# Ny Plan

1. Games: Cookie Clicker mer personalizerd

~~Sjakk, Rider, Jetpack joyride, Candy crush?, cookie clicker, gacha game~~

Jeg lager en standard cookie cliker spill, men fokuset er på hvor spesiell opplevelsen skal være for den enkelte. Mye informasjon om den spesifikke personen er mest viktig i denne.¨

«Biscuit Clicker»^TM

## Få gjort og Få lært

Gjort:

* Base game (Biscuit)
  + Funksjonalitet
    - Antal pr click
    - Automatisk farming
      * Offline mining?
    - Prestige?
    - Andre steder?
    - <https://www.youtube.com/watch?v=I6Dez_6bbb4>
  + Logo og fargepalett
    - Designmanual er ikke viktig
    - Identit
  + Gacha System
    - Rewards:
      * Permanent Boost?
      * Forskjellige fargepallett
  + Mini tutorial?
* Database:
  + Login
    - Kan spille uten å logge inn
    - Faktisk Konto med lagret data
      * Settings
        + Dark mode / Light mode?
        + Save button?
        + Auto save option
        + Toggle ingame tips
        + <https://www.gamedeveloper.com/design/create-better-game-settings-options-handy-checklist->
      * Progress
      * Lagret «items»
      * Bytte passord og sånt
  + Lagring av data method:
    - Main game: Save as javascript? Or convert to php?
    - Setting: PHP save?
    - Items: PHP save?
* Styling:
  + Å få siden til å se bra ut
  + Ingen detaljer
* Dokumentasjon (PDF):
  + IT-bruker
  + Sluttbruker

Få lært:

* Database
  + Tables med relations
    - (Her lagres alt, føler dette kan være en bra løsning).
  + PHP
    - Hente Data
    - Lagre data
    - Communication med Javascript / Base game ?

Hvis jeg kan:

* Achievements
* More upgrades:
* Lyd og desgin

## Ukeplan

Uke 1 (Fra med Uke 43) + Praksis Uker:

* Base game:
  + Gain on Click
  + Gain more on upgrade
  + When enough prestige
  + When prestige you get gems
  + Gems to gacha
  + Layout
    - CSS
    - Logo Fargepalett og typografi
  + Lære
* Ukeplan:
  + Mandag: Sett opp av andre nettsteder: Login og registerasjon
  + Tirsdag: Settings og see på summons (how to)
  + Onsdag: Items
  + Torsdag og Fredag: Fortsettelse

Uke 2 (Neste Termin uke):

* Database
  + Login
  + Sett up av Tabeler
  + Lagring av data

Uke 3 og inntil videre:

* Følge plannen øvert

# ~~Config alters~~

~~Motion.conf:~~

~~Log\_file /home/log/motion/motion.log~~

~~Target\_dir /Pictures/MotionPictures~~

~~Picture\_filename %Y%m%d---%H%M%S~~

~~Movie\_filename %t-%v-%Y%m%d---%H%M%S~~

# Prosjektbeskrivelse

Skriv noen linjer som beskriver løsningen din i boksen nedenfor.

(Hva slags teknologi vil du bruke, hvilket kodespråk, hva slags behov skal løsningen din dekke, hvem er målgruppen din og hva slags HW trenger du?)

|  |
| --- |
| 1. ~~Sikkerhetskamera system~~   ~~Denne løsningen er for de som trenger et sikkert sikkerhetssystem hjemme, jobbe eller annet. Fordi, de er kanskje redde for innbrudd, må passe på barn etter de drar eller er spent på hva som skjer når de er borte. Men håndteringen av kamera og program. Trenger så langt: et kamera, database og programvare. Trenger ikke noe spesielt kodespråk når det gjelder konfigurering. Dette kan man gjøre med Linux og motion.~~  Sikkerhetskamera system kunne ha vært et interessant opplegg og gøy, men oppgaven er litt for avansert for meg. Derfor gi opp. En grunn var kanskje oppsette vår. Jeg trenger å få kamera linket til hyper-v Virutal Machine. Etter hvert har jeg funnet ut at det kan, men har blitt fortalt at jeg kan ikke mest. Koble   1. Games: Cookie Clicker mer personalizerd   ~~Sjakk, Rider, Jetpack joyride, Candy crush?, cookie clicker, gacha game~~  Jeg lager en standard cookie cliker spill, men fokuset er på hvor spesiell opplevelsen skal være for den enkelte. Mye informasjon om den spesifikke personen er mest viktig i denne. |
|  |

# ~~PLAN 1~~

~~Hva slags oppgaver må du gjøre for å komme i mål?~~

|  |  |
| --- | --- |
| ~~Ting jeg må gjøre~~ | ~~Ting jeg må lære~~ |
| * ~~Sette opp kamera~~ * ~~Lære det jeg trenger, må og kan.~~ * ~~Motion eller Motion Pluss?~~ * ~~Database setup:~~   [~~https://motion-project.github.io/motionplus\_examples.html#database\_setup~~](https://motion-project.github.io/motionplus_examples.html#database_setup)  [~~https://www.linux.com/training-tutorials/how-operate-linux-spycams-motion/~~](https://www.linux.com/training-tutorials/how-operate-linux-spycams-motion/)  [~~https://www.linkedin.com/pulse/setting-up-surveillance-camera-linux-using-motion-ratan-mohapatra/~~](https://www.linkedin.com/pulse/setting-up-surveillance-camera-linux-using-motion-ratan-mohapatra/)   * ~~Chat-gpt~~ | * ~~Motion:~~   ~~Nedlasting~~  ~~Konfigurere kamera~~  ~~Forskjellig resolusjoner~~  ~~«Motion detection»~~  ~~Brannmur og det~~  ~~Hvem har til gang~~  ~~Repository til bilder /video~~  ~~Bilde eller video?~~  ~~Kan se gjennom kamera?~~ |
|  |  |

## ~~Ukeplan~~

* Uke 1: Planlegging
  + Plan for terminoppgaven
  + Lese Motion guides, videoer eksempler
  + Kamera set-up
  + Hva skal jeg overvåke?
  + Starte?
    - 1. Install motion

<https://www.linkedin.com/pulse/setting-up-surveillance-camera-linux-using-motion-ratan-mohapatra/>

* + - 2. Kamera og drivere.
* Uke 2: Første steg
  + Motion Konfigurering
    - Følge guides
    - Bilde eller video?
    - Lagring av bilder/video (Hvilke repositroy)
    - Live sending fra pc?
    - Hvem har tilgang til kamera
    - Sett opp MariaDB database
  + Brukerstøtte
    - Notering av prosses underveis. Det blir til en tururoail til slutt.
* Uke 3: Spesialering
  + Flere kameraer?
    - Ip kamera
    - Standard USB-kamera
  + Konfigurere til egen personlig tilstander
    - Brannmur
    - Tilgang
    - Lagring
    - Mer av det samme
  + Mer notering
* Uke 4: Finalisering
  + Forstå alt det jeg har gjort og kunne reprodusere resultatene
  + Siste sjekk med notering og dokumentring.

## ~~Motion (fra chat gpt)~~

Setting up the "Motion" software on Linux for camera monitoring and motion detection involves several steps. Here's a general guide to help you get started:

1. Install Motion:

First, you need to install the Motion software. The installation process may vary depending on your Linux distribution. Here are some common commands for popular package managers:

For Debian/Ubuntu:

```bash

sudo apt-get install motion

```

1. Configure Motion:

Motion's configuration file is typically located at `/etc/motion/motion.conf`. You can use a text editor to open and modify it:

```bash

sudo nano /etc/motion/motion.conf

```

Customize the configuration according to your needs. Here are some key settings to consider:

- `daemon`: Set it to `on` to run Motion as a daemon in the background.

- `width` and `height`: Set the resolution of your camera.

- `framerate`: Adjust the frame rate.

- `stream\_port`: Specify the port for the live stream (default is 8081).

- `stream\_localhost`: Set to `off` if you want to access the stream remotely.

- `target\_dir`: Configure where captured images and videos will be stored.

Save your changes and exit the text editor.

3. Create the Target Directory:

- Ensure the directory specified in the `target\_dir` option exists and that the Motion process has write permissions to it.

For example:

```bash

sudo mkdir -p /var/lib/motion

sudo chown motion:motion /var/lib/motion

```

***Dette kan ertsattes med MariaDB ?!***

1. ***Restart*** / Start Motion\*\*:

Start the Motion service using your distribution's service manager:

For systemd-based systems (e.g., Ubuntu 16.04+):

```bash

sudo systemctl start motion

``

You can also enable Motion to start at boot:

```bash

sudo systemctl enable motion # for systemd-based systems

```

5. Access the Live Stream:

- Open a web browser on the same machine where Motion is installed and navigate to `http://localhost:8081` (or the port you configured). You should see the live camera feed.

6. Access Motion's Web Interface (Optional):

- You can access Motion's web-based configuration interface at `http://localhost:8080`. This interface allows you to fine-tune motion detection and other settings.

7. Configure Motion Detection (Optional):

- Customize the motion detection settings in Motion's web interface or directly in the `motion.conf` file. This includes sensitivity, detection areas, and actions to take when motion is detected.

8. View Logs:

- Check the logs to monitor the operation of Motion. The logs are usually located in `/var/log/motion/` or as specified in the configuration file.

That's it! You should now have Motion up and running on your Linux system, capturing camera feeds and performing motion detection according to your configured settings.

## ~~Motion med MariaDB~~

To add a MariaDB database to your Motion setup, you can use the MySQL storage feature in Motion. This allows you to store information about captured events, such as motion-triggered images or videos, in a MariaDB (MySQL) database. Here's how to set it up:

1. Install MariaDB Server:

- If you haven't already, you need to install MariaDB on your Linux server. You can typically do this using your distribution's package manager:

For Debian/Ubuntu:

```bash

sudo apt-get install mariadb-server

```

Follow the prompts to set up a root password and other configuration options during the installation process.

2. Create a Database and User:

Log in to your MariaDB server as the root user:

```bash

sudo mysql -u root -p

```

Enter the root password when prompted.

- Create a new database for Motion. Replace `<database\_name>` with your preferred database name:

```sql

CREATE DATABASE <database\_name>;

```

- Create a dedicated user for Motion with privileges on this database. Replace `<username>` and `<password>` with your desired username and password:

```sql

CREATE USER '<username>'@'localhost' IDENTIFIED BY '<password>';

GRANT ALL PRIVILEGES ON <database\_name>.\* TO '<username>'@'localhost';

FLUSH PRIVILEGES;

```

Exit the MariaDB shell:

```sql

EXIT;

```

3. Configure Motion to Use MariaDB:

- Edit the Motion configuration file (typically located at `/etc/motion/motion.conf`):

```bash

sudo nano /etc/motion/motion.conf

```

- Find the following lines and configure them with your MariaDB database information:

```ini

sql\_log\_image on

sql\_log\_snapshot on

sql\_log\_mpeg on

sql\_log\_timelapse on

sql\_query INSERT INTO security(camera, filename, frame, file\_type, time\_stamp, event\_time\_stamp) VALUES('%t', '%f', '%q', '%n', '%Y-%m-%d %T', '%C');

sql\_dbname <database\_name>

sql\_host localhost

sql\_user <username>

sql\_passwd <password>

```

Replace `<database\_name>`, `<username>`, and `<password>` with the database name, username, and password you created in step 2.

Save the changes and exit the text editor.

4. Restart Motion:

- Restart the Motion service to apply the changes to the configuration:

For systemd-based systems (e.g., Ubuntu 16.04+):

```bash

sudo systemctl restart motion

```

For sysvinit-based systems (e.g., CentOS 6):

```bash

sudo service motion restart

```

Now, Motion is configured to log captured events into your MariaDB database. It will store information about motion events, including camera details, file names, timestamps, and event timestamps in the specified database. You can use SQL queries to retrieve and analyze this data as needed.