikling - Game engines

Til udvikling af spillet kræver det en game engine. En game engine er til for at udvikle spillet, da den sørger for at rendere spillet i 2D- eller 3D grafik, samtidig

med at den indeholder nogle funktionaliter til spillet. Der findes mange game engines til de forskellige behov, nogle af disse er:

- Unity3D
- Silverlight
- Flash
- HTML5 med og uden WebGL

De forskellige game engines har deres fordele og ulemper. Unity3D var der erfaring med da det er prøvet før og har rig mulighed for at udvikle til mange enheder udover internet browseren. Unity3D har også muligheden for at udvikles med UnityScript[2], C#, med mere. Der findes i forvejen rigtig mange funktionaliteter i Unity3D, hvilket gør det oplagt at lave spil i. Unity3D kan benyttes til både at lave 2D- samt 3D grafik. Den ene ulempe ved Unity3D er at den kræver et plugin i internet browseren før spillet kan spilles. Dette plugin kræver ikke meget tid til installation, men kan være et irritationsmoment samt eleverne ikke ved hvad de skal gøres og lukker spillet. Nogle skoler kræver adminstrator adgang for at installere noget på computeren, som kan forårsage, at spillet ikke kan spilles.

Silverlight er ved at dø ud og er begrænset hvor godt det kører på andre klienter end Windows. Der bliver ikke udviklet mere på Silverlight og kræver at Silverlight er installeret. Det er begrænset hvor mange enheder der understøtter Silverlight, hvilket er en ulempe ved videreudvikling. Det udvikles med C#, som er prøvet før og det kan dog være en fordel.

Flash er ligesom Silverlight ved at dø ud, da der ikke bliver udviklet mere på det andet end opdateringer til diverse sikkerhedsbrister. Adobe, som er udvikler af Flash, har også udgivet værktøjer som kan lave Flash til HTML5, som kan tyde på at Flash er et dødt projekt[3]. Flash kræver derudover også et plugin i internet browseren før det kan fungere.

HTML5 har den fordel at det kan køre i de fleste nyere internet browsers uden et plugin. Et framework kan blive nødvendigt for at få alle funktionaliteterne med i spillet. HTML5 understøtter heller ikke hardware acceleration, som gør at spillets grafik kan trækkes fra grafikkortet. Det vil altså kræve en processor som kan trække spillet. HTML5 kan kan også laves med WebGL som er et JavaScript API til at rendere 2D- og 3D grafik. I WebGL er hardware acceleration understøttet, som gør at der kan laves mere krævende spil, da det nu kan trækkes på grafikkortet. WebGL er understøttet i de fleste nyere internet browser fra start af, dog ikke i Internet Explorer, hvor det kræver installation af et plugin.

Ifølge kravspecifikationen så er nogle krav til spiludviklingen og valget af game engine, at det skal kunne køre i de moderne internet browsers i dag. Det skal samtidig også være et spil som børn finder spændende, da det er vist at det er nemmere at tage noget til sig hvis man synes om det. Det skal være muligt at spil kan indgå i undervisningen.

[2] UnityScript: JavaScript med Unity framework.
[3] http://www.adobe.com/dk/products/flash/flash-to-html5.html

	Unity3D	Silverlight	Flash	HTML5	WebGL
Kræver plugin	Ja	Ja	Ja	Nej	Nej*
Cross-platform	Windows Mac Linux	Windows Mac	Windows Mac Linux	Windows Mac Linux	Windows Mac Linux
Forsat udvikling	Ja	Nej**	Nej**	Ja	Ja
Programmerings-sprog	UnityScript C# Boo	C#	JSFL	HTML5	JavaScript

Tabel 3.1: Oversigt over de forskellige platforme.

** Kræver plugin til Internet Explorer.*
*** Mindre opdateringer til produktet, men ikke større udvikling.*

3.4 Spilfunktioner

Der skal laves et matematik spil med et spilkoncept (nævnt i afsnit 3.1) som eleverne finder interessant og engagerer sig i. Dette kræver at der bliver gjort noget spændende ud af spillet, som børnene finder interessant. Elever vil have at der sker noget i spillet som er spændende og viser hvordan de klarer sig. Til dette kræver det nogle animationer i spillet, som eleverne kan følge med i jo længere de når. Alt dette skal til for at engagere eleverne til at blive bedre til spillet og matematikken. Samtidig er det vigtigt at eleverne får feedback om det er rigtigt eller forkert ud for deres valg.

3.5 Hjemmeside

Der skal laves en hjemmeside til at køre spillet og vise spilhistorikker. Hjemmesiden kan laves fra bunden, men kræver en del viden om sikkerhed for blandt andet login-systemet. Samtidig er det et krav at det skal være nemt for lærere og elever at bestyre hjemmesiden. Et framework eller CMS[4] kan være at foretrække for at gøre det nemmere at udvikle hjemmesiden, da der allerede er taget højde for sikkerheden og brugerpanelet. Der findes forskellige CMS og frameworks på markedet i dag, nogle af de populære er:

- WordPress (CMS)
- FuelPhp (CMF)
- Drupal (CMS)
- Joomla (CMS)

[4] Content Management System - Et stykke software til at organisere samarbejdet på hjemmesiden

WordPress er oprindelig lavet til blogindlæg, men grundet et stort udvalg af plugins og de mange opdateringer kan det bruges til næsten hvad som helst. WordPress er blevet utrolig populært og der er meget hjælp at hente rundt omkring på internettet samt dokumentationen, som er udviklet af WordPress, er beskrevet med gode detaljer og eksempler. WordPress er kendt som et CMS system, men fungerer også som et framework. WordPress er open-source og fungerer med MySQL database (se mere under 3.7). Det at WordPress er open-source gør også at mange brugere er med til at forbedre og udvikle på systemet, hvilket også kan ses på udvalget af plugins. Udover plugins findes der også en masse temaer og widgets til systemet som er frit tilgængeligt.

FuelPhp er ikke ligesom WordPress et CMS, men nærmere et framework. Da det ikke er et CMS, så kræver det er brugerpanelet selv bliver lavet. Der er ikke kendskab til FuelPhp, men dokumentation er ligesom WordPress rigtig god.

Drupal er et populært CMS og bruges blandt andet af TV2.dk. Det kan benyttes af meget andet end MySQL, som kan være en fordel hvis der skal bruges et andet databasesprog. Drupal minder meget om WordPress med at det er open-source og udvikles af mange brugere, samt et stort udvalg af plugins. Der er ikke arbejdet med Drupal før, men indeholder en rigtig god dokumentation med online support i form af chat.

Joomla som også er et populært CMS system, men har i tidligere versioner fået kritik på deres rodet kode. Joomla benytter sig lige som de andre af MVC[5] og har mulighed for at benytte sig af MySQL og andre databasesprog. Joomla er ikke prøvet før, men indeholder som de andre rigtig god dokumentation og et forum til at finde hjælp fra andre brugere.

	WordPress	**FuelPHP**	**Drupal**	**Joomla**
Licens	GNU GPLv2	MIT	GNU GPLv2	GNU GPL
Database understøttelse	MySQL	MySQL	MySQL MariaDB PostgreSQL SQLite	MySQL MS SQL PostgreSQL

Tabel 3.2: Oversigt over frameworks og CMS systemer

3.6 Interaktion mellem klient- og server-side

Spillet skal køre fra klientens side, men skal kunne have mulighed for at kommunikere med server. Kommunikationen til serveren skal ske når spillet køres, for, at modtage bane information og sende spilhistorikker. For at gøre dette muligt findes der forskellige metoder som AJAX og WebSockets.

AJAX er hvor klienten åbner en ny side i baggrunden for så at indlæse eller sende information til serveren. Dette foregår uden at brugeren lægger mærke til det. En anden mulighed er WebSockets som kan sende eller modtage beskeder fra klienten. Det vil altså sige at serveren står for alt det krævende hvor i mod i AJAX sørger

[5] Model-View-Controller

klienten for at lave det hele før det kommer op på serveren.

Ved at sende eller modtage meget data på kort tid kan det være en fordel at vælge WebSockets, da AJAX skal åbne en ny side i baggrunden og lukke den, hver gang den skal sende eller modtage. WebSockets kræver dog derimod at et stykke software bliver sat op og kørt på serveren. Der er ikke arbejdet med WebSockets før, hvor imod AJAX er prøvet før og har kendskab dertil.

3.7 Database

Til både spillet og hjemmesiden kræver det en database med data omkring de tidligere spil, de forskellige baner og alle brugerne. Der er forskellige databasesprog, men den mest brugte er MySQL, som benytter relationer i databasen. MySQL er prøvet før og har den fordel, da det er så populært, at der er en stor bruger flok at henvende sig til hvis der ønskes hjælp. Det er samtidig robust og gennemprøvet, og det er næsten uundværligt ikke at have relationer i databasen i dag. De fleste CMS på markedet i dag er allerede lavet klar til MySQL, se tabel 3.2.

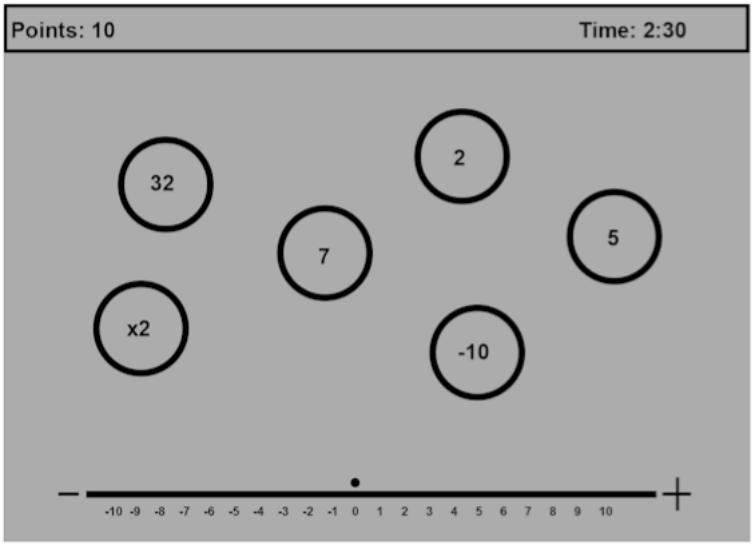
En anden mulighed end MySQL er NoSQL som benytter sig af objekter i stedet for relationer i databasen. Fordelen ved NoSQL er at det i nogle tilfælde kan performe bedre end MySQL. Det kan altså være en fordel at trække objekter ud end at trække data ud på kryds og tværs af tabellerne som i MySQL. Ulempen er at det kun er prøvet meget lidt før uden gode resultater og kan være svært at implementere i CMS systemer. De følgende CMS systemer der er beskrevet i afsnit 3.5 er ikke lavet til NoSQL.

4 Design

4.1 Spillet

Spillet skal udvikles med fokus på tallinjen og nogle animationer som gør spillet spændende. Talbobler kommer faldende ned fra toppen og skal sættes ind på tallinjen inden de rammer bunden. Når en talbobbel sættes korrekt ind på tallinjen stiger brugerens byggepoint og tiden begynder at gå. Hvis man ikke når at bygge sine point inden tiden er gået, mister man de point. Byggepointene stiger hurtigere jo flere talbobler man får rigtig inden man bygger, med den risiko at man kan miste alle sine byggepoint ved at sætte en talbobbel forkert eller være for langsom om at bygge. Af de point man får bygget bliver en bro bygget i baggrunden, så jo flere point man får bygget jo mere af broen bliver bygget. Ideen med at få en bro bygget er, at en bil skal køre over den uden at falde i vandet. Ved at sætte en talbobbel forkert kører bilen hurtigere eller påbegynder kørslen tidligere og har derved lettere ved at falde i vandet. Hvis bilen falder i vandet er spillet tabt og en platform skal sejle ud for at redde bilen op af vandet, så der kan prøves igen. Der er i starten af spillet en nedtælling på hvornår bilen starter med at køre. Bilen kan stoppe undervejs eller blive forsinket med at starte ved at brugeren rammer nogle bonustal på tallinjen. Når bilen er kommet sikkert over på den anden side, så er banen gennemført. Det er vigtigt at spillet opfylder kravene i afsnit 2 og testes for de use cases der er lavet, se appendiks A.

For at få et henblik på hvordan spillet skal se ud, er der lavet et mockup, se figur 4.1. Det er et simpelt design i en tidlig fase, som kun indeholder talbobler og tallinjen. Ideen med dette mockup er at få en ide om hvordan spillet skal komme til at se ud og skal hjælpe med at udvikle spillet i starten.



Figur 4.1: Mockup af spillet.

Mockup viser at talboblerne komme faldende ned med en hastighed og kan trækkes ned på tallinjen ved hjælp af musen. Tallinjen kan justeres ved at klikke på + eller - ved siden af tallinjen.

4.1.1 Game design

Det er vigtigt at få styr på game design, som omhandler hvordan spillet spilles og hvilke regler der er. Denne liste er udarbejdet sammen med folkeskolelære Erik, se reference [1].

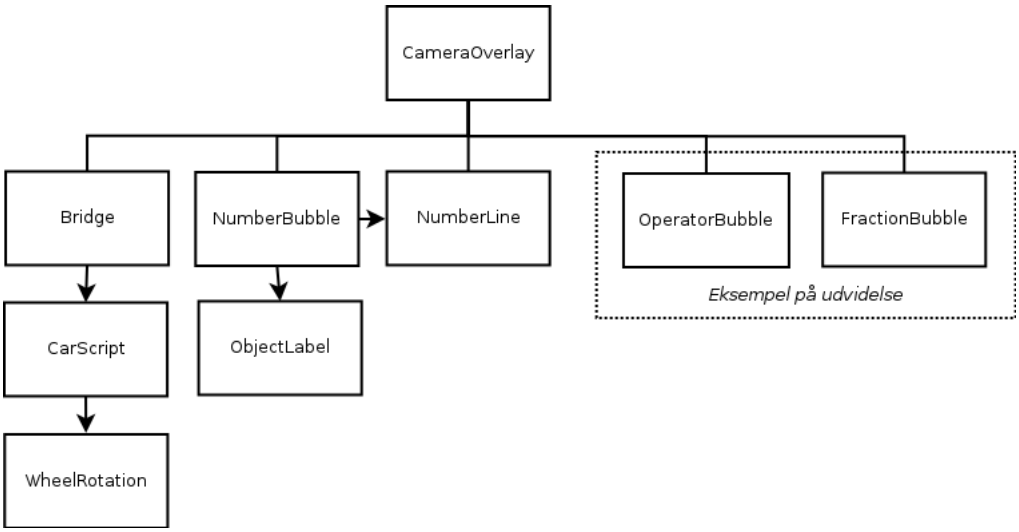
- 100 % af max-point ved at ramme rigtigt på tallinjen.
- 20 % af max-point ved at ramme næsten rigtig på tallinjen (+/- 1 fra det rigtige).
- Ved at ramme et tal som et bonus tal går op i, så får man en bonus. Bonus gør at bilen bremser og venter med at køre i et stykke tid.
- Ved ikke at nå et tal inden det falder ned, mister man muligheden for at bruge den talbobbel. Talboblen bliver ødelagt og erstattes af en ny.
- Ved at placere et tal forkert på tallinjen, får man en straf. Denne straf er at bilen kører hurtigere eller påbegynder sin kørsel tidligere.
- Når bilen er kommet uskadt over på den anden side, vinder man spillet.
- Hvis bilen falder ned i vandet undervejs, taber man spillet.
- Når man bygger/indløser sine bygge-point, bygges der på broen hvis man har point nok.
- Når man ikke at indløse sine bygge-point, mister man alle sine bygge-point. Det samme hvis man undervejs i byggeprocessen sætter tal forkert på tallinjen.

Da der skal være flere baner og brugeren selv skal have mulighed for at lave sine egne baner, så det er vigtigt at spillet er fleksibelt. Det skal være muligt at ændre på følgende variabler i spillet:

- Broens længde.
- Bro point for hvert stykke der bygges.
- Bilens hastighed.
- Bilens start tid.
- Antal talbobler.
- Talboblernes vægt/hastighed.
- Talintervallet talboblerne skal indeholde.
- Bonustal, hvilket tal tallinjen skal gå op i før der kommer bonus.

4.1.2 Spiludvikling

Spillet udvikles med et objekt orienteret sprog, som gør det lettere at implementere nye funktionaliteter senere hen.



Figur 4.2: Klassediagram af spillet.

Det er vigtigt at have en overordnet struktur, da der skal holdes styr på objekterne. CameraOverlay holder styr på objekter ved at have dem i en liste, så de altid er lette at få adgang til. De enkelte objekter har direkte adgang til hinanden, som NumberBubble og NumberLine da de kan kollidere med hinanden og dele information.

4.2 Hjemmeside

Hjemmesiden skal have et fint og simpelt design som gør det nemt for brugeren at navigere rundt. Det skal også være muligt at vælge en bane via en tabel som indeholder information omkring alle de baner brugeren har adgang til. Tabellen skal kunne sorteres som brugeren ønsker det og ved mange baner skal tabellen have have sidetal. Der skal være gruppefordeling på de forskellige bruger, så det kan sættes op på en hel skole med flere klasser/grupper, så brugerne ikke går ind og forstyrrer hinandens baner. Når en bane spilles skal banen kunne bedømmes i form af stjerner, så andre brugere kan se vurderingen af de forskellige baner. Vurderingen er vigtig, da det er muligt for hver enkelt bruger at lave sine egne baner med forskellige sværhedsgrader. Det skal også kunne lade sig gøre at ændre i baner som allerede findes, ved at oprette en nyere version af banen uden at den oprindelige bliver slettet. Det er nødvendigt at den oprindelige bane ikke bliver ændret, da brugerne har mulighed for at snyde med point fordelingen på de forskellige baner. Til det benyttes WordPress da dens dokumentation er så god og beskrivende samt udvalget af plugins og dets flotter brugerpanel.

Lærerne skal have mulighed for at se hvordan deres elever klarer sig, så der skal være en side med spilhistorikker. Disse spilhistorikker skal vises i en tabel med information omkring banen og brugeren. Det skal igen være muligt i tabellen at sortere efter hvad der ønskes som point, antal fejl og så videre. Udover at sortere i tabellen skal der være forskellige visninger af spilhistorikker baseret på forskellige brugere, baner og grupper. Dette skal gøres for at gøre det let for lærerne når de skal bedømme eleverne. Hvis der er mange elever skal tabellen deles over flere sider. For at få et overblik over en klasse skal det udover at vises som en tabel også kunne vises grafisk i form af diagrammer, så lærerne kan sammenligne eleverne. Diagrammet skal have mulighed for at zoome ind og ud hvis der er mange elever.

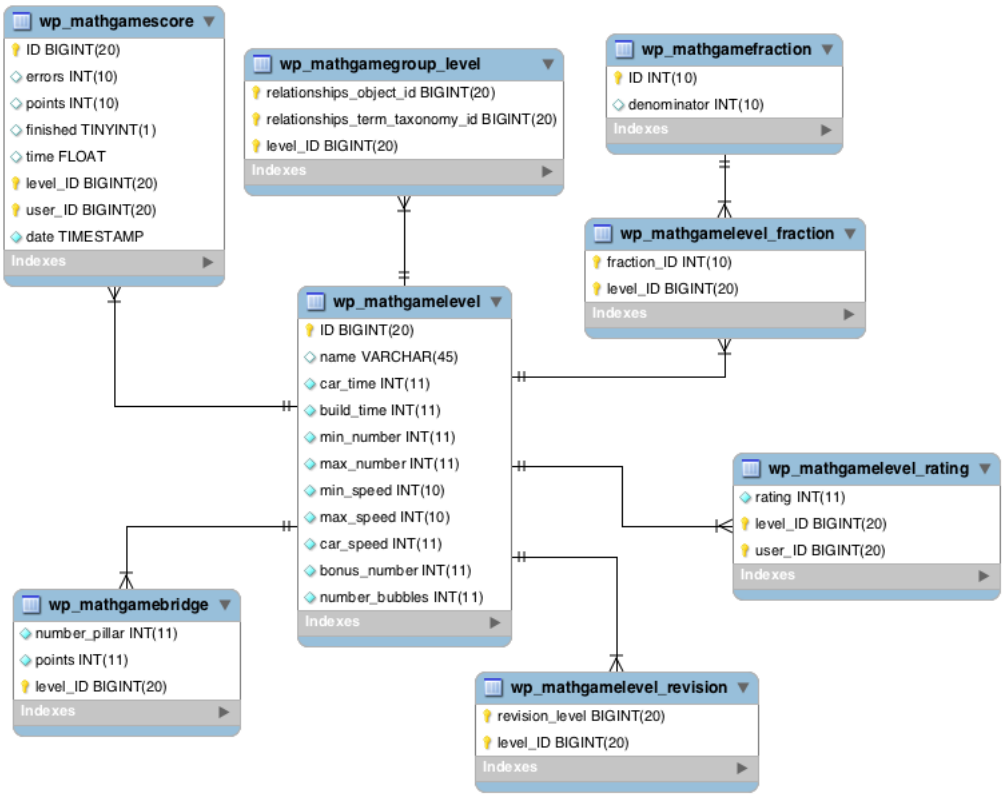
Det er vigtigt at sammenspillet mellem spil og hjemmeside fungerer og opfylder kravene i afsnittet 2 og testes for de use cases der er lavet, se appendiks A.

4.3 Database

Til at gemme data som pointfordeling, antal fejl, bruger-data, med mere, kræver det datalagring i form af en database. Her bruges der en relationsdatabase for at dele informationerne op i forskellige tabeller. WordPress laver i forvejen dens egen database i MySQL, se reference [26], men skal udvides så den passer sammen med resten af funktionerne til spillet (se appendiks C).

WordPress har allerede brugersystemet klart, så skal det udvides til at indeholde spilhistorikker samt bane informationer. Der benyttes værktøjet MySQL Workbench, se reference [27], til at designe database udvidelsen. Dette giver mulighed for at designe databasen som et klassediagram og derefter implementere det til MySQL da MySQL Workbench selv kan generere MySQL queries[1] ud fra database designet.

[1] Query er en kommando med en forespørgsel til databasen. Dette benyttes hver gang der skal kontakt med databasen.



Figur 4.3: Spillets database udvidelse til Wordpress

Denne databasestruktur gør, at der kan oprettes baner til hver enkelt gruppe samt der er gjort plads til spilhistorikker. Banerne har også mulighed for at blive bedømt af hver bruger og ved ændring af en bane så skal der oprettes en revision af den oprindelige bane. Dette sikrer at der ikke snydes ved at klare de lette baner og derefter lave dem svære. Der er udover det gjort klar til brøker i banerne i databasen hvis det skulle blive aktuelt i senere udvikling af spillet.

Der benyttes InnoDB som er standard lagerings motor for MySQL. Det har en del features at tilbyde som blandt andet foreign-keys. Foreign-keys benyttes ved relationer og gør det muligt at forbinde tabeller med hinanden.

5 Implementation

5.1 Spillet

Spillet bliver lavet i Unity3D, da det viser sig at have gode muligheder for at lave alt fra 3D-spil til ganske almindelige 2D-spil. Det er muligt at lave rigtig avanceret spil med Unity3D, så der er rig mulighed for udvidelse af spillet senere hen. Samtidig så understøtter det de fleste moderne internet browsers med et plugin og kan udvikles til enheder som mobiltelefoner og tablets.

Spillet gør, som beskrevet i 4.1.2, brug af objekter til at indeholde figurenes funktioner. Ved hjælp af objekter gør det mere overskueligt, da koden er delt op i hver sine objekter og hvis der ønskes nye funktionaliteter senere hen, så skal der blot tilføjes nye objekter med nye funktionaliteter, som også vist i figuren 4.2.

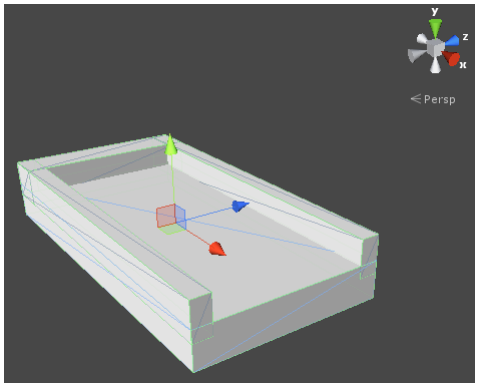
Unity3D har nogle funktioner i forvejen i deres framework til JavaScript, som bliver brugt rundt om i koden.

- **Start**: Første funktion der bliver kaldt når objektet er klar til brug.
- **Update**: Funktion der bliver kaldt ved hvert frame.
- **FixedUpdate**: Funktion der ligesom Update, men benyttes ved brug af fysiske kræfter.
- **OnGUI**: Funktion der benyttes til at tegne eller skrive information på skærmen.
- **OnCollisionEnter**: Funktion der bliver kaldt ved en kollision.
- **OnCollisionExit**: Funktion der bliver kaldt når kollisionen stopper.

Spillet er implementeret i Unity3D med UnityScript (JavaScript) og er delt op i tre dele bilen, broen og talbobler samt tallinjen.

5.1.1 Bilen

Bilen som skal over broen kræver nogle effekter for at det skal ligne en bil der kører over, samtidig med at den skal have nogle funktioner. Unity3D har allerede nogle af disse funktioner og effekter klar, som fysiske krafter med tyndgekraft (se afsnit 5.1.5.1), kollision detektering (se afsnit 5.1.5.3) og Wheel Collider (se afsnit 5.1.5.2). Den fysiske kraft giver noget realisme på bilen og gør det muligt for den, at falde i vandet hvis broen ikke er bygget færdig. Det afhænger altså så om underlaget er solidt og til stede under bilen. Hvis bilen kører i vandet, så er der lavet en tømmerflåde til at samle bilen op ved et funktionskald. Funktionen bliver kaldt med bilens x-position og når tømmerflåden kommer hen til bilens x-position så samles bilen op på tømmerflåden og sejles tilbage til broen. Denne funktion samt designet af tømmerflåden er igen meget simpel.

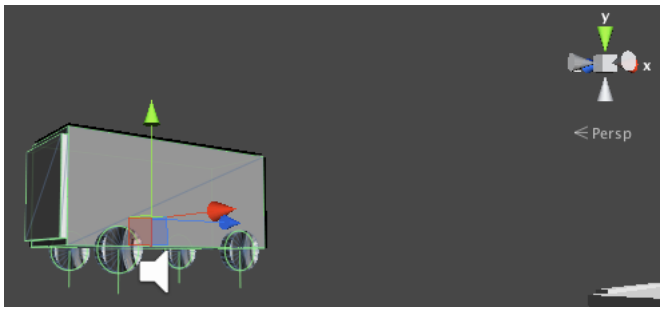


Figur 5.1: Tømmerflåde design til at samle bilen op.

Bilen er lavet ved hjælp af programmet Autodesk Maya[1]. Bilen består af i alt fem dele, hvor et rektangel er bilens krop, med fire tilhørende cylinder som hjul. Dette er en meget simpel form for bil der er lavet i 3D, men da hovedfokus ligger på funktionaliteten, så benyttes simple modeller. Disse modeller kan altid erstattes med nye 3D-modeller uden at funktionaliteten skal ændres. Der er taget udgangspunkt i en vejledning omkring det at lave et bilobjekt og give den funktionaliteter som en bil, se reference [6]. Wheel Collider gør det muligt at få hjulene til skubbe bilen frem eller tilbage afhængig af drejningsmomentet / hastighed. Bilens hjul skal samtidig rotere når bilen kører, så animationer skal laves ved siden af og afhænger af hvor hurtig bilen kører. Dette er igen gjort ved hjælp af vejledningen i referencen [6].

For at visualisere når bilen stopper er der tilføjet et bremselys på bilen, som ligesom bilen er meget simpel. Bremselyset er et objekt i Unity3D, som skifter farve når bilens stop-funktion bliver kaldt. Bilen skal også lyde som en bil og til det er der brugt lyde som alt efter hvor hurtig bilen kører ændrer tone. Der er ved hjælp af reference [8] fundet lyde som benyttes til bilen. Lyde bliver brugt i Unity3D som en AudioSource og har dertil nogle funktioner. De funktioner der er brugt er at afspille lydene i loop, enkeltvis, stoppe dem igen og ændre tone så frekvensen bliver højere og lavere. Reference til lydene er angivet i kilde-koden i CarScript.js.

[1] Autodesk Maya - http://www.autodesk.com/products/autodesk-maya/overview



Figur 5.2: Bil design i Unity3D.

5.1.2 Broen

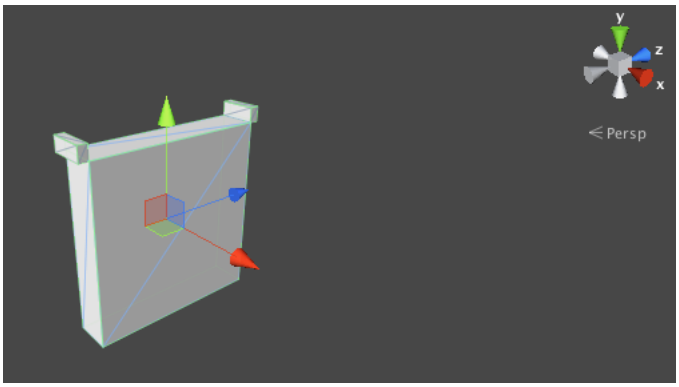
Broen som bilen skal køre over skal laves så den passer til længden af broen. Der skal derfor en algoritme til at bestemme hvor mange brostykker der skal laves og hvor de skal placeres. Ved initialisering af broen oprettes brosøjler og veje så broen er klar til at blive bygget. Brosøjlerne gemmes under vandet og vejene gemmes oppe i skyerne. Ved at bygge broen kommer brosøjlerne op af vandet og når de er kommet op giver de information til vejene om at dale ned. Vejene daler ned på brosøjlerne ved at de får aktiveret en tyngdekraf i y-aksen (se 5.1.5.1) og derved ser naturlige ud når de falder.

```
1  function StartBridge() {
2    // Remove rigidbody from car in Y-axis.
3    car.rigidbody.constraints &= ~RigidbodyConstraints.↩
         FreezePositionY;
4    var x = road.transform.localScale.x;
5    var y = road.transform.position.y;
6
7    // Calculate bridge length and set position.
8    var bridgeLength = (x * (pillarPoints.Length - 1)) / 2;
9    pillar.transform.position.x = -(x * (pillarPoints.Length - 1)) / ↩
         2;
10   var pos = pillar.transform.position;
11   pillarStartY = pos.y;
12
13   // Set road position and initialize array.
14   road.transform.position.x = pos.x + x/2;
15   pillars = new GameObject[pillarPoints.Length];
16   roads = new GameObject[pillarPoints.Length-1];
17   pillars[0] = pillar;
18   roads[0] = road;
19
20   // Set start and car position.
21   land.transform.position.x = pillar.transform.position.x - land.↩
         transform.localScale.x/2 - 0.5;
22   carScript.setXPos(land.transform.position.x);
23
24   // Create the bridge objects (roads and pillars)
25   for (var i = 1; i < pillarPoints.Length; i++) {
26     pillars[i] = Instantiate(pillar, Vector3(-bridgeLength + (x*i),↩
           pos.y, pos.z), pillar.transform.rotation);
27     if (i < pillarPoints.Length-1) {
28       roads[i] = Instantiate(road, Vector3(pillars[i].transform.↩
             position.x + x/2, y, pos.z), road.transform.rotation);
```

```
29      }
30    }
31
32    // Set end by cloning start object and rotation 180 degrees.
33    finishLand = Instantiate(land, Vector3(pillars[i-1].transform.↩
          position.x + land.transform.localScale.x/2 + 0.5, land.↩
          transform.position.y, land.transform.position.z), land.↩
          transform.rotation);
34    finishLand.transform.rotation.y = 180;
35    carScript.length = finishLand.transform.position.x - finishLand.↩
          transform.localScale.x / 3;
36  }
```

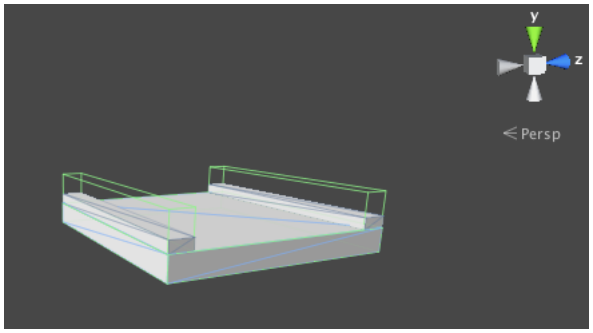
Listing 5.1: Initialisering af broen

For at gøre det lettere at designe broen, så er der i forvejen lavet en bro-søjle og en vej til broen. Når broen bliver initialiseret, så genskaber den bro-søjler og vejene ved at klone de allerede eksisterende bro-stykker. Ved kloning indsættes de klonede objekter i en liste, så der kan holdes styr på broens objekter. Der er liste for både bro-søjler og vejene.



Figur 5.3: Bro-søjle design i Unity3D.

Bro-søjlens design er lavet, så vejene passer lige ned i og ikke kan glide ud til siderne. Dette er også med til at skabe et mere realistisk billede af broen, i stedet for en hel flad bro.



Figur 5.4: Broens vej design i Unity3D.

Broens vej design er lavet ligesom med bro-søjlen, så der er kommet nogle forhøjninger på i siderne. Dette er for at sikre, at bilen ikke kører ud over broen på grund

af bilens køreegenskaber samt udseendet af broen.

5.1.3 Talbobler og tallinjen

Talbobler bruges til at trække ned på tallinjen og skal have en værdi. Denne værdi udregnes tilfældigt i objektet ved at bruge en random generator. Dette gør at den kan finde et tilfældigt tal i et interval.

Talbobler som skal trækkes ned på tallinjen skal have en hastighed de falder ned med. Denne hastighed kan opfattes som en vægt på talboblerne og jo højere vægten er, jo hurtigere falder de. Unity3D har nogle funktioner som opdatere ved hvert frame[2], men når der arbejdes med objekter med fysiske kræfter så er det at anbefale at benytte funktionen FixedUpdate. Hvis FixedUpdate ikke bliver brugt på objekter med fysiske kræfter, så kan objekterne hakke.

```
1  function FixedUpdate () {
2    if (!visible || freeze) {
3      return;
4    }
5
6    if (myTransform.position.y > -1 && !dead)
7    {
8      myTransform.position.y -= speed * Time.deltaTime;
9    }
10   else if (!clicked && myTransform.position.z == 0)
11   {
12     renderer.material.color = Color.red;
13     if (!dead) {
14       cameraOverlay.bubbleDie();
15       timeBubbleDie = Time.timeSinceLevelLoad + cameraOverlay.↩
             numberBubbleSpawnTime;
16     }
17     dead = true;
18     myTransform.tag = "Untagged";
19     if (timeBubbleDie <= Time.timeSinceLevelLoad) {
20       GameObject.Destroy(gameObject);
21       cameraOverlay.setNumberBubbles();
22     }
23   }
24   if (!dead && myTransform.position.z == 0) {
25     if (cameraOverlay.bonusNumber != 0 && random % cameraOverlay.↩
         bonusNumber == 0) {
26       renderer.material.color = Color.Lerp(Color.white, bubbleColor↩
             , Time.timeSinceLevelLoad % 1);
27     }
28     if (myTransform.position.y < 0.8)
29     {
30       renderer.material.color = new Color(1.0F,0.6F,0.0F);
31     }
32     else if (myTransform.position.y < 2)
33     {
34       renderer.material.color = Color.yellow;
35     }
36   }
37
38 }
```

[2]Frame er hver gang skærmen opdateres på computeren

Listing 5.2: Talbobler animation

Når en talbobbel trækkes i, så kræver det at museposition fra computeren oversættes til Unity3D kameraets egen position. Disse positioner udregnes ved klik på en talbobbel.

Dette foregår ved først at registere når musen holdes nede og laver musepositionen om til spillets position for derefter at centrere objektet til musens position.

```
1  function OnMouseDown() {
2    clicked = true;
3    oldPosition = myTransform.position;
4    if (col || dead || freeze) {
5      return;
6    }
7
8    renderer.material.color = bubblePressedColor;
9    screenPoint = Camera.mainCamera.WorldToScreenPoint(myTransform.↩
         position); // Object in screen coordinates
10
11   var mousePos = Camera.main.ScreenToWorldPoint(new Vector3(Input.↩
         mousePosition.x, Input.mousePosition.y, screenPoint.z)); // ↩
         Mouse position in world coordinates
12
13     myTransform.position.x = mousePos.x; // Centering object
14     myTransform.position.y = mousePos.y; // Centering object
15   offset = mousePos - myTransform.position;
16
17
18 }
```

Listing 5.3: Mus klikket i Unity3D

Når musen trækkes samtidig med den bliver holdt inde, så følger objektet med musen. Når talboblen kommer under et punkt på y-aksen, ændrer talboblens z-akse sig, så den passer med tallinjens z-akse. Dette gøres så talboblerne og tallinjen kommer i perspektiv og ikke falder ned på hinanden.

```
1  function OnMouseDrag() {
2    if (col || dead || freeze) {
3      return;
4    }
5
6      if (myTransform.position.y > 2.1) {
7        myTransform.position.z = 0;
8        screenPoint = Camera.mainCamera.WorldToScreenPoint(↩
             myTransform.position);
9      } else {
10       myTransform.position.z = -4;
11       myTransform.localScale.x = 0.5;
12     myTransform.localScale.y = 0.5;
13       screenPoint = Camera.mainCamera.WorldToScreenPoint(↩
             myTransform.position);
14   }
15   var curScreenPoint = new Vector3(Input.mousePosition.x, Input.↩
         mousePosition.y, screenPoint.z);
16
```

```
17      var curPosition = Camera.main.ScreenToWorldPoint(curScreenPoint↩
            ) + offset;
18    myTransform.position = curPosition;
19  }
```

Listing 5.4: Mus træk af talbobbel i Unity3D

Ved at slippe musen så slippes objektet og falder på plads.

```
1   function OnMouseUp() {
2     clicked = false;
3     if (!onNumberLine) {
4       myTransform.position.z = 0;
5     } else {
6       return;
7     }
8     if (col || dead || freeze) {
9       return;
10    }
11    renderer.material.color = bubbleColor;
12
13    if (myTransform.position.y > 10) {
14      myTransform.position.y = 10;
15    } else if (myTransform.position.y > 0) {
16      size();
17    } else {
18      myTransform.localScale.x = 0.5;
19      myTransform.localScale.y = 0.5;
20    }
21    myTransform.position = oldPosition;
22  }
```

Listing 5.5: Mus sluppet i Unity3D

Det er vigtigt at se om en talbobbel er valgt eller om den er faldende, da den kun skal komme ned på tallinjen hvis den er valgt. Derfor sættes en variabel på talboblen om den er klikket på eller sluppet. Når talboblerne bliver trukket ned på tallinjen, så skal talboblen regne ud om den passer på den valgte plads på tallinjen. Udregningen foregår ved, at talboblen tager værdien fra tallinjen og lægger til dens egen værdi hvor den sender besked til CameraOverlay. Efter den har sendt besked fjernes talboblens objekt, da der skal være plads til en ny talbobbel. Talboblen har altså til ansvar om den bliver placeret rigtigt. Hvis museknappen slippes ved kollision på tallinjen, så finder talboblen ud af om det er rigtigt, næsten rigtigt eller forkert. Derefter sender den besked til CameraOverlay for at give besked og eventuelt give point til brugere, hvor den derefter fjerner talboblen for at gøre plads til en ny.

```
1   function Update() {
2     if (onNumberLine && !clicked) {
3       var newNo = random + cameraOverlay.numberOnLine;
4
5       if (newNo == numberLine.number) {
6         onNumberLine = false;
7         cameraOverlay.pointsCorrect(random);
8       } else if (newNo == (numberLine.number - 1) || newNo == (↩
            numberLine.number + 1)) {
9         onNumberLine = false;
10        cameraOverlay.pointsAlmostCorrect(random);
11      } else {
12        onNumberLine = false;
```

```
13        cameraOverlay.timePenalty();
14      }
15      GameObject.Destroy(gameObject);
16    }
17  }
```

Listing 5.6: Indsættelse af talbobbel på tallinjen

Når en talbobbel bliver sat korrekt ind på tallinjen så skal en knap komme frem med at man kan bygge. Denne knap laves i Unity3D i funktionen OnGUI ved at lave et GUI objekt til at visualisere knappen. Denne knap skal have en nedtællingstid inden knappen skal forsvinde igen. Ved at tiden går forsvinder knappen og bygge-pointene nulstilles. Hvis man når at trykke på knappen, så bliver bygge-pointene smidt over i total point og nulstilles herefter. Total point starter på 0 ved hver bane og kan kun tælles op ved at få fra bygge-point. Total point er altså bare en variabel som bliver talt op ved at bygge. Bygge-pointene bliver talt op med 10 point undervejs, men hvis brugeren benytter combo, så bliver pointene hurtigere talt op. Combo'en benytter den algoritme med at den tæller 10 point op hver gang der bygges ganget med hvor høj comboen er. Comboen bliver talt op hver gang der bliver smidt et rigtig tal ned på tallinjen og nulstilles hver gang der bliver bygget, tiden går eller der kommer et tal ned forkert på tallinjen.

Talboblerne er, ligesom ved broen, kun et objekt som bliver klonet ved spilstart. Fordelen ved at klone objektet mange gange er, at de klonede objekter også får alle funktionerne og scripts med fra det oprindelige. Når der bliver klonet objekter, så ryger de ind i en liste af talbobler for, at holde styr på dem og tildele hver deres værdier.

En talbobbel består udover et objekt også af en tekst med en værdi. Denne værdi sættes sammen med talboblen, men teksten skal følge talboblen, når talboblen bevæger sig. For at få teksten til at følge talboblen er der brugt noget færdigkodet fra Unity3Ds egen wiki, se reference [5].

Tallinjen består af nogle objekter i Unity3D, som har mulighed for at kollidere med talboblerne. Ved kollision af en talbobbel og tallinjen er det muligt at finde ud af hvilke tal der bliver indsat.

Først skal der findes ud af om der er kollision på tallinjen og om brugeren holder museknappen inde eller om den er sluppet.

```
1  function OnCollisionEnter(collider : Collision) {
2
3    if (clicked) {
4      if (collider.gameObject.tag == 'NumberLine')
5      {
6        numberLine = collider.gameObject.GetComponent(NumberLine);
7        renderer.material.color = Color.green;
8        onNumberLine = true;
9      }
10   }
11 }
```

Listing 5.7: Kollision på tallinjen

Tallinjen som brugeren skal trække talboblerne ned på kræver, udover at kollidere med talbobler, også at tallinjen kan justeres let. Da det er et krav om at det skal

være simpelt og naturligt at justere på tallinjen er det gjort ved, at trække talboblerne ud til siderne for at øge eller mindske tallinjen. Implementationen kræver at positionen er kendt for den valgte talbobbel, så når talboblen trækkes ud over siderne øges eller mindske tallinjen. Jo længere man trækker talboblerne ud over siden jo hurtigere justere tallinjen sig.

```
1  myTransform.position = curPosition;
2
3  if (myTransform.position.x > 20 && Time.timeSinceLevelLoad > ↩
       timeDragged) {
4    // To the right very fast
5    cameraOverlay.higherNumberLine();
6    timeDragged = Time.timeSinceLevelLoad + 0.05;
7  } else if (myTransform.position.x > 15 && Time.timeSinceLevelLoad >↩
       timeDragged) {
8    // To the right fast
9    cameraOverlay.higherNumberLine();
10   timeDragged = Time.timeSinceLevelLoad + 0.1;
11 } else if (myTransform.position.x > 12 && Time.timeSinceLevelLoad >↩
       timeDragged) {
12   // To the right slow
13   cameraOverlay.higherNumberLine();
14   timeDragged = Time.timeSinceLevelLoad + 0.2;
15 } else if (myTransform.position.x < -20 && Time.timeSinceLevelLoad ↩
       > timeDragged) {
16   // To the left very fast
17   cameraOverlay.smallerNumberLine();
18   timeDragged = Time.timeSinceLevelLoad + 0.05;
19 } else if (myTransform.position.x < -15 && Time.timeSinceLevelLoad ↩
       > timeDragged) {
20   // To the left fast
21   cameraOverlay.smallerNumberLine();
22   timeDragged = Time.timeSinceLevelLoad + 0.1;
23 } else if (myTransform.position.x < -12 && Time.timeSinceLevelLoad ↩
       > timeDragged) {
24   // To the left slow
25   cameraOverlay.smallerNumberLine();
26   timeDragged = Time.timeSinceLevelLoad + 0.2;
27 }
```

Listing 5.8: Justering af tallinjen ved at trække talboblerne ud til siderne

Når tallinjen øges så er det vigtigt stadig at holde styr på hvad indekset er på tallinjen og dette gøres ved at tælle ned hver gang den øges.

```
1  function higherNumberLine() {
2    for (var i = 0; i <= noNumbers; i++) {
3      var tmp : NumberLine = numbersLine[i].GetComponent(NumberLine);
4      tmp.increaseNumber();
5    }
6    numberOnLineIndex--;
7    onNumberLine();
8  }
```

Listing 5.9: Øger tallinjen

Ved at mindske tallinjen så er det samme process bare hvor indekset bliver talt op i stedet.

For at zoom ind eller ud på tallinjen, benyttes scroll på musen, som er et input og

kan registeres i Unity3D. Ved at zoome ind så fjernes der elementer på tallinjen, så tallene på tallinjen får et mindre interval. Ved at zoome ud tilføjes der elementer på tallinjen, så tallinjen indeholder flere tal. Denne process starter med at fjerne alle tallinjens elementer og oprette nye, da tallinjen sørger for at bruge hele skærmens bredde. Efter objekterne er oprettet igen, så kræver det at de får tildelt deres nye talværdi.

```
1 function deleteNumberLine() {
2   var tmp = numbersLine[noNumbers/2].GetComponent(NumberLine);
3   middleNumberOnLine = tmp.number;
4   for (var i = 0; i <= noNumbers; i++) {
5     GameObject.Destroy(numbersLine[i]);
6   }
7 }
```

Listing 5.10: Fjerner tallinjen

Efter tallinjen er fjernet, så genopretter den tallinjen ud fra nogle nye værdier

```
1  function setNumberLine() {
2    space = 20.0 / noNumbers;
3    var c = 0;
4    for (var j = 0.0; j <= noNumbers; j++) {
5      numbersLine[c] = Instantiate(numberLine.gameObject, new Vector3↩
           ((j*space) - 10.0, 0, -4), numberLine.transform.rotation);
6      numbersLine[c].renderer.material.color = Color.gray;
7      c++;
8    }
9
10   for (var i = 0; i <= noNumbers; i++) {
11     var tmp : NumberLine = numbersLine[i].GetComponent(NumberLine);
12     tmp.setNumber(i + middleNumberOnLine - (noNumbers/2));
13     if (numberOnLine == i + middleNumberOnLine - (noNumbers/2)) {
14       numbersLine[i].renderer.material.color = Color.cyan;
15       numberOnLineIndex = i;
16     }
17   }
18 }
```

Listing 5.11: Genopretter tallinjen ud fra nye værdier

Tallinjens tal består af objekter for at kunne kollidere med talboblerne. Disse objekter på tallinjen har en objekt-klasse, som har funktioner til at ændre, øge eller mindske tallene på objektet. Når tallinjen bliver initialiseret, så sættes hvert objekt ind i en liste, for at holde styr på dem. Derved er det let at løbe igennem alle tallinjens objekter hvis der skal ændres i værdierne.

5.1.4 Optimering i Unity3D

Der er gjort brug af reference [7] for at optimere spillet og gøre det så lidt krævende som muligt. Der er sat typer på de forskellige variabler som gør JavaScript meget hurtigere. Der er også gemt objekter i midlertidige variabler i stedet for at kalde objektet flere gange samt hvis et objekt ikke er synligt så stopper dens funktionalitet. Dette er kun på talbobler, tallinjen og andre ting som ikke behøver at gøre noget hvis det ikke er synlig. Denne optimering er fulgt gennem hele udviklingen.

5.1.5 Unity3D funktioner

5.1.5.1 Rigidbody

Rigidbody er fysiske krafter som allerede er implementeret i Unity3D fra start. Rigidbody kan forstås som en tyngdekraft på objekter og gør det let at lave realisme på objekter. Det har den fordel at der er en masse variabler som kan indstilles ved brug af rigidbody, som at låse krafterne i xyz-akserne. Ved brug af Rigidbody gør det det muligt for objekter at støde ind i hinanden og kollidere[3].

5.1.5.2 Wheel Collider

Wheel Collider er lavet i Unity3D for at realisere en kørende bil. Wheel Collider pålægges bilens hjul og kan realisere en motor på bilen. Derudover har det indbygget collision deteketering, som hjælper med at holde bilens hjul på vejen[4].

5.1.5.3 Kollision detektering

Kollision detektering er et redskab i Unity3D, som gør at ting der støder ind i hinanden kan aflæses. Ved en kollision er det muligt at læse hvilke objekter der kolliderer og hvad de indeholder. Til det bruges funktionerne OnCollisionEnter og OnCollisionExit.

5.2 Hjemmeside

Hjemmesiden er lavet i WordPress og sørger selv for at sætte en startside op ved installation. Der er så implementeret et nyt tema, som er designet til siden og bestemmer indholdet på siden. For at gøre dette let og overskueligt er der brugt Bootstrap, se reference [18]. Bootstrap er et framework som gør det let at lave design til hjemmesider og gør hjemmesiderne responsive så de fungerer på forskellige opløsninger, da det indeholder en masse funktioner og design elementer.

Der benyttes WordPress' egne funktioner til sikkerhed samt imod database injections. Database injections er hvor en bruger sender database kommandoer via hjemmesiden og kan derved sende nogle kommandoer som rydder databasen og meget mere. Det er derfor vigtigt at sikre sig mod database injections. For at beskytte siden mod database injections, bruges prepare-statements inden data bliver sendt eller modtaget fra databasen.

```
1 $groups = $wpdb->get_results($wpdb->prepare(
2   "
```

[3] http://docs.unity3d.com/Documentation/Components/class-Rigidbody.html
[4] http://docs.unity3d.com/Documentation/Components/class-WheelCollider.html

```
3     SELECT t.term_id, t.name
4     FROM $wpdb->group_level gl
5     INNER JOIN $wpdb->terms t ON gl.relationships_term_taxonomy_id = t.term_id
6     WHERE level_ID = %d
7     ", $level
8 ));
```

Listing 5.12: Eksempel på prepared-statements med Wordpress

Udover at have prepare-statements med, så bruges WordPress' funktion til at hive data ud af databasen.

- $wpdb->get_results
- $wpdb->get_col
- $wpdb->get_row
- $wpdb->get_val

Disse funktioner bruges alt efter hvad man skal have ud af databasen. Hvis man skal have flere rækker og kolonner ud af databasen, så bruges $wpdb->get_results da dataen kommer ud som et array med indeholdende objekter. En række præsenterer hvert element i array'et og kollonerne er objekter for hver række. De andre funktioner bruges hvis der skal hentes enkelte kolonner, enkelte rækker eller bare en variabel ud af databasen.

For at få spil og hjemmeside til at fungere sammen og sende oplysninger mellem hinanden, kan det gøres med JavaScript funktioner. JavaScript bruges til det da spillet indlæses ved hjælp af klienten. Ved hjælp af JavaScript er det derfor muligt at tilgå funktioner i Unity3D ved at initialisere spillet og sende beskeder via JavaScript til Unity3D. Dette benyttes til at initialisere baner i spillet som tid, bro-længde og så videre i spillet. Her er der brugt jQuery, se reference [17], som Unity3D allerede gør brug af ved initialisering af spillet.

```
1 u.initPlugin(jQuery(".unityplayer")[0], "<?php print THEMEROOT; echo $game; ?>");
```

Listing 5.13: Initialisering af Unity3D plugin.

```
1 u.getUnity().SendMessage("MainCamera", "getCarTime", <?php echo $curlevel->car_time; ?>);
```

Listing 5.14: Sende beskeder til Unity3D funktion med parameter.

For at få Unity3D til at sende beskeder til serveren kræver det nogle funktioner i JavaScript som Unity3D kan tilgå. Dette benyttes i starten og slutningen i hvert spil. Det er nemlig vigtigt at JavaScript først sender beskeder til Unity3D når spillet er indlæst og klar. Unity3D kalder funktionen UnityIsReady når spillet er indlæst og klart. Funktionen UnityIsReady sørger dernæst for at sende information om banen til Unity3D, så spillet har den data det skal bruge for at oprette banen. Når spillet er færdig bliver funktionen UnityFinished kaldt, som tager argumenter med information om hvordan brugeren klarede det nuværende spil. Denne information skal lageres

i databasen og da klienten ikke kan tilgå databasen, så kræver det nogle ekstra funktioner. Til det bruges AJAX[5], som gør det at den kalder en ekstern side til at få disse informationer sendt uden at brugeren lægger mærke til det. Der er valgt at bruge AJAX, da det er begrænset hvor meget data der skal sendes frem og tilbage mellem klienten og serveren. De eneste gange der bliver brugt AJAX i spillet er når et spil sluttes og ved vurdering af banen. For at lave AJAX kald bruges et JavaScript framework kaldet jQuery, se reference [17]. jQuery er et populært JavaScript framework og er lavet for at simplificere JavaScript. jQuery gør det hurtigere og lettere at lave JavaScript til hjemmesider, da det indeholder en masse funktionaliteter og genveje til JavaScripting.

```
1  jQuery.ajax({
2    type: 'POST',
3    cache: false,
4    url: "<?php echo home_url() . '/wp-admin/admin-ajax.php'; ?>",
5    data: {
6      action: 'addScoreToLevel',
7      level: <?php echo $level; ?>,
8      security: '<?php echo $ajax_nonce; ?>',
9      point: points,
10     error: errors,
11     time: playtime,
12     finish: finished
13   },
14   success: function(data, textStatus, XMLHttpRequest) {},
15   error: function(MLHttpRequest, textStatus, errorThrown) {
16     alert("<?php _e('Could not upload score.', 'wpbootstrap'); ?>")↩
           ;
17   }
18 });
```

Listing 5.15: AJAX funktion til at sende data til databasen efter endt spil

AJAX kalder til en php-fil i WordPress, som sørger for at videresende data til AJAX funktionen. Denne AJAX funktion laves i functions.php, som er en fil indeholdende med diverse PHP-funktioner. Hertil er funktionen AddScoreToLevel oprettet for, at sende dataen op til databasen. Udover at sende alt data med omkring spillet, så sendes en kode med, som skal sikre mod falske AJAX kald. For at lave denne kode bruges WordPress funktionen wp_create_none(key) som genererer en kode ud fra tid, key og bruger id.

Efter AJAX-kaldet så kaldes en af funktionerne success eller error, som meddeler om AJAX kaldet er udført korrekt eller om det giver en fejl.

For at få oversættelse på siden benyttes Wordpress' egne funktioner til at gøre dette. Alt dette foregår i PHP og gør det muligt for enhver at oversætte hjemmesiden til deres eget sprog. De enkelte funktionskald der er brugt:

- ___(text, domain): Udskriver ikke teksten og bruges ved at sætte tekst til variabler.
- _e(text, domain): Udskriver teksten som almindelig PHP echo og bruges når der skal udskrives en tekst som ved brug af PHP echo.

[5] Asynchronous JavaScript and XML, se reference [20]

Ved hjælp af programmet Poedit, se reference [21], kan den finde alle disse funktionskald og gøre det muligt at indsætte en ny tekst i stedet. Derefter indstilles WordPress til ens sprog ved at ændre WPLANG i wp-config.php. Indtil videre er det kun engelsk og dansk som er tilgængelig. Hjemmesiden blev lavet på engelsk, men da produktet skulle testes på danske skoleelever er der derved lavet en dansk oversættelse.

WordPress har allerede dens egen brugersystem, men mangler at kunne placere disse brugere ind i deres grupper/klasser. WordPress har den fordel, da der er en masse brugere af det, at det er rigt på indhold af plugins. Disse plugins er tilgængelige for alle og kan være alt fra gratis til at koste penge. Ved hjælp af plugins, så bruges et plugin ved navn User Groups, se reference [13]. Dette plugin er meget simpelt og gør det muligt at oprette grupper og tildele disse grupper til hver enkelt bruger. Udover User Groups er der krav på endnu et plugin, da hver enkelt bruger skal have deres egne rettigheder. Til det bruges et plugin User Role Editor, se reference [14], som gør det meget enkelt at ændre på bruger rettigheder for hver enkelt bruger. Disse rettigheder er med til at bestemme hvilke sider de enkelte brugere har adgang til og hvilke funktionaliteter der er til rådighed. For at gøre det let at bestemme hvilke sider der skal være adgang til for hver bruger er der lavet tre slags menuer til lærere, elever og gæster. Ved at ændre i disse menuer kan der gives adgang til de forskellige sider. Det er altså op til adminstratoren af siden at bestemme hvilke roller der skal have adgang til hvilke sider.

Implementationen med at observere elevernes udvikling i spillet er lavet ved at kunne sortere mellem bane, gruppe og brugere. Her er det muligt at se over en tabel som sorterer efter højeste score, laveste fejl og bedste tid. Hvis der ønskes at se en udvikling over tiden, så kan tabellen ændres til et diagram og ved hjælp af HighCharts, se reference [15], kan der zoomes ind og ud på grafen. HighCharts gør det let at benytte da diagrammet har rige muligheder for blandt andet at have flere akser i samme diagram. Dataen er i form at JSON[6], som PHP har funktioner til at lave data om til ved at bruge json_encode(array).

5.3 Database

WordPress laver allerede sine egen tabeller databasen med plads til bruger-data i MySQL. Det skal der ikke laves noget om på, da WordPress allerede har sikkerheden i orden med at kryptere, hashe og salte kodeord, samt information omkring brugerne og brugersystemet fungerer som det skal. MySQL er valgt, da WordPress allerede gør brug af det og det benytter relationer frem for objekter. Til at gemme banerne og spilhistorikkerne, så kræver det at denne database udvides, se afsnittet 4.3.

For at gøre det let i installationen af produktet, så laves database udvidelsen som et plugin i WordPress, se reference [16]. Ved hjælp af dette plugin er det betydeligt lettere at sætte projektet op, da der ved oprettelse af database tabellerne skal sørges for at oprette tabellerne i rigtig rækkefølge, da de er i relation til hinanden. Plugin'et sørger selv for at installere denne database udvidelse ved et enkelt tryk.

[6] JavaScript Object Notation - Simpel datastruktur som benyttes i JavaScript

For at oprette database tabeller benyttes MySQL query CREATE TABLE og der benyttes BIGINT til ID, som går mellem

$$-9,223,372,036,854,775,807 \ - \ 9,223,372,036,854,775,807$$

ID bruges på de fleste tabeller for, at identificere tabellerne, da ID'erne er unikke i hver tabel. ID er altid positivt, så der bruges UNSIGNED BIGINT som går fra

$$0 \ - \ 18,446,744,073,709,551,615$$

Dette er gjort for at gøre plads til de mange baner der kan laves, antallet af brugere, spilhistorikker og så videre. Ved at se på UNSIGNED INT kan forskellen tydelig ses da UNSIGNED INT *kun* går fra

$$0 \ - \ 4,294,967,295$$

```
1   CREATE TABLE IF NOT EXISTS `mathgamelevel` (
2   `ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
3   `name` varchar(45) DEFAULT NULL,
4   `car_time` int(11) NOT NULL,
5   `build_time` int(11) NOT NULL,
6   `min_number` int(11) NOT NULL,
7   `max_number` int(11) NOT NULL,
8   `min_speed` int(10) unsigned NOT NULL,
9   `max_speed` int(10) unsigned NOT NULL,
10  `car_speed` int(11) NOT NULL,
11  `bonus_number` int(11) NOT NULL,
12  `number_bubbles` int(11) NOT NULL,
13  PRIMARY KEY (`ID`)
14) ENGINE=InnoDB
```

Listing 5.16: Opret level-tabel i databasen for at gemme banerne

For at udføre relationer til de andre tabeller skal der tilføjes en foreign-key, som understøttes i InnoDB. Foreign-key bruges til at identificere en anden tabel og er altså til at forbinde tabeller. Det er vigtigt at plugin som User Groups er aktiveret da plugin sørger for at lave database tabeller så det er muligt at have grupper med. Hvis det ikke er aktiveret når databaseudvidelsen skal laves, så kommer der en fejlmeddelse op med at plugin'et ikke er aktiveret. Grunden til denne fejlmeddelelse er at databaseudvidelsen gør brug af User Groups.

```
1   CREATE TABLE IF NOT EXISTS `mathgamescore` (
2   `ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
3   `errors` int(10) unsigned DEFAULT NULL,
4   `points` int(10) unsigned DEFAULT NULL,
5   `finished` tinyint(1) DEFAULT NULL,
6   `time` float unsigned DEFAULT NULL,
7   `level_ID` bigint(20) unsigned NOT NULL,
8   `user_ID` bigint(20) unsigned NOT NULL,
9   `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
10  PRIMARY KEY (`ID`,`level_ID`,`user_ID`),
```

```
11   KEY `fk_mathgamescore_mathgamelevel1_idx` (`level_ID`)
12 ) ENGINE=InnoDB
```

Listing 5.17: Opret spilhistorik-tabel i databasen ved at lave en mange-til-én relation

Når databasen er lavet skal der indsættes data i tabellerne og dette gøres ved hjælp af MySQL query INSERT. Denne kommado bliver brugt når der oprettes baner, når spillet er færdigt og ved vurdering af banen.

```
1 INSERT INTO $wpdb->score
2 ( errors, points, finished, time, user_ID, level_ID )
3 VALUES ( %d, %d, %d, %f, %d, %d )
```

Listing 5.18: Indsæt spilhistorik i databasen efter endt spil

Det skal selvfølgelig også være muligt at hente data ud af databasen for de enkelte grupper og brugere. Dette gøres med MySQL query SELECT, som bliver brugt hvergang der skal hentes data ud fra databasen som valg af bane, visning af baner, spilhistorikker, med mere. Når der trækkes data ud fra databasen er det muligt at finde gennemsnittet af den data der hives ud af databasen ved at benytte funktionen AVG i query'en. Udover gennemsnitsberegning er det også muligt at hive det største tal ud, få antallet af rækker, med mere.

```
1 SELECT AVG(rating)
2 FROM $wpdb->level_rating
3 WHERE level_ID = %d
```

Listing 5.19: Få gennemsnittet af banens bedømmelse

Ved at slette en bane er det vigtigt at relationerne bliver slettet i den rigtige rækkefølge, da de afhænger af hinanden. Rækkefølgen er at der først ledes efter revisioner af den bane som skal slettes, derefter slettes spilhistorikkerne og til sidst selve banen. Slet bane foregår med MySQL query DELETE.

```
1 DELETE FROM $wpdb->level
2 WHERE ID = %d
```

Listing 5.20: Slet bane fra databasen

6 Resultater

6.1 Test med unge elever

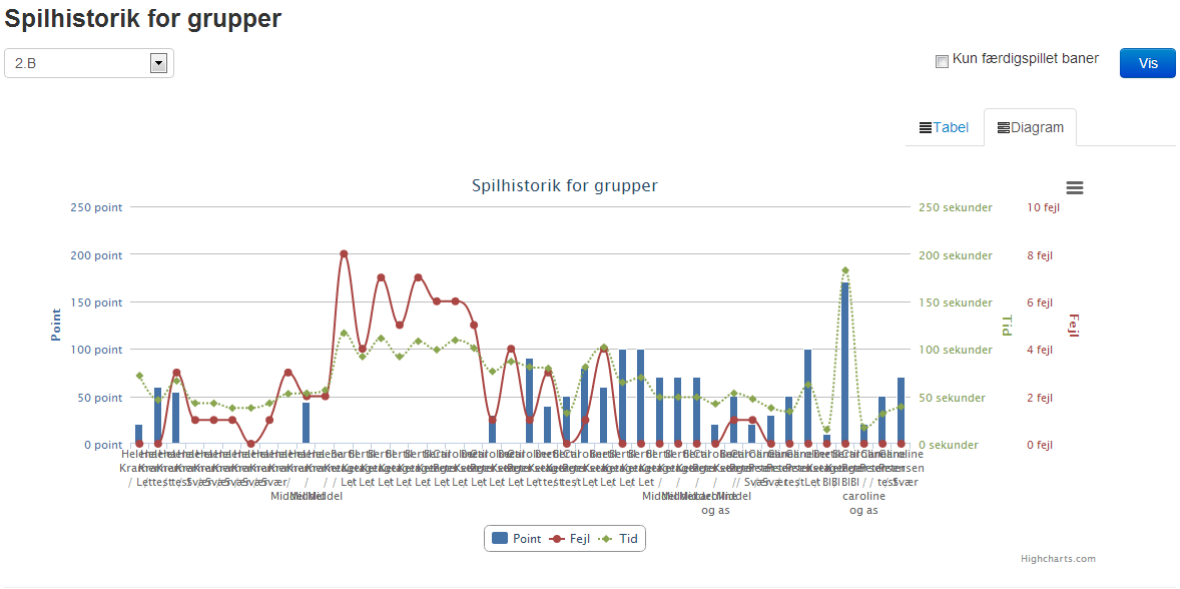
Spillet er testet i nogle små klasser for at se hvordan unge elever klarer sig i spillet. Samtidig har de unge elever svaret på nogle spørgsmål angående spillet og hvordan de normalt klarer sig i matematikken. Spørgeskemaerne indeholder spørgsmål om de enkelte elever forstod spillet, hvor sjovt de syntes det var og hvad der kunne gøre spillet bedre. Samtidig er der spurgt om hvordan børnene har det med matematikken for, at se hvordan det hænger sammen med hvad de synes om spillet. Imens eleverne har testet spillet er der observeret hvordan de klarer sig undervejs i spillet og talt med eleverne.

Der var bekymringer med at plugin til Unity3D ikke kunne blive installeret da det måske krævede special bruger adgang til computeren. Det gik fint med at installere Unity3D plugin på computerne og kan tyde på, at der ikke kræves special bruger adgang for at installere plugin til internet browseren. Installationen foregik nemt og eleverne klarede det selv uden problemer.

Spillet er testet i to klasser, en 2. klasse og 4. klasse. Eleverne fik hver i sær et login til hjemmesiden og skulle derfra selv finde ud af hvordan de skulle spille spillet. Alle eleverne brugte internet browseren Google Chrome. I begge klasser var der i starten frustation over ikke at kunne finde ud af spillet og hvordan skulle man gøre. Eleverne prøvede at trække tal ned på tallinjen og lagde ikke mærke til at tallinjen ændrede sig og at byg-knappen blev synlig. I løbet af de første 10 minutter begyndte de enkelte elever at finde ud af hvordan spillet fungerede ved at prøve sig frem. Eleverne havde dog problemer med at trække tal ned som ikke kunne ses på tallinjen, da de ikke vidste hvordan tallinjen skulle justeres. Efter at eleverne havde spillet et stykke tid fik de at vide at de kunne lave baner og var straks rigtig vilde efter at oprette deres egne baner. Nogle af eleverne havde svært ved at forstå de enkelte parameter der kunne indtilles ved bane oprettelse. Da de fik lavet banen gik nogle af dem kun ind og prøvede banen i stedet for at gemme den hvilket ikke var meningen med ban oprettelse. Udover det så var success stor ved oprettelse af baner.

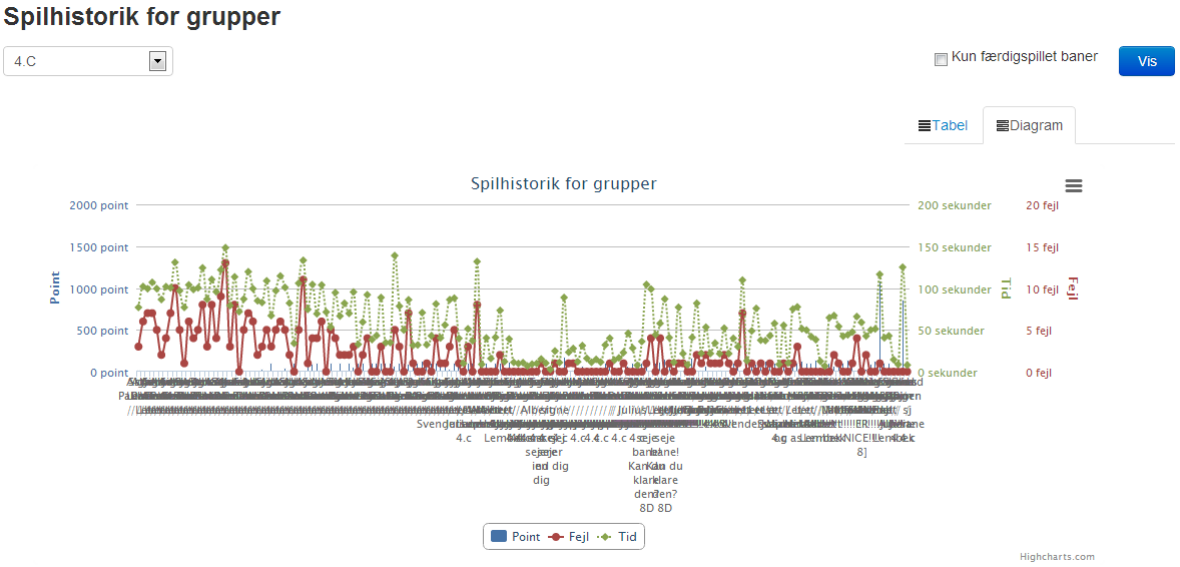
De første baner blev klaret af eleverne, men uden brug af combo point på tallinjen. Der var altså ingen elever som vidste at dette kunne lade sig gøre. Der opstod også nogle fejl i spillet som ikke kendtes før, som at tallene nogle gange ikke blev registeret på tallinjen. Dette var meget sjældent, men er en fejl der skal rettes da det kan få eleverne til at tro de laver en fejl.

Ved hjælp af grafer for de to klasser er det muligt at se en udvikling over tiden



Figur 6.1: Spilhistorik for 2. klasse.

Ved 2. klasse kan det ses at de forskellige spil indeholder en del fejl og at pointene ikke er så høje. Dette kan nemt tyde på at eleverne har haft svært ved at forstå spillet i starten. Det hjælper som tiden går at pointene bliver højere og fejlene er reduceret.



Figur 6.2: Spilhistorik for 4. klasse.

Ved 4. klasse kan det ses, lige som ved 2. klasse, at eleverne bliver bedre til spillet som tiden går. Der var flere der prøvede spillet i 4. klasse og eleverne spillede spillet i længere tid samt de prøvede flere baner ad gangen. Udviklingen er dog mere markant i 4. klasse og tyder på at de har haft lettere ved spillet.

Der var lavet 3 baner til at starte med let, middel og svær. Nogle af eleverne valgte dog at starte med den svære og fik en svær start. Dette har måske været fordi de selv synes de er gode til matematik og vil have noget udfordring eller at de bare har trykket uden at læse banens navn.

Eleverne fik oprettet deres egne baner, hvilket eleverne var helt vilde med at prøve. De syntes det var spændende selv at være kreative og kunne sammensætte deres egne baner, selv bestemme sværhedsgraden og udfordre sine venner.

Eleverne i 2. klasse syntes samlet set at spillet var mellem middel og svært og de fleste kunne forstå spillet. Deres vurdering af spillet samt deres matematikevner lå mellem godt og middel.

Eleverne i 4. klasse syntes samlet set at spillet var middel og næsten alle kunne forstå spillet. Spillet er lige som i 2. klasse at de synes spillet er mellem god og middel ligesom deres evner i matematikken.

Nogle af eleverne fik ikke svaret på side 2 i spørgeskemaet og tælles ikke med i den samlede vurdering, samt nogle elever skrev tekst i stedet for at sætte krydser.

	2. klasse	**4. klasse**
Sværhedsgrad	Middel / svært	Middel
Forstod spillet	Ja	Ja
Vurdering af spillet	God / middel	God / middel
Evner i matematik	God / middel	God / middel
Lyst til at spille	Ja	Ja

Tabel 6.1: 2. klasse og 4. klasse samlet besvarelser

Det kan tyde ud fra observationer og spørgeskemaerne, at spillet er for svært i starten. Der mangler stadig nogle ting som animationer og hjælp til hvordan spillet skal spilles. Dette skal gøres uden tekst, da eleverne ikke læser hvad der står alligevel.

Eleverne blev også spurgt om hvad der kunne forbedre spillet i senere udvikling, hvor til de fleste ønskede flere og sejere biler samt flere broer og flere udfordringer. Nogle elever var så kreative at de ønskede nogle mere specifikke udvidelser til spillet.

- Bedre grafik og flere farver.
- Historie til spillet.
- Lave og designe sine egne broer.
- Lave og designe sin egen bil.

Alt i alt har spillet nok været for svært i 2. klasse og egnet sig bedre til 4. klasse. Spillet kan have en svær tilgang, da eleverne har svært ved at se hvordan man spiller spillet. Det kræver altså en mere tydelig tilgang hvordan man gør, som kan gøres med alt fra animationer til små hints.

6.2 Resultater i løbet af udviklingen

Der er foretaget løbende test i udviklingen af projektet for, at se om produktet lever op til forventninger. Der er kommet få problemer op gennem udviklingen, som:

1. Problem med at trække talbobler. Det fejler når de trækkes ned på tallinjen, da z-aksen ændrer sig, det vil sige, at talboblerne bliver forskudt i forhold til musen.
2. Bilens hjul roterer rundt om den forkerte akse.
3. Bilen kører skævt og ud over broens sider
4. Tallinjen sætter forkert indeks når den bliver justeret.
5. Spillet vises i et meget lille vindue i Internet Explorer.
6. Bilen kan falde ned på en bro-søjle og blive der uden at falde i vandet. Dette betyder at spillet ikke kan tabes.

Nogle af disse fejl er blevet rettet efterfølgende som de er blevet testet.

1. Udregning af musens placering med talboblen blev rettet, se afsnit 5.1.3.
2. Rotationsaksen blev rettet.
3. Bilens køreegenskaber blev rettet, samt broens design blev ændret, se afsnit 5.1.2.
4. Ved justering af tallinjen blev indekset talt op på tallinjen.
5. Endnu ikke løst, men virker i de andre internet browsers. Tyder på at JavaScriptet ikke kan finde højde og bredde af internet browseren.
6. Knap til at genstarte bane er lavet, men bør laves så bilen dør og banen genstarter.

7 Diskussion

Der er udviklet et spil til unge elever med henblik på matematikken. Spillet har i følge resultaterne i afsnit 6 klaret sig udmærket i forhold til det er første prototype. Spillet har givet eleverne nogle udfordringer, men også fået dem til at indse at matematikspil kan være sjovt og lærerigt. Eleverne i 2. klasse havde problemer med spillet i starten og er muligvis for svært for dem at sætte sig ind i og som er noget der skal gøres ved. Efter eleverne begyndte at finde ud af spillet fungerede det fint. Eleverne i 4. klasse havde problemer i starten af spillet, men fandt ud af spillet efter få minutter og så blev spillet pludselig rigtig sjovt for dem.

Der skal altså arbejdes med introen til spillet og hvordan man spiller det, da hvis spillet ikke er til at finde ud af i starten, så lukker eleverne spillet ned og gider ikke bruge tid på det. Der var ingen af klasserne der fandt ud af at bruge combo-systemet og byggede point efter hver talbobbel. Det var meningen med spillet at tallinjen skulle bygges højere og højere op indtil man ikke kunne længere. Derved kunne eleverne udfordre sig selv og finde baner der var mere passende i sværhedsgraden.

Nogle af eleverne svarede at de ikke ville spille spil i matematiktimerne og blev udspurgt hvorfor de ikke havde lyst til det. Eleverne havde misforstået spørgsmålet og troede at spørgsmålet lød om de kun ville spille computerspil i matematiktimerne og ikke have matematiktimer. De forskellige svar til spørgeskemaet kan altså have forskellige meninger efter hvilken elev der har udfyldt dem.

Eleverne viser sig at være glade for at være de kreative og designe deres egne elementer i spillet. Det har været en god ide at kunne designe sine egne baner og bør udvides så eleverne har større frihed i spillet. Der blev lavet så svære baner at det næsten var umuligt at klare dem. Dette er ikke meningen med oprettelser af baner og bør nok være at eleverne selv skal kunne klare banen inden den kan gemmes og bliver lagt ud. Nogle elever efterspørger end da om det ikke er muligt at designe sin egen bil samt de forskellige broer. Denne ide er tænkt lidt videre på og kan implementeres ved at eleverne får point ved at klare en bane og kan bruge disse point på nye materialer. Disse materialer kan være alt fra nye ting til broen til at designe sin bil og gøre den til sin egen.

Jo mere der gøres for at eleverne synes spillet er sjovt og fedt, jo mere lærer de. Hvis de bliver så motiveret af at spille et matematik spil at der går konkurrence i hvem der har de sejeste ting, så ville det blive en stor success.

7.1 Fremtidig udvidelser

Udover at rette de fejl der opstod ved testen, som beskrevet i afsnit 6, så vil spillet kunne udvikles til ældre elever. For at gøre dette, kræver det nogle nye og flere udfordringer, da elever i højere klasser skal have andre udfordringer end at plusse og minusse. Nye udfordringer kan være alt fra brøker til ligninger. For at få det med i spillet, så kræver det blot at implementere nogle nye objekter, ligesom der er gjort med de almindelige talbobler. Database udvidelse kan dog blive nødvendig hvis det skal passe sammen med oprettelse af baner på hjemmesiden, samt det i nogle tilfælde kan blive nødvendigt at ændre på tallinjens design.

Animationer i spillet og guide spilleren til at benytte combo, få bygget broen og få talboblerne ned på tallinjen bør implementeres i en endelig version. Som det ses i testen, se afsnit 6.1, med skoleeleverne, så kan dette være vigtigt for at de forstår spillet og synes spillet er interessant. Dette kræver en del arbejde og tid i projektet da det skal testes løbende med skoleeleverne som funktionerne bliver implementeret. Grafikken i spillet står også til klar forbedring eftersom der kun bliver brugt simpelt grafik i form af simple objekter, hvilket eleverne også nævner i afsnit 6.1. Dette er noget som kan forbedre spillet markant og vil øge elevernes interesse og motivation til spillet.

Hjemmesiden kan også, udover at rettes for fejl, indeholder flere funktioner. Nye funktioner som at eleverne kan udfordre hinanden ved at sende baner til hinanden og lave konkurrencer i klasserne.

Hjemmesiden skal også have mere indhold og mere design for at gøre den mere indbydende. Hjemmesiden skal indeholde tabeller over de bedst bedømte baner, mest spillede baner, med mere. Dette kan sagtens laves ud fra de data der er tilgængelig i databasen lige nu og kræver bare at der bliver gjort plads til det på siden.

8 Konklusion

Der er alt i alt lavet et større projekt med indeholdende brugersystem på en hjemmeside, samt et spil der formår at udfordre eleverne i matematikken. Spillet har, udover at give eleverne nogle udfordringer, også givet muligheden for eleverne selv at udfordre sin klasse ved at oprette sine egne baner. Eleverne har altså mulighed for at prøve deres egne ting af og selv være de kreative. Spillet er testet i to klasser og har vist at spillet er spilbart, men kræver nogle forbedringer og justeringer for at alle eleverne kan være med. Disse forbedringer og justeringer omhandler tilgangen til spillet, da spillet er svært at finde ud af i starten. Spillet klarer sig ikke godt de første minutter hos eleverne, da de ikke aner hvordan det fungerer. Efter de første minutter er gået med spillet, så går det op for eleverne selv hvad det går ud på og hvordan spillet skal spilles. Udover de forbedringer og justeringer er der ikke lagt noget avanceret grafik på selve spillet og kan forbedre spillet markant. Udover de forbedringer der kan laves til spillet, så er det også gjort muligt at udvikle senere på spillet ved at oprette nye objekter. Det gør det muligt at udvikle spillet til større klasser og komme med flere udfordringer. Udover eleverne har der også været fokus på at lærerne skal kunne observere eleverne og se deres udvikling. Lærerne har så fået mulighed for at se elevernes udvikling ved hjælp af tabeller og grafer over tid. Tabellerne og graferne kan sorteres for enkelte baner, brugere og klasser. Ved hjælp af denne mulighed har lærerne chancen for at give eleverne den sværhedsgrad de spiller efter.

Referencer

Spiludvikling

[1] *Folkeskole lære Erik med personlig erfaring med elever i folkeskolen.*

[2] *Computerspil bedre end matematiklærer.*
http://videnskab.dk/kultur-samfund/computerspil-bedre-end-matematiklaerer

[3] *Spilundervisning produceret af Multimedieforeningen.*
http://www.spilundervisning.dk/10grunde.asp

[4] *Unity3D dokumentation.*
http://unity3d.com/learn/documentation.

[5] *Unity3D GUIText til at følge object - kildekode.*
http://wiki.unity3d.com/index.php?title=ObjectLabel

[6] *Unity3D bil object i Unity3D - vejledning.*
http://www.gotow.net/andrew/blog/?page_id=78

[7] *Unity3D optimering*
http://docs.unity3d.com/Documentation/ScriptReference/index.Performance_Optimization.html

[8] *Lyde til spillet.*
http://www.freesound.org

[9] *Flash.*
http://www.adobe.com/devnet/flash.html

[10] *Silverlight.*
http://msdn.microsoft.com/en-us/library/gg130945(v=vs.95).aspx

[11] *HTML5.*
http://www.html5rocks.com

Hjemmeside udvikling

[12] *WordPress dokumentation.*
http://codex.wordpress.org

[13] *User Groups - WordPress plugin til gruppe fordeling.*
http://wordpress.org/plugins/user-groups

[14] *User Role Editor - WordPress plugin til at ændre adgang til de forskellige brugere.*
http://wordpress.org/plugins/user-role-editor

[15] *HighCharts - JavaScript diagram.*
http://www.highcharts.com/

[16] *Lave plugin i WordPress.*
https://codex.wordpress.org/Writing_a_Plugin

[17] *jQuery dokumentation.*
http://docs.jquery.com

[18] *Bootstrap - CSS framework udviklet af Twitter.*
http://twitter.github.io/bootstrap

[19] *PHP dokumentation.*
http://php.net

[20] *AJAX med jQuery.*
http://api.jquery.com/jQuery.ajax

[21] *Poedit til oversættelse.*
http://www.poedit.net

[22] *Drupal CMS.*
https://drupal.org

[23] *FuelPHP framework.*
http://fuelphp.com

[24] *Joomla CMS.*
http://www.joomla.org

[25] *WebSocket.*
http://www.websocket.org

Database udvikling

[26] *MySQL Community Edition.*
http://www.mysql.com/products/community

[27] *MySQL Workbench.*
http://www.mysql.com/products/workbench

[28] *NoSQL.*
http://nosql-database.org

A

Use cases

Der findes tre forskellige roller.

- Elev: Bruger med begrænset tilladelse.
- Lære: Bruger med mere tilladelse, men stadig begrænset
- Admin: Adminstrator af siden.

A.1 Træk talbobbel ned på tallinjen

Rolle: Elev / lære / admin

Scenarie:

- Tryk og træk talbobbel ned på tallinjen.
- Talboblen placeres rigtigt på tallinjen og indekset ændrer sig på tallinjen.
- Talboblen forsvinder.
- Byg-knappen vises.

Alternativ scenarie:

1. Talboblen placeres forkert på tallinjen og talboblen forsvinder, tallinjen nulstilles, byg-knappen forsvinder og byggepointene nulstilles.
2. Talboblen når at falde i vandet og kan ikke trykkes på. Der bliver lavet en ny talbobbel.

A.2 Byg med pointene

Rolle: Elev / lære / admin

Scenarie:

- Byg-knappen vises.

- Byg-knappen trykkes på for at investere pointene.
- Broen bliver bygget.

Alternativ scenarie:

1. Der er ikke nok point til at bygge broen og broen bliver ikke bygget.

A.3 Justering af tallinjen

Rolle: Elev / lære / admin

Scenarie:

- Tryk og træk talbobbel ud i højre side.
- Tallinjen bliver ved med at stige, så længe talboblen er ude til højre.
- Talboblen trækkes ind mod midten.
- Tallinjen stopper med at stige.

Alternativ scenarie:

1. Talboblen trækkes til venstre og tallinjen falder.

A.4 Zoom ind og ud på tallinjen

Rolle: Elev / lære / admin

Scenarie:

- Scroll op ved hjælp af musen.
- Tallinjen zoomer ind ved at mindske tal-intervallet.

Alternativ scenarie:

1. Scroll ned ved hjælp af musen og tallinjen zoomer ud ved at indsætte flere tal på tallinjen.
2. Tallinjen er allerede i det største eller mindste zoom og tallinjen ændrer sig ikke.

A.5 Opret bane

Rolle: Elev / lære / admin

Scenarie:

- Gå til spil og tryk på opret bane.
- Udfyld felterne.
- *Klik prøv for at prøve banen.*
- Klik gem.

Alternativ scenarie:

1. Felterne blev udfyldt forkert og brugeren kan prøve igen.

A.6 Spil en bane

Rolle: Elev / lære / admin

Scenarie:

- Gå til spil og tryk på spil.
- Der vises en tabel ud for hver gruppe. Tabellen indeholder banerne med information.
- Tryk på en bane.
- Spillet starter og kom igang med banen ved at trykke start.

Alternativ scenarie:

1. Der er ingen spil foreløbig og siden er tom.

A.7 Se spilhistorikker

Rolle: Lære / admin

Scenarie:

- Gå til spil og tryk på se spil.
- Mulighed for at ændre visning mellem brugere, baner og grupper.
- En form vises med søgning efter specifik data.
- Tabel vises med spilhistorikker.

- *Tryk på diagram for at få data vist i et diagram.*

Alternativ scenarie:

1. Der kan ikke findes tidligere spil og tabellen er tom.

A.8 Ændre en bane

Rolle: Elev / lære / admin

Scenarie:

- Gå til spil og tryk rediger spil.
- Der vises en tabel ud for hver gruppe. Tabellen indeholder banerne med information
- Tryk på en bane.
- Banens information kommer ind i nogle felter.
- Ret felterne til noget nyt.
- *Klik prøv for at prøve banen.*
- Klik gem for at oprette en ny version af banen.

Alternativ scenarie

- Der findes ingen baner at redigere.
- Det er ikke muligt at ændre en revision af en tidligere bane.

A.9 Slet en bane

Rolle: Elev / lære / admin

Scenarie:

- Gå til spil og tryk rediger spil.
- Der vises en tabel ud for hver gruppe. Tabellen indeholder banerne med information
- Tryk på en bane.
- Tryk på ikonet med slet.
- En advarsel kommer op og tryk slet.

Alternativ scenarie

- Der findes ingen baner at redigere.

A.10 Tilføj brugere

Rolle: Lære / admin

Scenarie:

- Login på siden.
- Tryk på brugernavnet for at komme ind til brugerpanelet.
- Tryk på tilføj ny og udfyld felterne.

Alternativ scenarie:

1. Man har ikke lære eller admin rettigheder.

A.11 Ændre i brugerinformation

Rolle: Lære / admin

Scenarie:

- Login på siden.
- Tryk på brugernavnet for at komme ind til brugerpanelet.
- Tryk på alle brugere og vælg en bruger for at rette.

Alternativ scenarie:

1. Man har ikke lære eller admin rettigheder.

A.12 Tilføj gruppe

Rolle: Lære / admin

Scenarie:

- Login på siden.
- Tryk på brugernavnet for at komme ind til brugerpanelet.
- Tryk på User Groups og indtast navn og tilføj gruppen.

Alternativ scenarie:

1. Man har ikke lære eller admin rettigheder.

A.13 Tilføj grupper til en bruger

Rolle: Lære / admin

Scenarie:

- Login på siden.
- Tryk på brugernavnet for at komme ind til brugerpanelet.
- Tryk på alle brugere og tilføj gruppe til bruger.

Alternativ scenarie:

1. Man har ikke lære eller admin rettigheder.

B

Installation

Produktet er testet med følgende

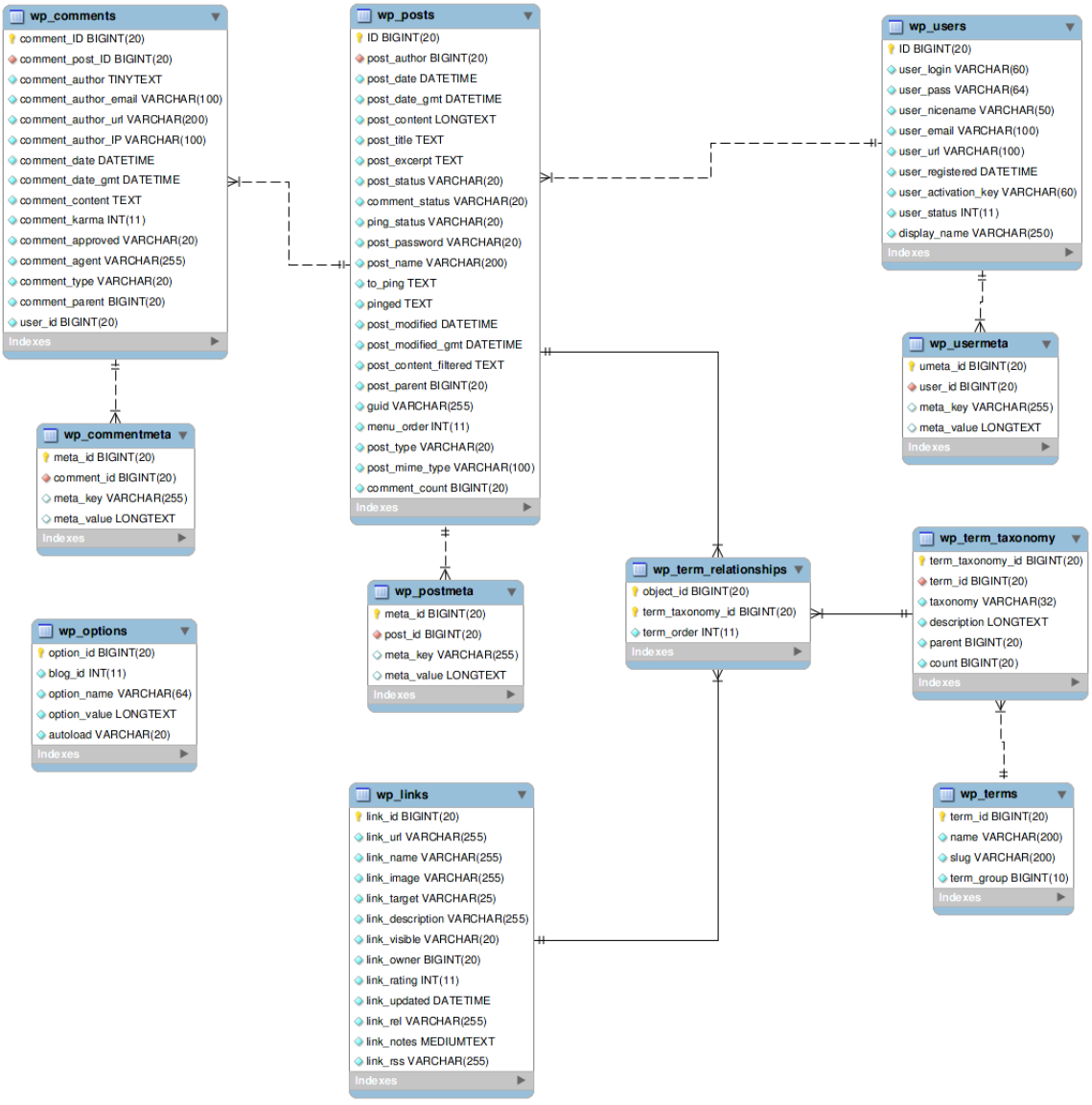
- Apache 2.2.14
- PHP 5.3.1
- phpMyAdmin 3.5.8 med MySQL version 5.1.44

1. Udpak bscf132.zip og læg den op på en web-server.
2. Gå ind i internet browseren og åben siden til web-serveren med produktet.
3. Udfyld database information.
4. Login på siden og gå ind på adminpanelet.
5. Gå til Plugins og aktiver følgende:
    - jQuery Updater
    - User Groups
    - User Role Editor
    - Math Game database
6. Gå til Appearance -> Themes og vælg wpbootstrap af Mads Lundt
7. Gå til Tools -> Import -> WordPress.
8. Vælg backup.xml.
9. Gå til Menus under Appearance og læg menuerne over på lære menu, bruger menu og gæste menu.

WordPress kan sættes til et andet sprog ved at ændre WPLANG i wp-config.php. I denne version er Dansk (da_DK) og engelsk understøttet.

C

Wordpress database

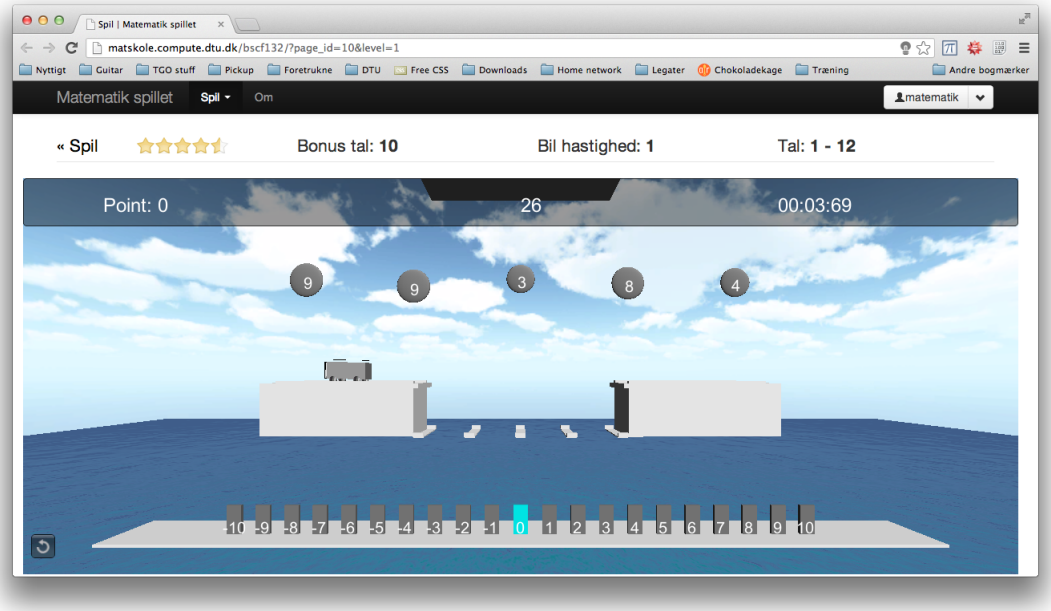


Figur C.1: Wordpress database.

D

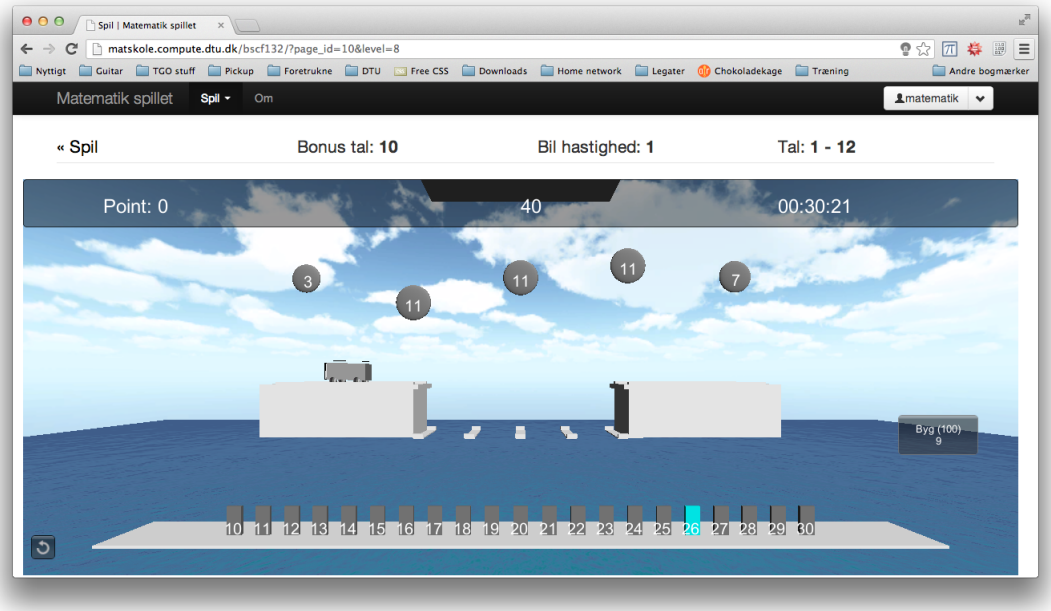
Hvordan spillet skal spilles

Spillet fungerer ved at der er nogle faldende talbobler som skal placeres på tallinjen inden de rammer vandet. Talboblerne kan trækkes i ved hjælp af musen og tallinjen kan justeres ved at trække talboblerne ud over siderne.



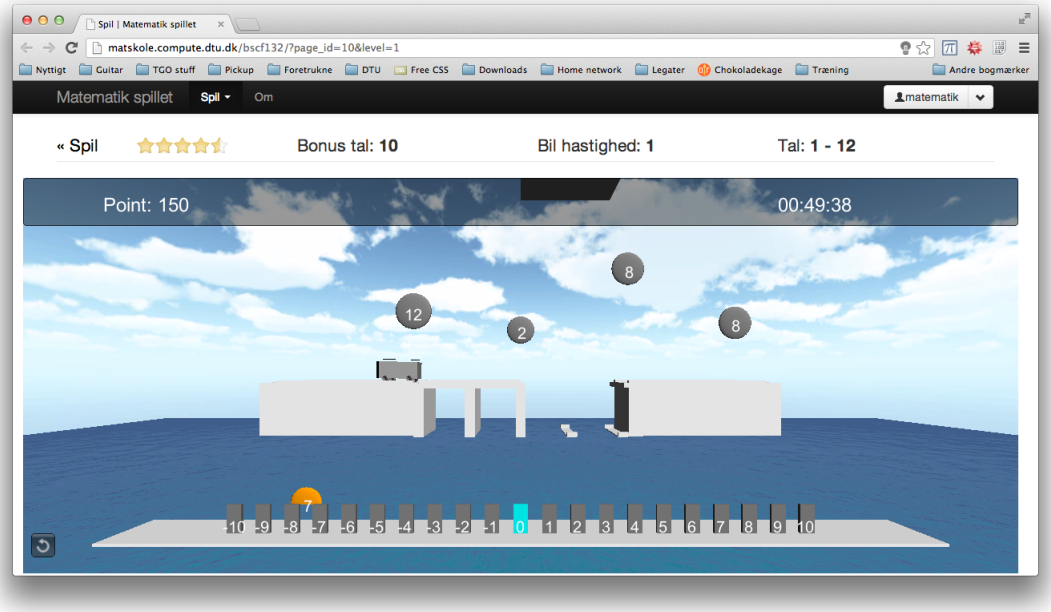
Figur D.1: Spillet startes.

Når talboblerne bliver trukket korrekt ned på tallinjen, så kommer byg-knappen frem. Man kan vælge at bygge sine byggepoint eller satse og få endnu flere byggepoint inden der bygges. Ved at trække flere talbobler ned inden der bliver bygget, så stiger ens byggepointpoint hurtigere. Hvis der fejles ved indsættelse af talbobbel på tallinjen eller byg-knappens tid løber ud, så mister man sine byggepoint.



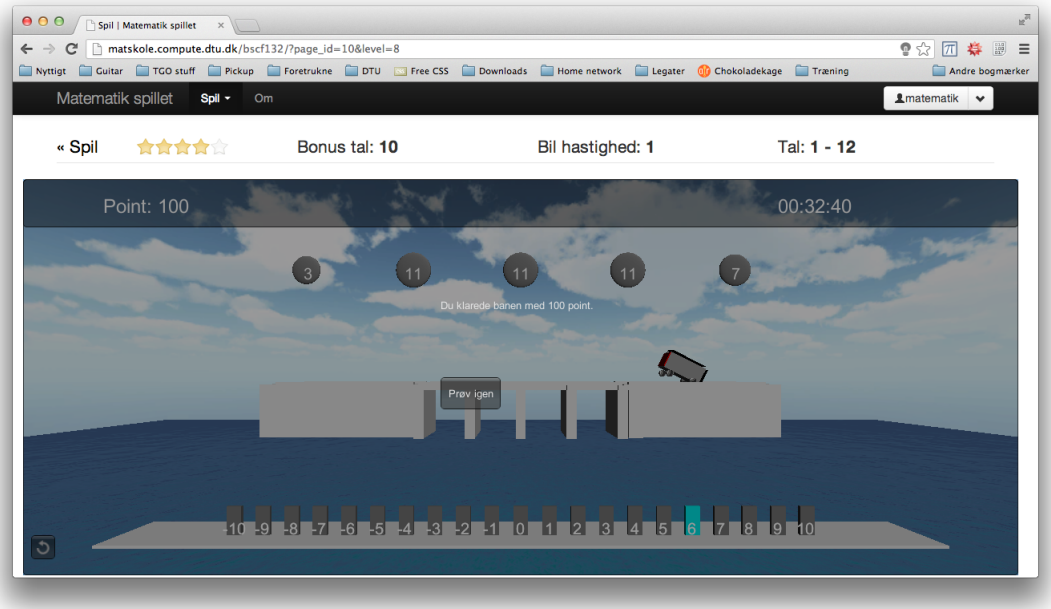
Figur D.2: Tal placeres korrekt på tallinjen.

Når der trykkes på byg-knappen så bliver byggepointene indløst til point i spillet. Hvis man har nok point så bygges noget af broen.



Figur D.3: Broen bliver bygget efter at have trykket byg.

Ved at få bilen sikkert over på den anden side af broen er spillet klaret, men hvis bilen falder i vandet undervejs så er spillet tabt. Det gælder altså om at benytte sig af at få så mange byggepoint inden man bygger for at blive hurtigst muligt færdigt.



Figur D.4: Spil gennemført.

Spillet kan prøves på

matskole.compute.dtu.dk/bscf132

Rolle	**Brugernavn**	**Kodeord**
Elev	user	test
Lære	teacher	test

Tabel D.1: Login til siden.

E

Risikoanalyse

Uge	Aktivitet	Risiko
1	Projekt planlægning	1
2	Projekt planlægning	1
3	Analysering af matematiske spil	1
4	Fortsat analyse samt skitsering af spil	1
5	Start på Unity og påbegyndelse af spil	1
6	Implementering af talbobler	2
7	Implementering af talbobler	2
8	Implementering af talbobler og tallinje	2
9	Implementering af tallinje	2
10	Implementering af database og logging	4
11	Implementering af grafik	2
12	Implementering af grafik	2
13	Implementering af grafik	2
14	Implementering af levels	2
15	Implementering af levels	2
16	Implementering af levels og små rettelser	2
17	Test	5
18	Test	5
19	Rapport skrivning	1
20	Rapport skrivning	1
21	Rapport skrivning	1
22	Rapport skrivning	1

www.compute.dtu.dk

Institut for Matematik og Computer Science
Danmarks Tekniske Universitet
Matematiktorvet, bygning 303 B
2800 Kgs. Lyngby
Danmark
Tel: (+45) 45 25 30 31
Fax: (+45) 45 88 13 99
E-mail: compute@compute.dtu.dk