**Opgavesæt til skriftlig eksamen i Distribueret Programmering 1. juni 2015**

Vægtning af spørgsmål:

Opgave 1.1 3 points

Opgave 1.2 3 point

Opgave 1.3 4 points

Opgave 2.1 3 points

Opgave 2.2 3 points

Opgave 2.3 3 points

Opgave 2.4 3 points

Opgave 3.1 4 points

Opgave 3.2 4 points

Opgave 4.1 5 points

Opgave 4.2 5 points

Opgave 5.1 15 points

Opgave 5.2 10 points

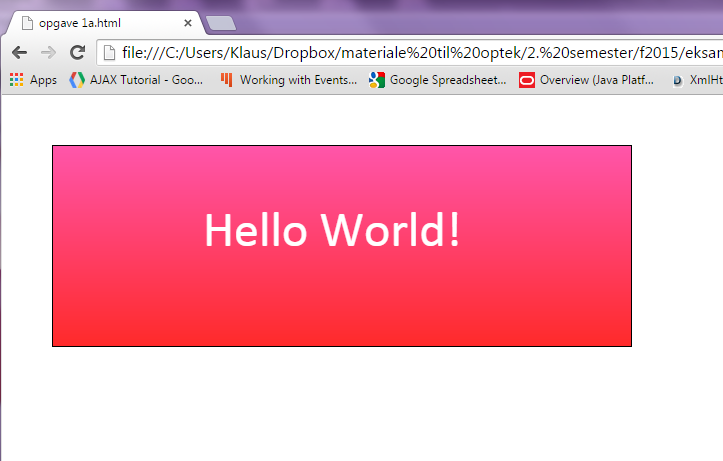
Opgave 5.3 15 points

Opgave 6 20 points

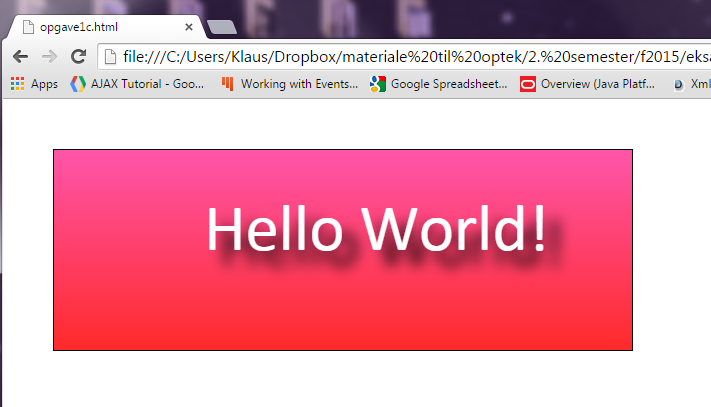
Alle hjælpemidler er tilladt

**Opgave 1**

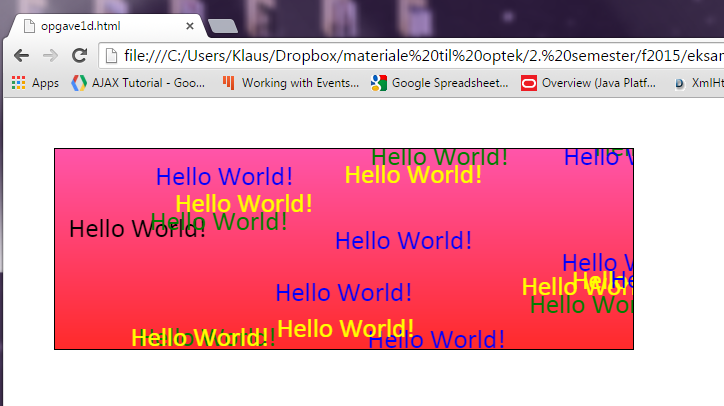
1. Brug canvas og css til at programmere dette billede (eller noget, der ligner nogenlunde). Teksten har font size 40px og font family Calibri. Farvegradienten løber lineært fra boksens øvre kant til dens nedre kant uden farvestop undervejs. Gem din besvarelse i opgave1a.html.



1. Tag udgangspunkt i opgave1a.html og tilføj en sort skygge til teksten, som vist på nedenstående screendump. Du skal ikke bestræbe dig på at ramme dette grafiske udtryk på screendumpet præcist. Gem din besvarelse i opgave1b.html.

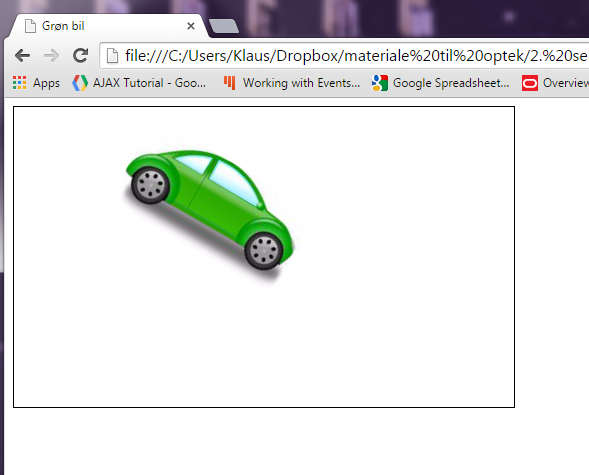


1. Tag igen udgangspunkt i opgave1a.html. Opret et array, hvis elementer er farverne ’black’, ’yellow’, ’blue’ og ’green’. Opret JavaScript-kode, som skriver teksten 20 gange i boksen Teksten må gerne rage ud over boksen, og de enkelte eksemplarer må gerne overskrive hinanden. De enkelte tekster skal have tilfældige begyndelseskoordinater indenfor boksen. Tekstfarven skal i hvert eksemplar af teksten være en tilfældigt valgt farve i det oprettede farvearray. Nedenfor er vist et eksempel på, hvordan output kan være. Gem din besvarelse i opgave1c.html.

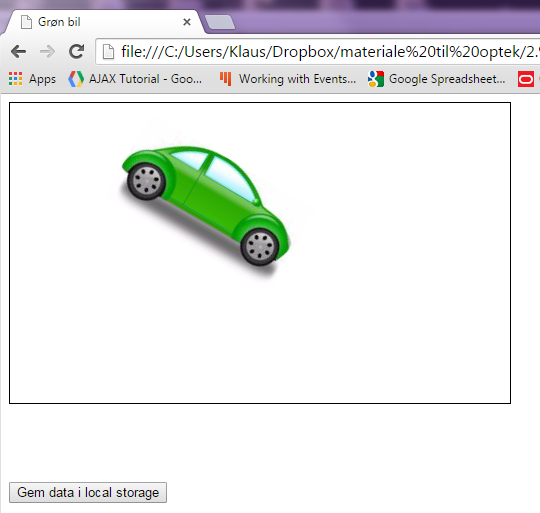


**Opgave 2**

1. I dokumentet opgave2Template.html er der en image-tag med source = img/car.jpg. Visibility for denne tag er sat til hidden, så billedet ikke ses. Tag udgangspunkt i dette dokument og brug fabric.js til at tegne billedet af bilen i det canvas, som også er defineret i opgave2Template.html. Billedet skal have attributterne left sat til 200, top til 100 og angle til 30, se nedenstående figur. Gem din besvarelse i opgave2a.html.

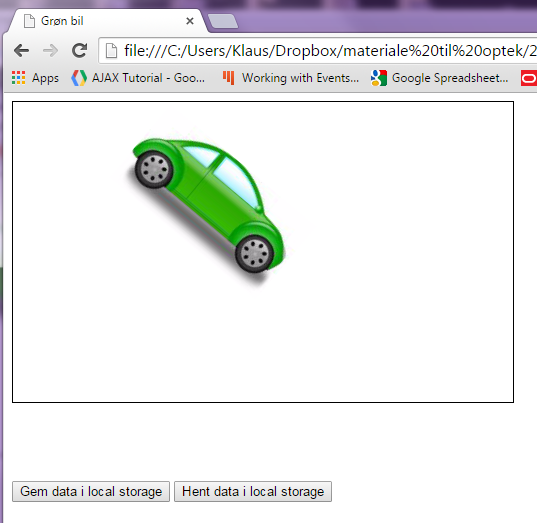


1. Tilføj en button til dokumentet og skriv ved hjælp af jQuery en kode, som bevirker, at billedets vinkel (angle) gemmes i local storage, når man klikker på knappen, se nedenstående figur. Gem din besvarelse i opgave2b.html.



1. Billedets vinkel (angle) er sat til 30, altså et heltag (integer). Imidlertid gemmer local storage alle data i string format, så når vi skal bruge denne gemte værdi, skal den konverteres til integer igen. Find evt. ved hjælp af en Google-søgning ud af, hvilken JavaScript-metode der konverterer string til integer.

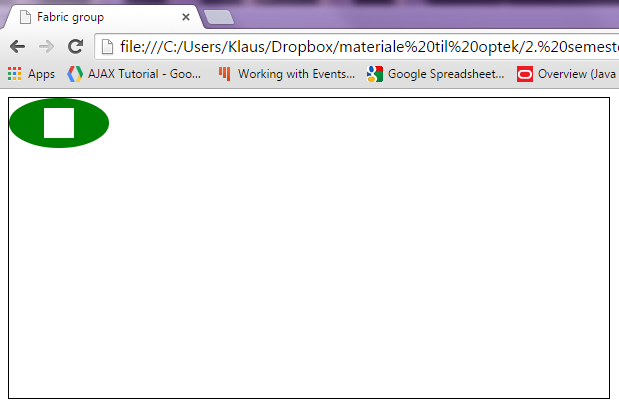
Tilføj nu en ny button, og skrive ved hjælp af jQuery en kode, som bevirker, at værdien for billedets vinkel hentes fra local storage. Læg 10 til denne værdi og gentegn canvas. Gem din besvarelse i opgave2c.html.



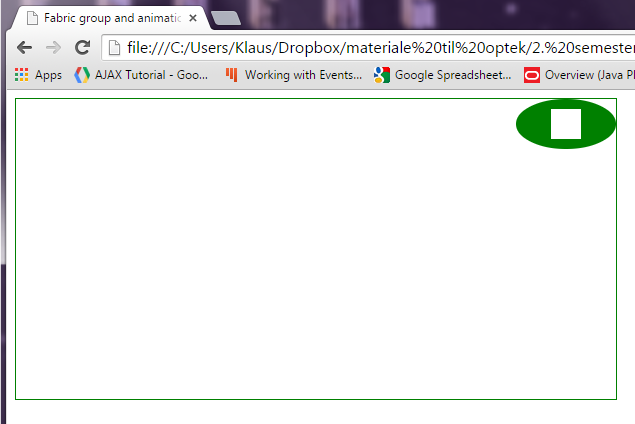
1. Skriv videre på koden i opgave2c.html, således, at bilen drejer 10 grader, hver gang man klikker på knappen med teksten ”Hent data i local storage”. Gem din besvarelse i opgave2d.html.

**Opgave 3**

1. Opret ved hjælp af fabric.js en cirkel og et kvadrat. Cirklen har radius 50 og en skaleringsfaktor i y-aksens retning på 0.5, mens kvadratet har sidelængden 30. Cirklen (eller rettere sagt ellipsen) er grøn, og kvadratet er hvidt. De to figurer knyttes til en group, således at de begge anbringes i midten af gruppen. Kvadratet er foran cirklen. Justér left og top for denne group, således at figuren er anbragt oppe i venstre hjørne af canvas. Canvas har width 600 og height 300. Gem din besvarelse i opgave3a.html. Du kan se et løsningsforslag på nedenstående figur.

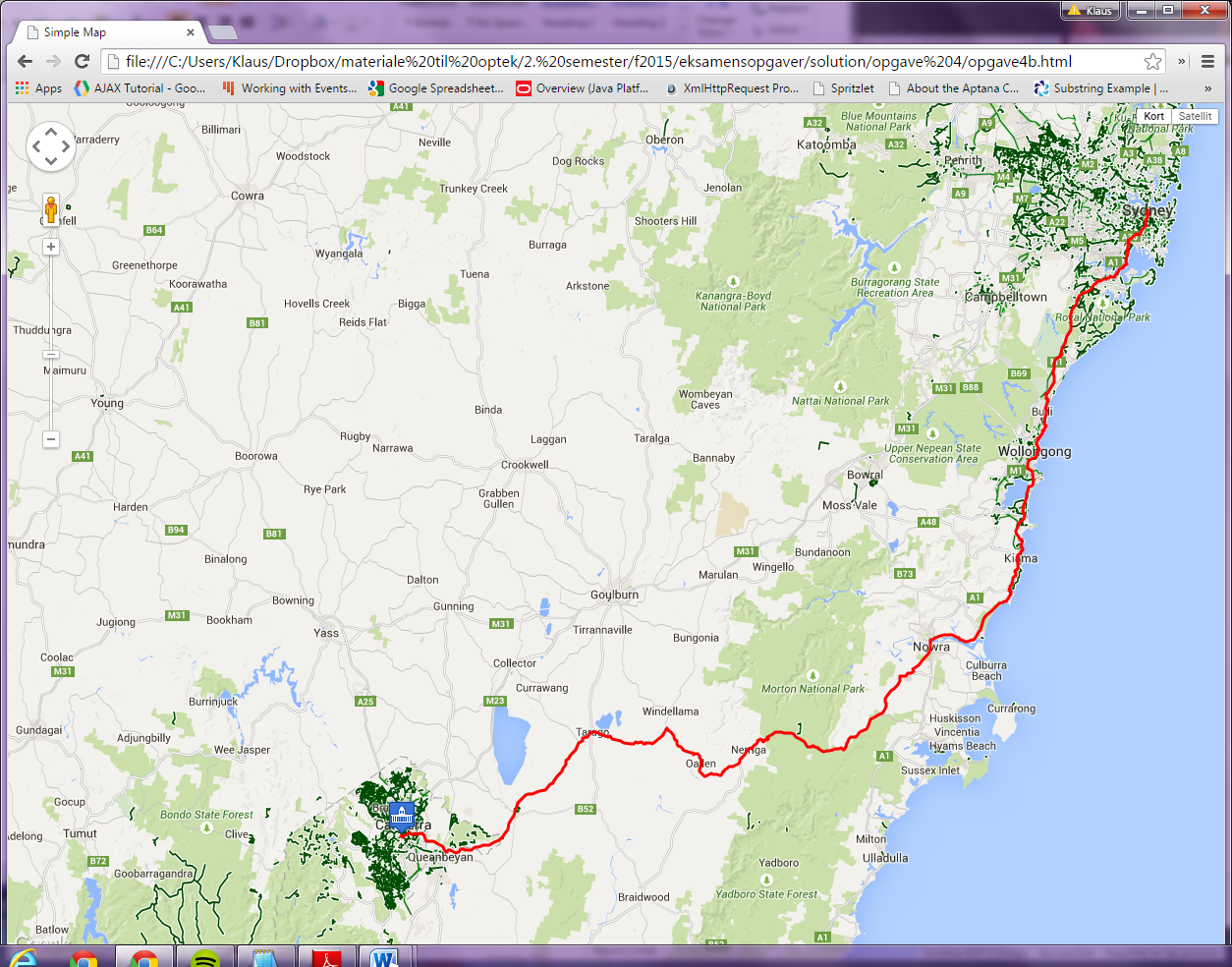


1. Opret en animation, som flytter gruppen vandret over til canvas’ højre kant. Du må gerne bruge din viden om, at width for canvas er 600, til at ”hardcode” animationen. Animationen skal vare 6 sekunder og skal være med easing sat til easeInBounce. Gem din besvarelse i opgave3b.html. Du kan se et løsningsforslag på nedenstående figur.



**Opgave 4**

1. I dokumentet opgave4Template.html findes koden til at fremstille et kort over den sydøstlige del af Australien. Brug dette som udgangspunkt til at sætte en ikon (img/capital.png) i Canberra. Lav dernæst en rollover-effekt, så capital.png skiftes ud med capital\_rollover.png ved mouseover og omvendt ved mouseout. Gem din besvarelse i opgave4a.html.
2. Skriv dernæst kode, så cykelruten mellem Canberra og Sydney indtegnes, når man klikker på ikonet i Canberra. Ruten har standardfarven blå. Lav denne om til rød. Der anbringes også standardmæssigt to markører, A og B, i start- og slutpunkterne for ruten. Skriv kode, som fjerner disse markører. Du skal evt. bruge netsøgninger for at løse dette punkt. Gem din besvarelse i opgave4b.html. På nedenstående figur kan du se en løsning af opgaven.



**Opgave 5**

1. Lav en side til oprettelse af 2 typer af data (se tabel 1 og 2). Bemærk ved tilføjelse af data fra tabel 2 (spil) skal det være muligt at vælge tilknytningsforhold til eksisterende data fra tabel 1 (virksomhed). Alt data skal hentes og gemmes på jeres egen cloud storage hos Parse.com og alt data fra tabel 1 og 2 skal oprettes i jeres Parse.com cloud storage.
2. Lav en Bootstrap navigation bar der kan navigere til hovedside, login side og signup side. Hovedside må gerne være tom, alternativt kan den kombineres med side fra Spørgsmål 3. Det er valgfrit om I vil anvende Single Page Application navigation eller standard page navigation.

Login skal indeholde en standard login formular hvor brugeren kan indtaste password og username. Login processen foretages ved hjælp af Parse.com.

Signup skal indeholde en standard signup formular hvor brugeren kan indtaste password, email og username. Signup processen foretages ved hjælp af Parse.com.

1. Lav en side til at vise virksomheder og deres spil. Siden skal bestå af 2 ruder hvor den venstre rude viser en liste af spilvirksomheder (behøver kun at vise navn) og den højre rude viser 2 spil. Ved tryk på et virksomhed i liste (venstre rude) skal den højre rude opdateres med spil information (navn og udgivelses årstal). Information om virksomheder og spil skal hentes fra jeres egen cloud storage hos Parse.com.

Siden skal være mobiloptimeret således at når browservinduet er mindre end 768px (Bootstrap grid col-sm) rykker den højre rude under den venstre.

**Opgave 6**

Lav en hjemmeside der ved hjælp af Google Maps viser gps-lokationerne for de 5 største spil virksomheder for årene 2011 til og med 2014. GPS koordinaterne skal hentes fra jeres cloud storage på Parse.com. Markører skal kun vises for et år af gangen og skal kunne skiftes ved hjælp af knapper eller lignende. For at løse dette spørgsmål kan det være nødvendigt at foretage en søgning af, hvordan man fjerner en marker fra en map.

# Data til opgaver ([baseret på Top 25 companies by Game revenues](http://www.newzoo.com/free/rankings/top-25-companies-by-game-revenues/))

**Tabel 1 - Forskellige spilvirksomheder**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Virksomhed** | **2014\*** | **2013\*** | **2012\*** | **2011\*** | **GPS latitude** | **GPS longitude** |
| Sony | 2 | 3 | 2 | 1 | 51.350149 | -0.462472 |
| Activision Blizzard | 5 | 4 | 1 | 2 | 53.357836 | -6.226448 |
| Microsoft | 3 | 2 | 3 | 3 | 47.639623 | -122.128374 |
| EA | 4 | 5 | 4 | 4 | 37.522571 | -122.253832 |
| Nintendo | 9 | 7 | 6 | 5 | 34.969752 | 135.756198 |
| Tencent | 1 | 1 | 5 | 6 | 22.539307 | 113.955727 |

\*Placering for det år ud fra indtægter.

**Tabel 2 - Spil produceret af de forskellige virksomheder**

|  |  |  |
| --- | --- | --- |
| **Spil** | **Virksomhed** | **Udgivelses årstal** |
| Starcraft 1 | Activision Blizzard | 1998 |
| Warcraft 1 | Activision Blizzard | 1994 |
| Borderlands 2 | Sony | 2012 |
| Gran Turismo 4 | Sony | 2005 |
| Halo 3 | Microsoft | 2007 |
| Gears of War | Microsoft | 2006 |
| FIFA Soccer 12 | EA | 2011 |
| MADDEN NFL 2014 | EA | 2003 |
| League of Legends | Tencent | 2009 |
| Unreal Tournament | Tencent | 2014 |
| Super Mario Bros | Nintendo | 1985 |
| Tetris | Nintendo | 1989 |