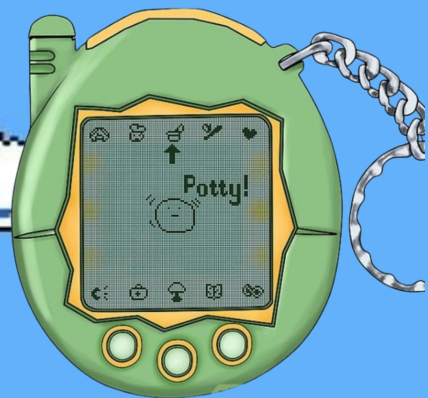


INF-1400  
17.02.2026  
Anne Bosch





# Plan for dagen



- Oversikt over arbeidskrav 2
  - Kravene til løsningen deres
  - Hva skal levers og hvor
  - Vurdering
- Versjonskontroll med git
- Arbeidsprosess mot GitHub
- Oppgave for denne uka



Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

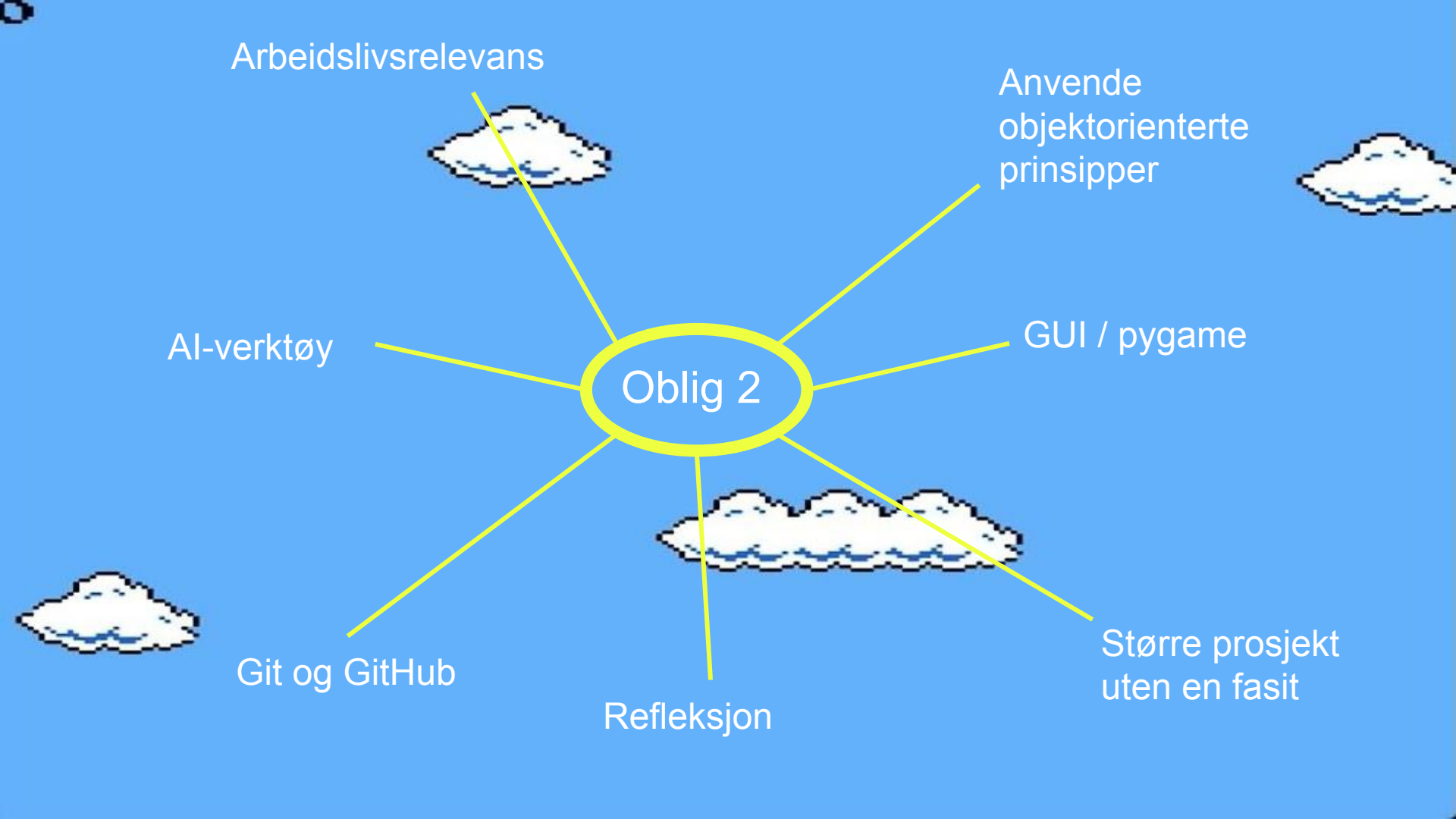
AI-verktøy

Oblig 2

Git og GitHub

Refleksjon

Større prosjekt  
uten en fasit



Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

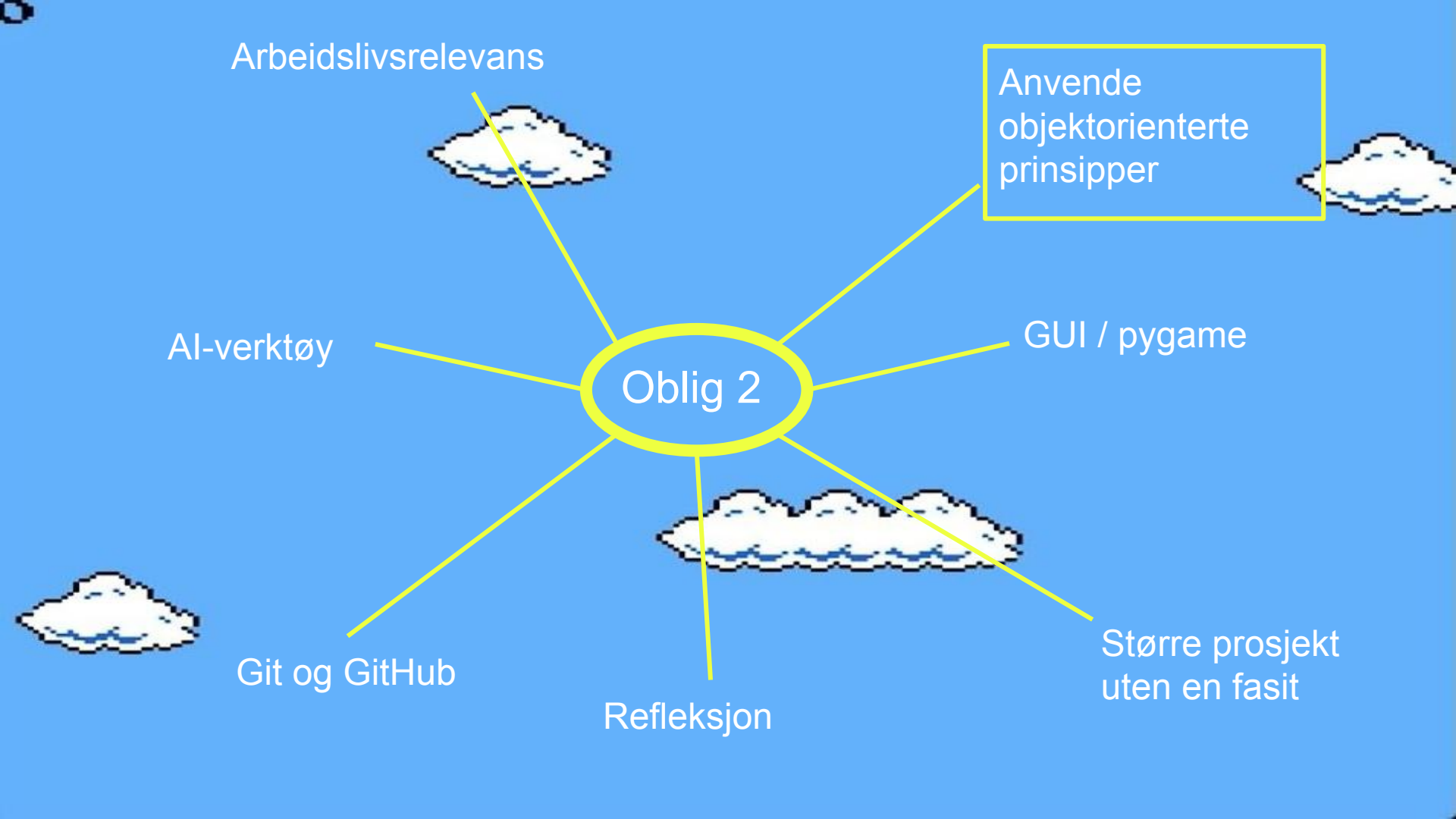
Oblig 2

AI-verktøy

Git og GitHub

Refleksjon

Større prosjekt  
uten en fasit



# Objektorienterte prinsipper

- Klasser, instanser og objektreelasjoner
- Arv mellom klasser
- Komposisjon
- Polymorfi
- Abstrakte baseklasser
- Informasjonsgjemming og tilgangskontroll (encapsulation)

Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

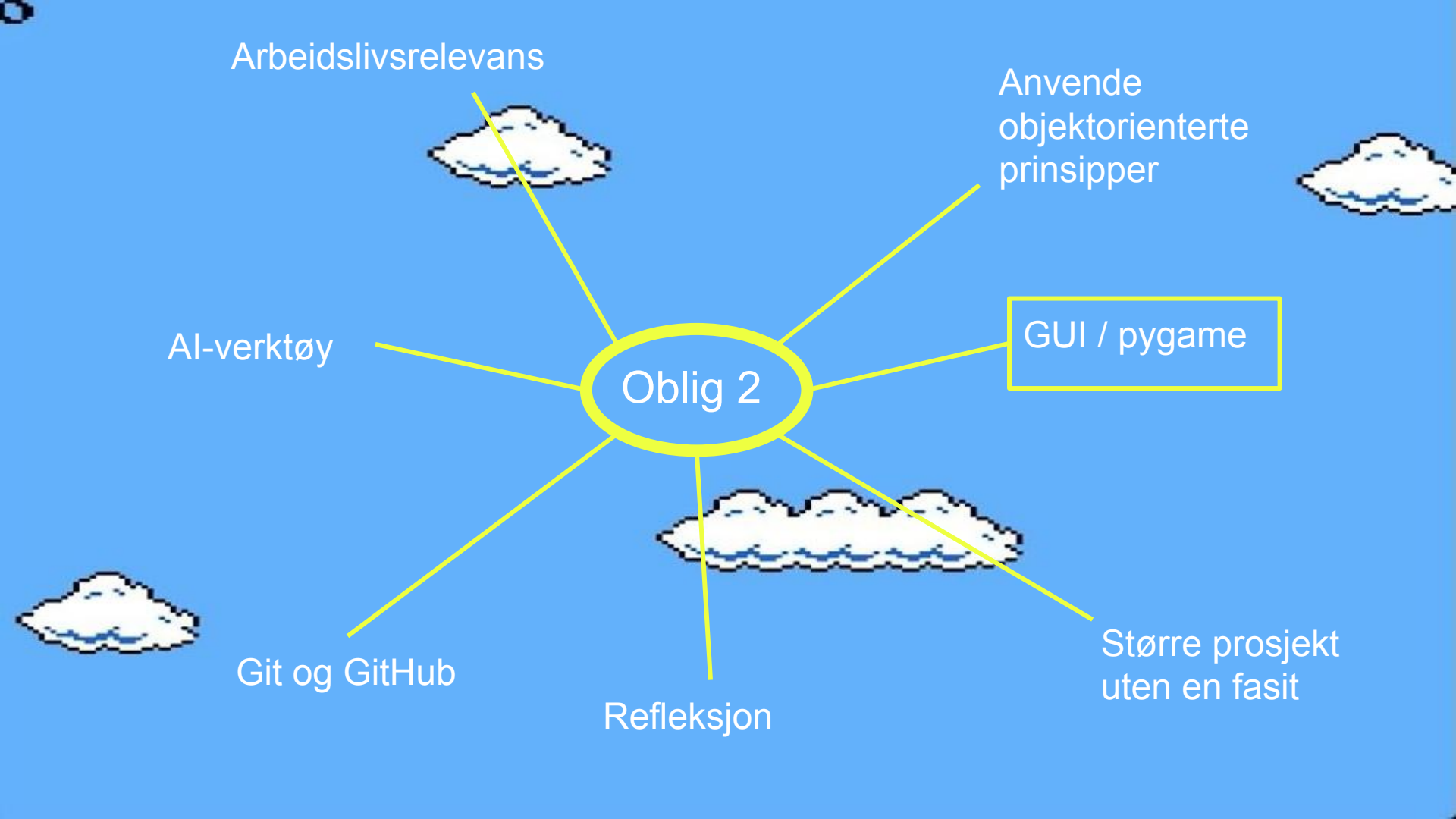
Oblig 2

AI-verktøy

Git og GitHub

Refleksjon

Større prosjekt  
uten en fasit



Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

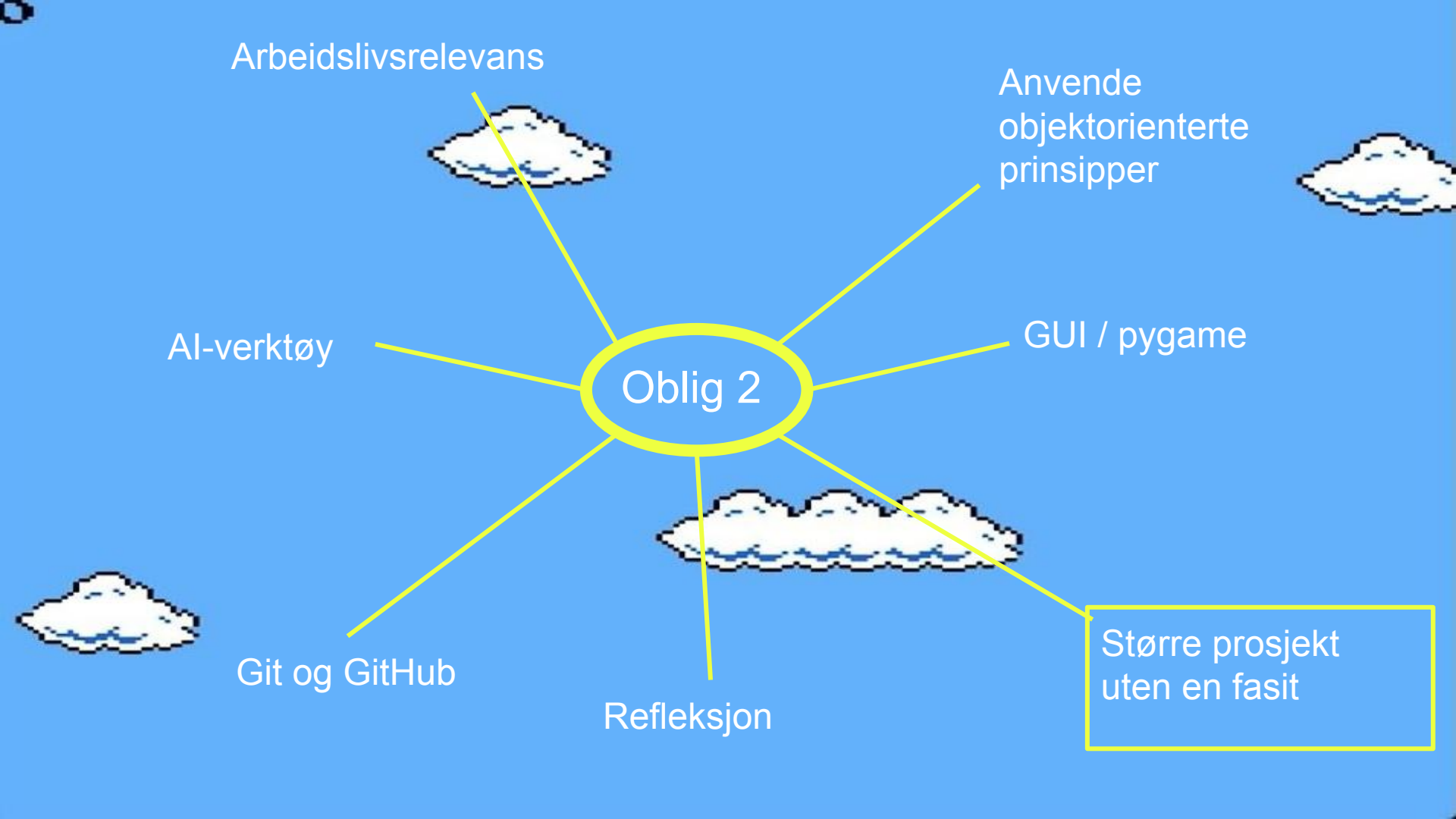
AI-verktøy

Oblig 2

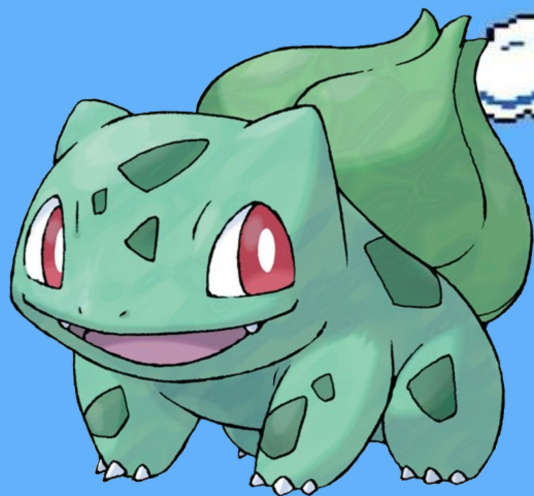
Git og GitHub

Refleksjon

Større prosjekt  
uten en fasit



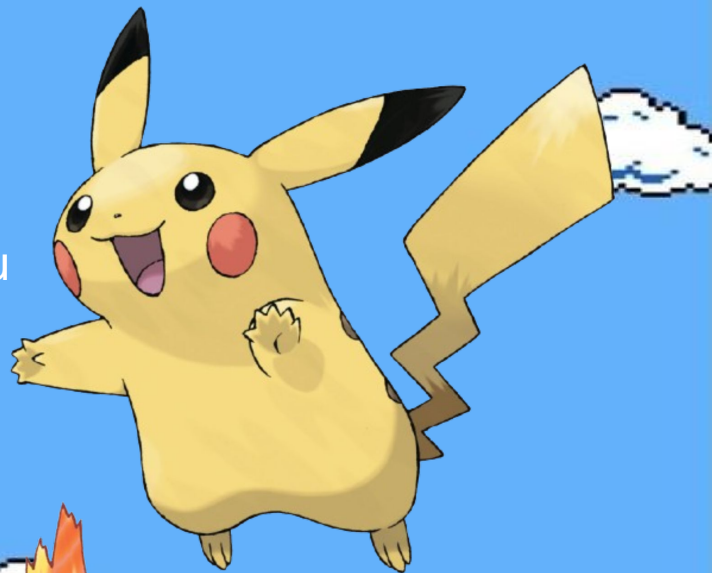
Krav til løsningen dere skal utvikle





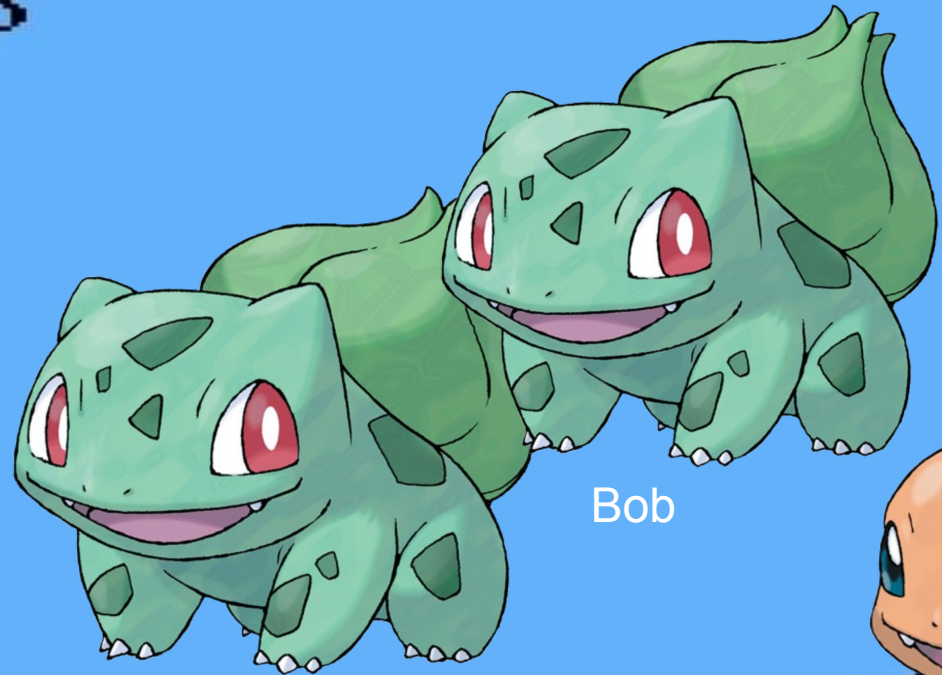
Bulbasaur

Pikachu



Charmander

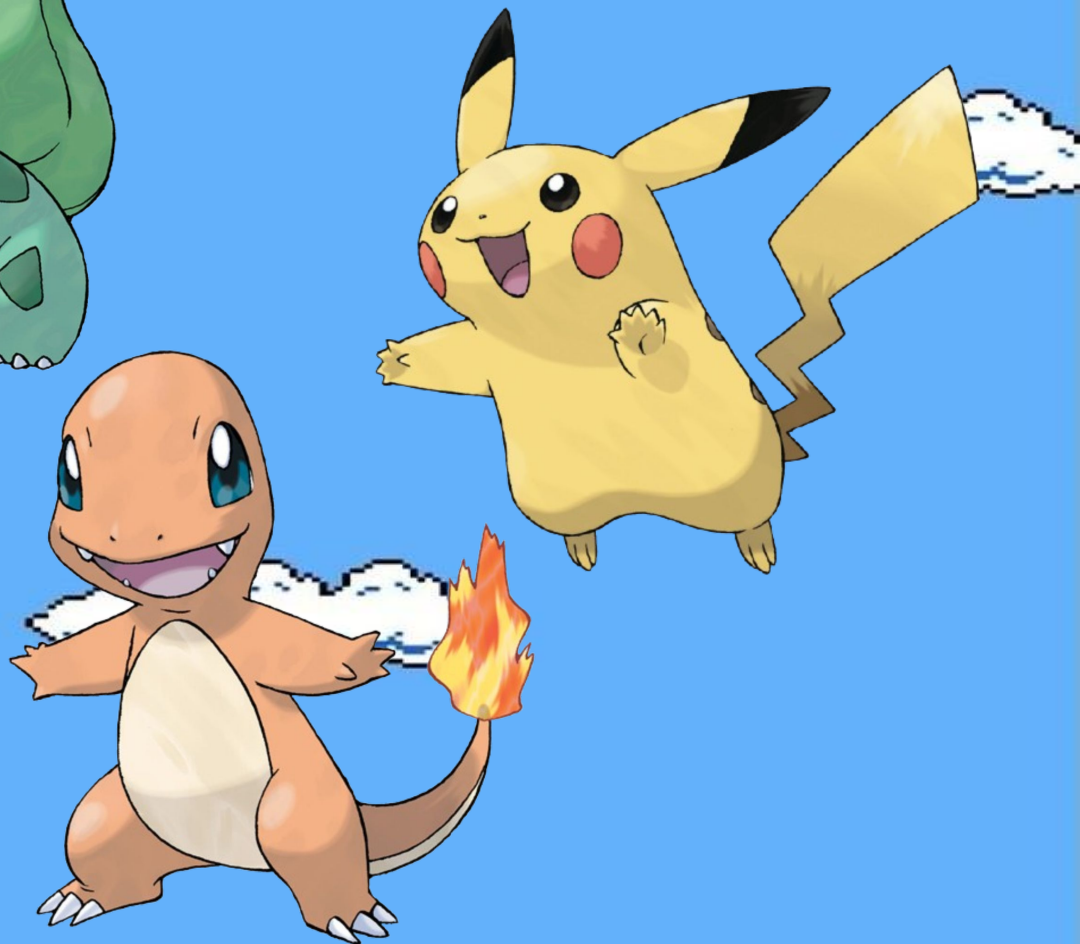
Arter / species



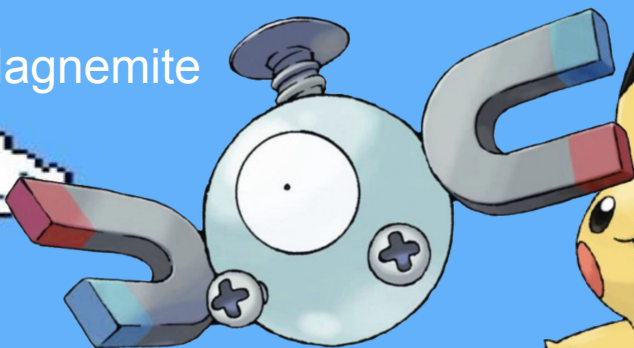
Bob

Fred

Flere Pokémon av  
samme art



Magnemite

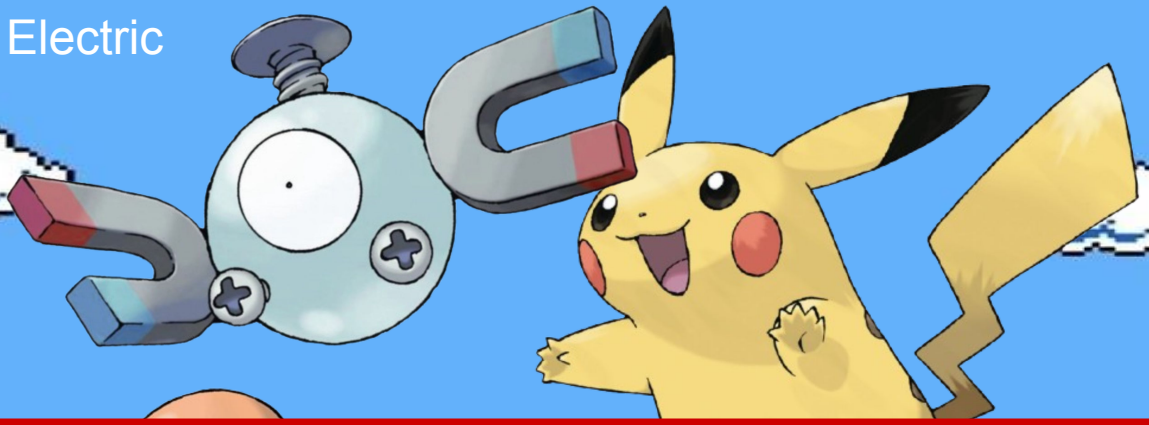


Rapidash

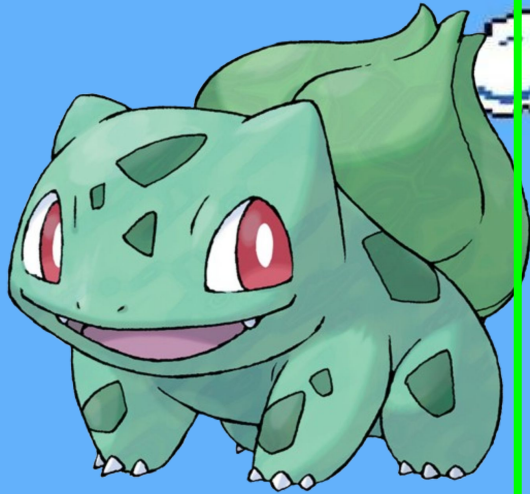


Minst 3 ulike arter

Electric



Grass



Minst 3 ulike typer

Fire



Fred

157 sek



Behov



Energy



Nutrition



Happiness

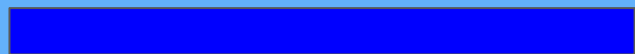
Humør/tilstand: Fornøyd

Fred

157 sek



Behov



Energy



Nutrition



Happiness

Humør/tilstand: Fornøyd

157 sek

Fred



Tiden går



verdien til behovene synker

Behov



Energy



Nutrition



Happiness

Humør/tilstand: Fornøyd

200 sek

Fred



Tiden går

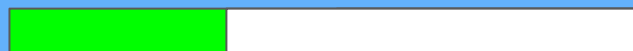


verdien til behovene synker

Behov



Energy



Nutrition

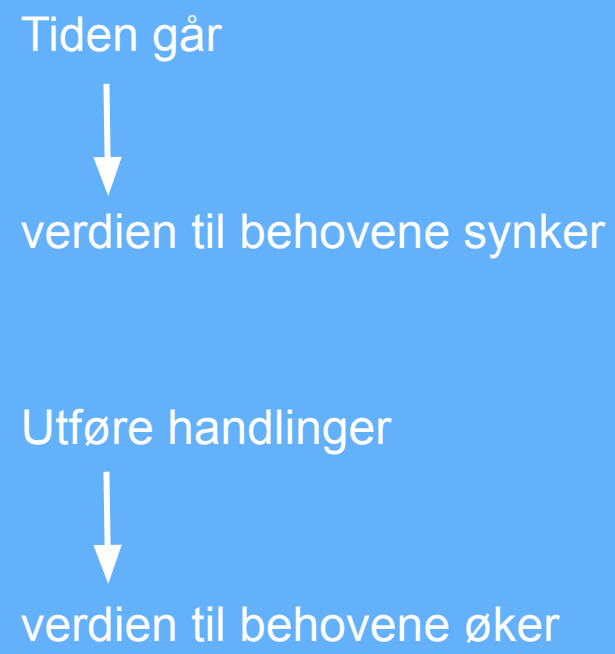


Happiness

Humør/tilstand: Urolig



	Behov
<div><div></div></div>	Energy
<div><div></div></div>	Nutrition
<div><div></div></div>	Happiness
Humør/tilstand: Urolig	





Behov	
<div><div></div></div>	Energy
<div><div></div></div>	Nutrition
<div><div></div></div>	Happiness
Humør/tilstand: Urolig	

Tiden går

↓

verdien til behovene synker

Utføre handlinger

↓

verdien til behovene øker

**fred.gi\_mat()**



Behov	
<div><div></div></div>	Energy
<div><div></div></div>	Nutrition
<div><div></div></div>	Happiness
Humør/tilstand: Nøytral	

Tiden går

↓

verdien til behovene synker

Utføre handlinger

↓

verdien til behovene øker

**fred.gi\_mat()**

210 sek

Luke



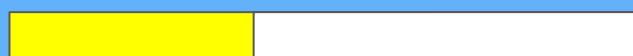
Behov



Energy



Nutrition



Happiness

Humør/tilstand: Tilfreds

Tiden går



verdien til behovene synker

Utføre handlinger



verdien til behovene øker

**luke.gi\_mat()**

210 sek

Luke



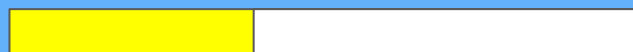
Behov



Energy



Nutrition



Happiness

Humør/tilstand: Tilfreds

Tiden går



verdien til behovene synker

Utføre handlinger



verdien til behovene øker

**luke.gi\_mat()**



# Base case requirements


- Simulator - ta vare på deres Pokémon
  - Grafisk brukergrensesnitt - pygame
  - Interaksjon
  - Minst 3 ulike arter
  - Minst 3 ulike typer
  - Behov
  - Mulig å skille Pokémon-instanser av samme art
  - Tidsaspekt
  - Pokémon reagerer på interaksjon
- Første versjon
  - Felles krav



# Utvidelser


1

2



0025 **Pikachu**


ELECTRIC



0001 **Bulbasaur**


GRASS  
POISON

Evolution chart




#0001  
**Bulbasaur**  
Grass · Poison

→  
(Level 16)



#0002  
**Ivysaur**  
Grass · Poison

→  
(Level 32)



#0003  
**Venusaur**  
Grass · Poison

3

4



DEFENSE → ATTACK ↓	NOR	FIR	WAT	ELE	GRA	ICE	FIG	POI	GRO	FLY	PSY	BUG	ROC	GHO	DRA	DAR	STE	FAI
NORMAL													1/2	0			1/2	
FIRE		1/2	1/2		2	2						2	1/2		1/2		2	
WATER			2	1/2		1/2			2				2		1/2			
ELECTRIC				2	1/2	1/2				0	2					1/2		
GRASS		1/2		2		1/2		1/2	2	1/2		1/2	2		1/2		1/2	
ICE		1/2	1/2		2	1/2			2	2					2		1/2	
FIGHTING	2					2		1/2		1/2	1/2	1/2	2	0		2	2	1/2
POISON					2			1/2	1/2				1/2	1/2			0	2



Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

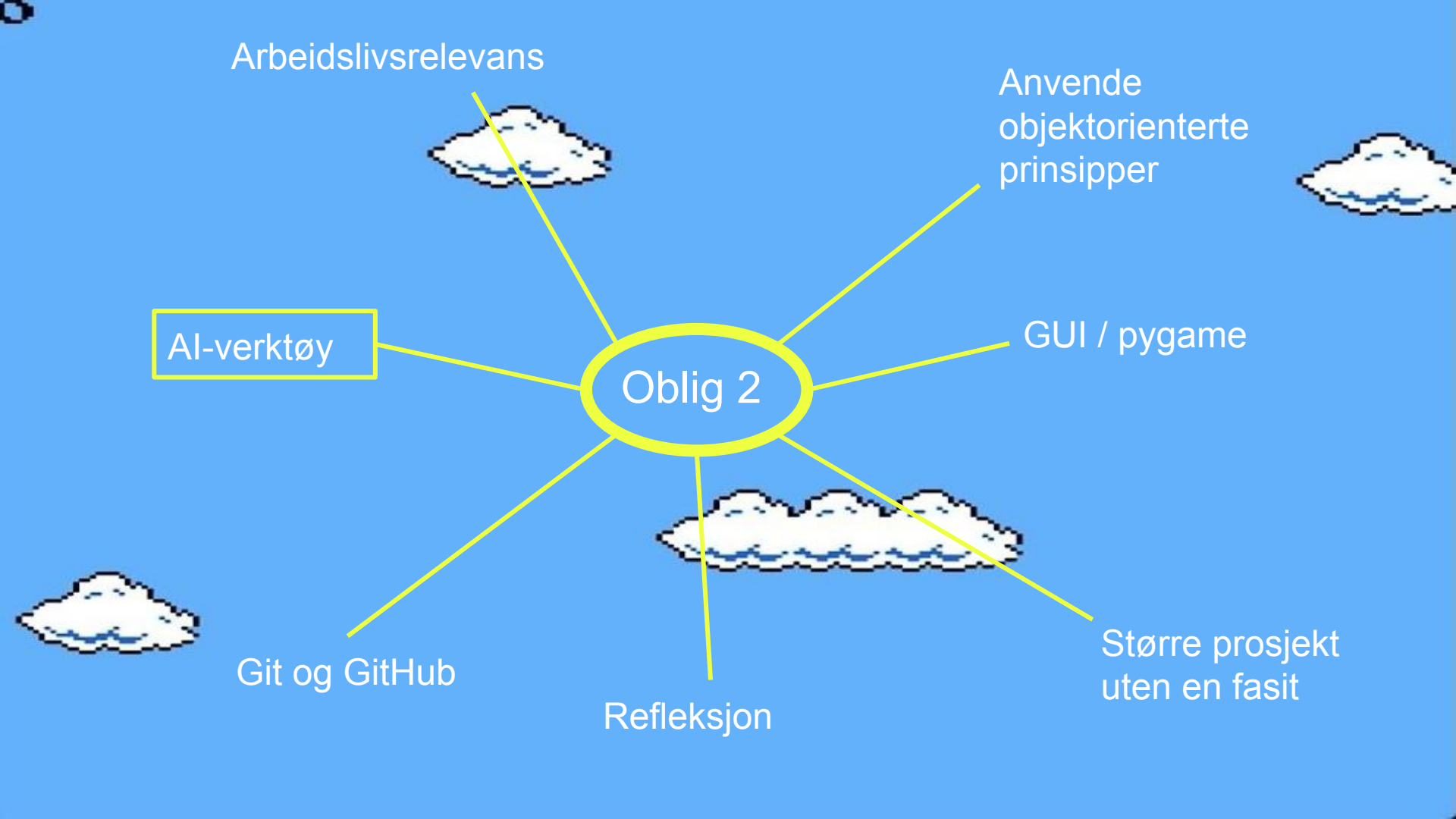
AI-verktøy

Oblig 2

Git og GitHub

Refleksjon

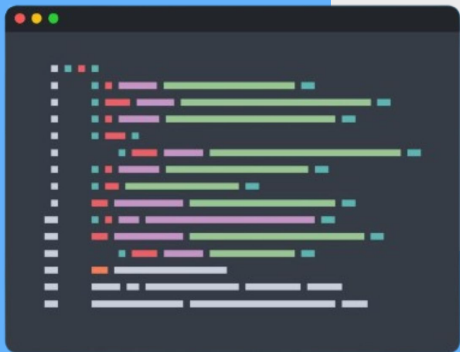
Større prosjekt  
uten en fasit



# KI-bruk

## 4. Avansert KI-samarbeid

Du kan delegere tydelig avgrensede oppgaver (f.eks. utkast til avsnitt, skrijving av dataense-skript, koding, simuleringer, utforming av visuelle plakater, generering av bilder/video/musikk, oversettelse av eget utkast). Verifisering er obligatorisk.



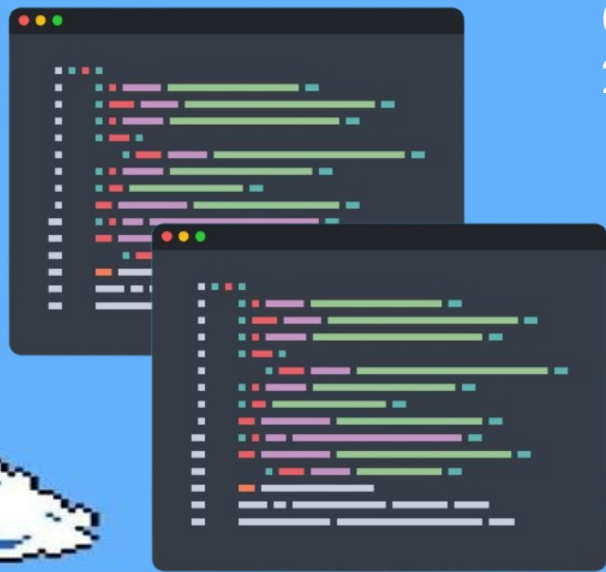
Ulike løsninger

Fredagens forelesning

# Leveranser - 3 deler

10 minutter  
fagsamtale

Gjennomføring  
23. - 27. mars



Kode - 2 versjoner

Frist: 20. mars

Rapport



# Fagsamtaler



- Muntlig vurdering
- Objektorientert programmering
- Triller terning tema
- Ikke presentasjon
- Hjelpelæreren stiller åpne spørsmål
- Designvalg - forståelse og begrunne
- Drøfting av ulike muligheter
- Lete etter kunnskapen deres



- 10 minutter
- Individuelt
- 23. - 27. mars
- Booking - førstemann til mølla
- Fysisk/digital
- Bestått/ikke bestått
- Info canvas
- Tilrettelegging
- Utfordringer og mestring
- Ute i jobb



# Vurdering - Objektorienterte prinsipper

<b>Klasser, instanser og objektrelasjoner</b>	<b>Arv mellom klasser</b>	<b>Komposisjon</b>	<b>Polymorfisme</b>	<b>Abstrakte baseklasser</b>	<b>Informasjons- gjemming og tilgangsstyring (encapsulation)</b>
Design av klasser og prosjektstruktur					
Instansiering av objekter					
Relasjoner mellom klasser					
Tilstandsending					
Ansvarsfordeling mellom klasser					

# Rapporten

- Designvalg
- Drøfte egen AI-bruk
- Profesjonelle utviklerpraksiser - Versjonskontroll
- Ikke AI-genereres

Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

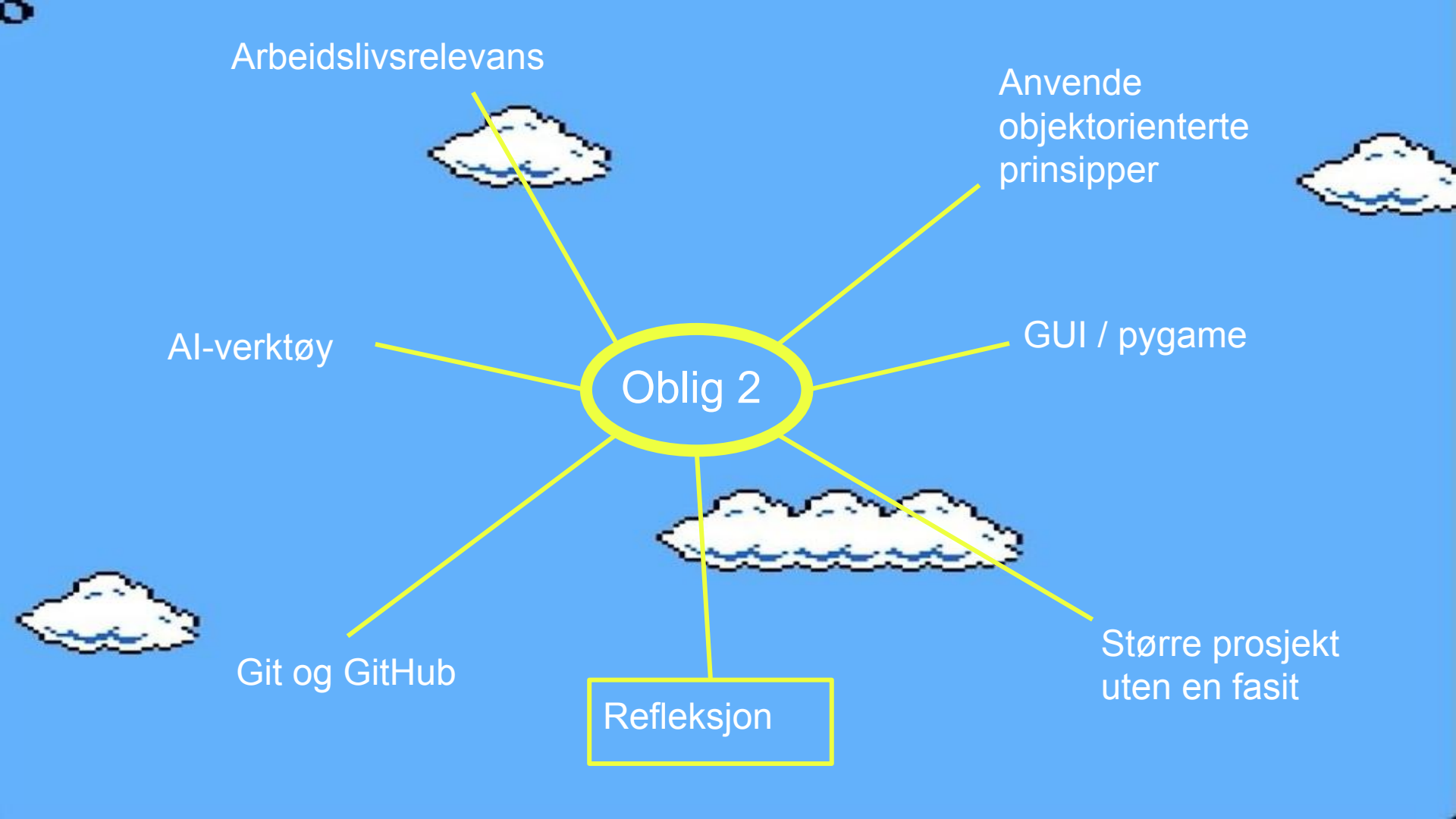
AI-verktøy

Oblig 2

Git og GitHub

Refleksjon

Større prosjekt  
uten en fasit



Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

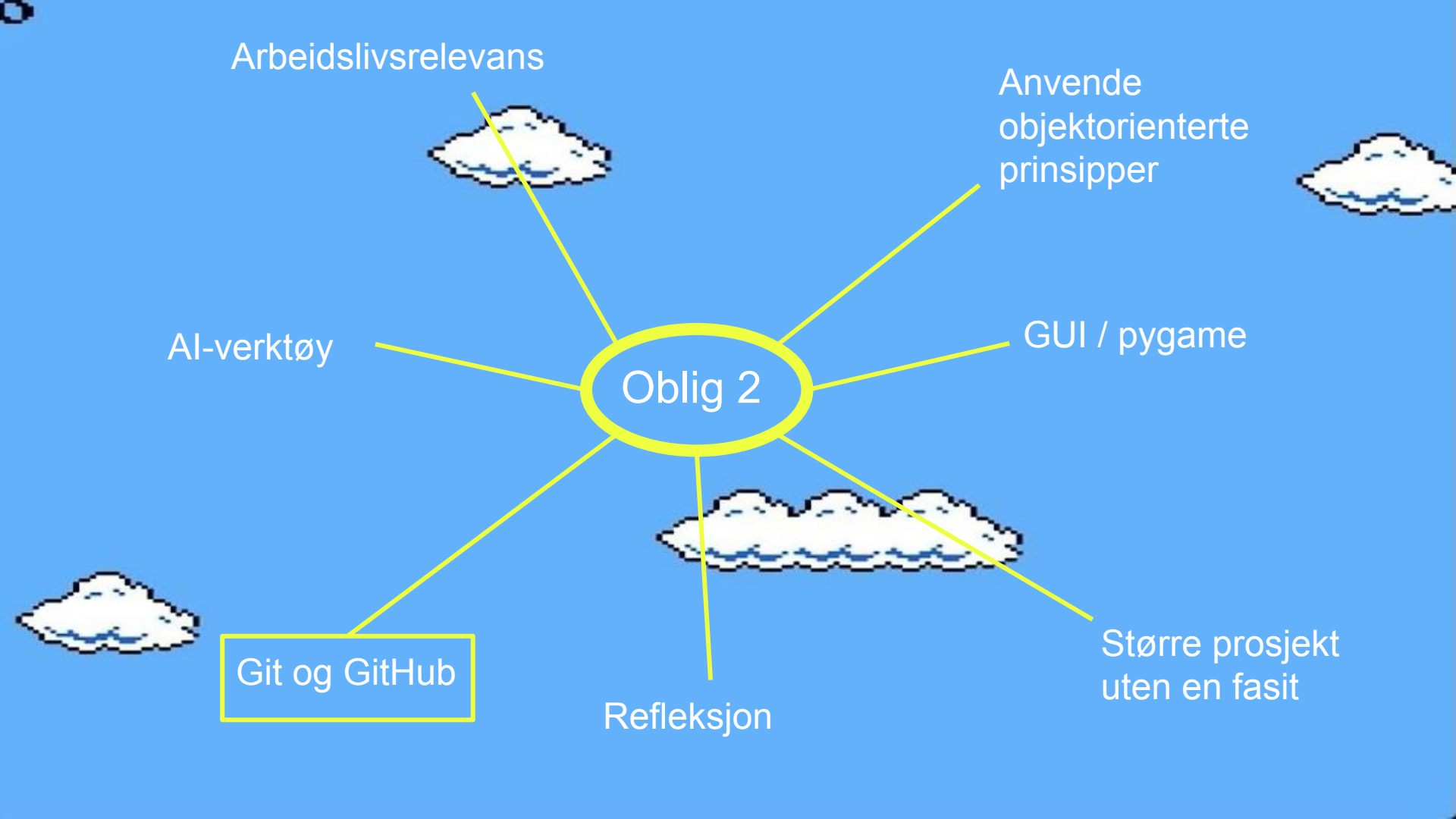
AI-verktøy

Oblig 2

Git og GitHub

Refleksjon

Større prosjekt  
uten en fasit



# Hvor skal dere levere?

- GitHub (Classroom)
- `report.md`
- Koden i prosjektet
- Hjemmeoppgave denne uka - sette opp git og GitHub

THIS IS GIT. IT TRACKS COLLABORATIVE WORK  
ON PROJECTS THROUGH A BEAUTIFUL  
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

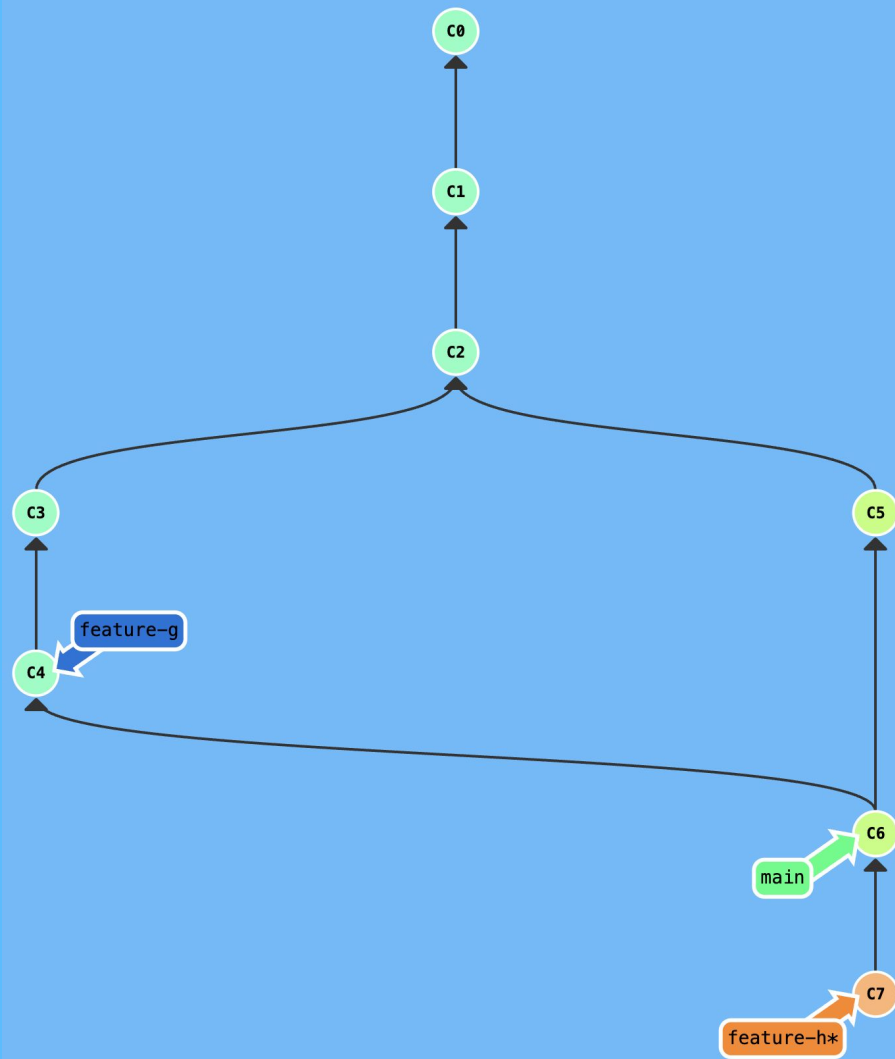
NO IDEA. JUST MEMORIZE THESE SHELL  
COMMANDS AND TYPE THEM TO SYNC UP.  
IF YOU GET ERRORS, SAVE YOUR WORK  
ELSEWHERE, DELETE THE PROJECT,  
AND DOWNLOAD A FRESH COPY.



# Hva er git?

- Versjonskontrollsystem
- Kontroll over: ulike versjoner av din kode
- Repository
- gitignore
- Oversikt over endringer over tid
- Hoppe mellom versjoner





# Hva er git?



- Versjonskontrollsystem
- Kontroll over: ulike versjoner av din kode
- Repository
- gitignore
- Oversikt over endringer over tid
- Hoppe mellom versjoner
- Linus Torvalds
- 2005
- 2008

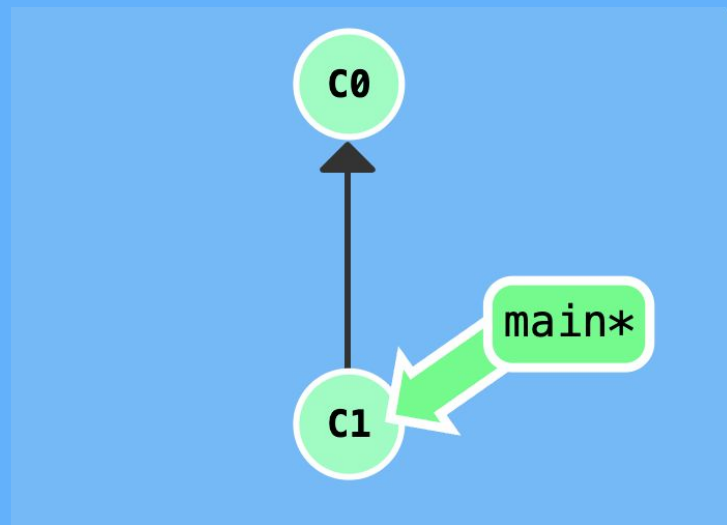


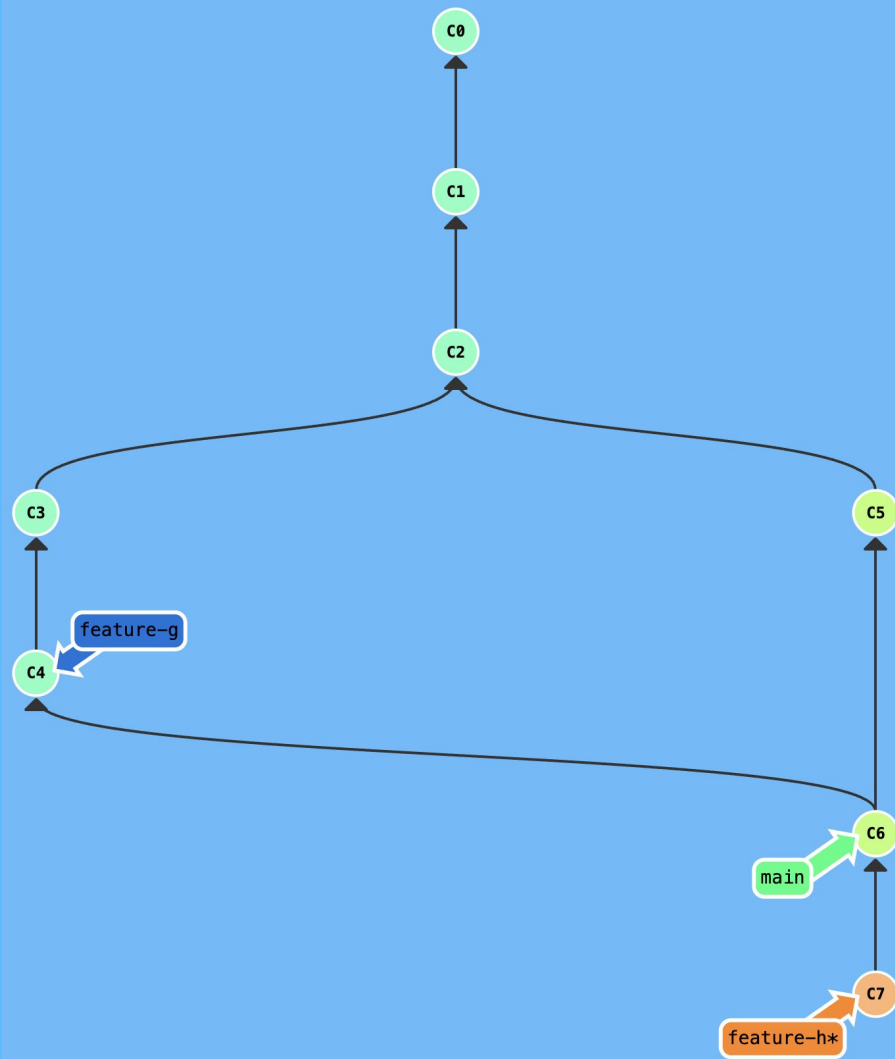
# Konsepter

- Commit
- Staging-området /index
- Branch
- HEAD
- Remote
- Kommandoer

# Commit

- Snapshot av hele repoet
- Hash / id
- Parent
- Melding
- Tilknyttet forfatter
- Tidsstempel





```
annebosch@t forelesninger % git show 0c96aa19575f1647c88258ec9f317e1d1
commit 0c96aa19575f1647c88258ec9f317e1d158aaa89 (HEAD -> main, origin/main, origin/HEAD)
Author: ambosch <anne.bosch@n
Date: Tue Feb 3 15:53:55 2026 +0100
```

update lecture plan

```
diff --git a/semesterplan.MD b/semesterplan.MD
index f5d5623..21b9c41 100644
--- a/semesterplan.MD
+++ b/semesterplan.MD
@@ -1,5 +1,5 @@
-Forelesninger
+
+### Forelesninger
+
+ - Tirsdag 14:15-16:00
```

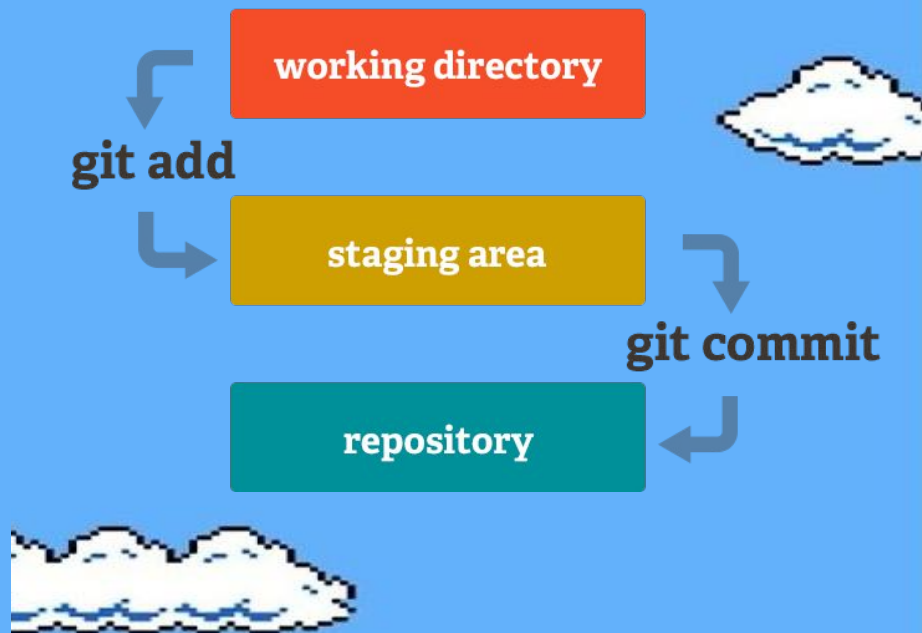


	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT  
MESSAGES GET LESS AND LESS INFORMATIVE.

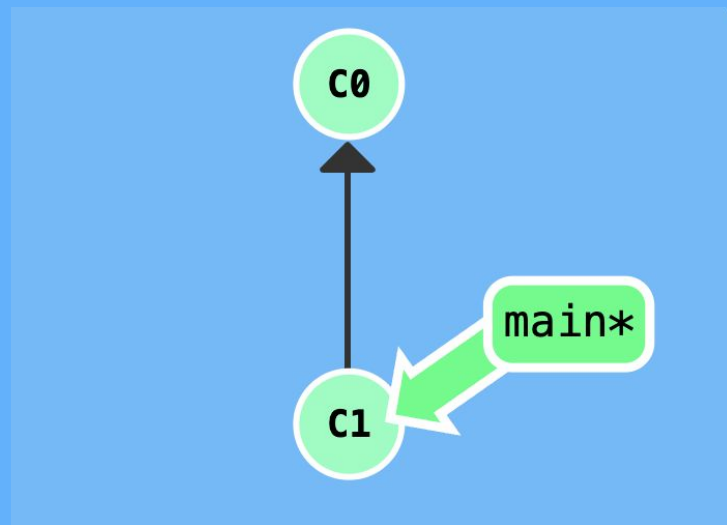
# Staging-området

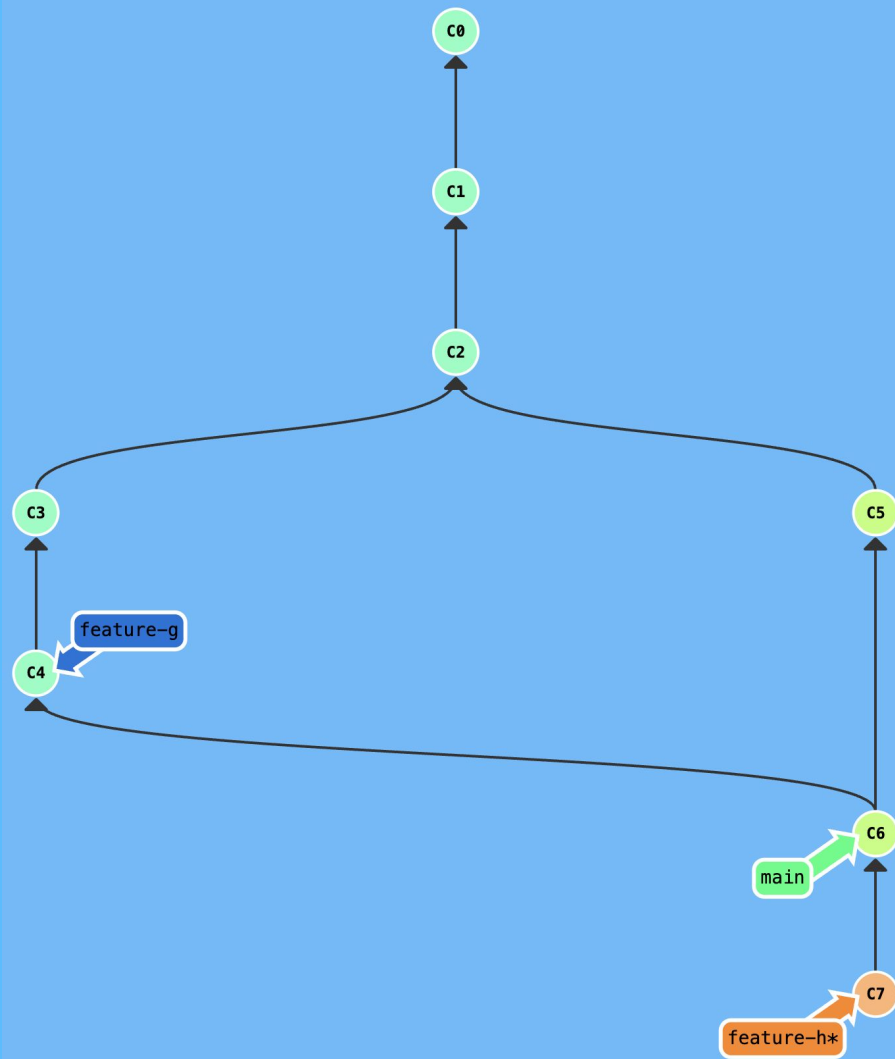
- Sette sammen en commit
- Ikke alle endringer må bli med
- Hele filer
- Enkelte linjer



# Branch

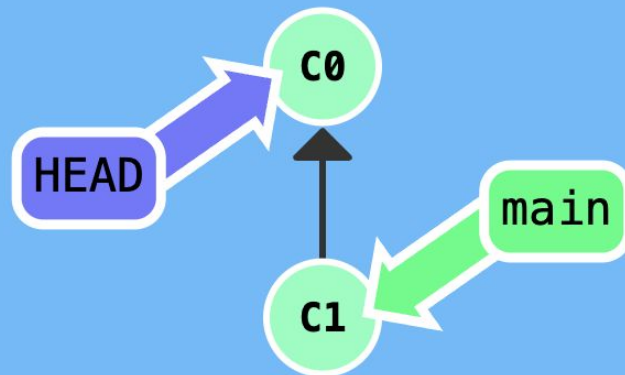
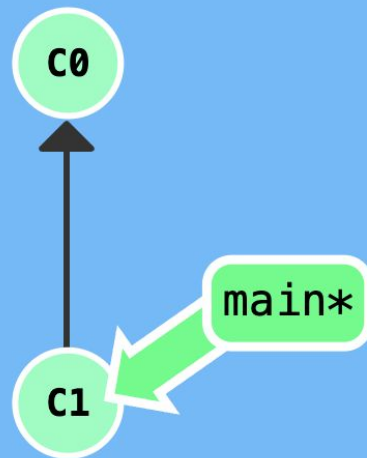
- Peker til en commit
- Pekeren har et navn
- Utforsking
- Systematisering
- Isolering
- Velge hva du ønsker å hente inn i main med en merge
- Teamarbeid





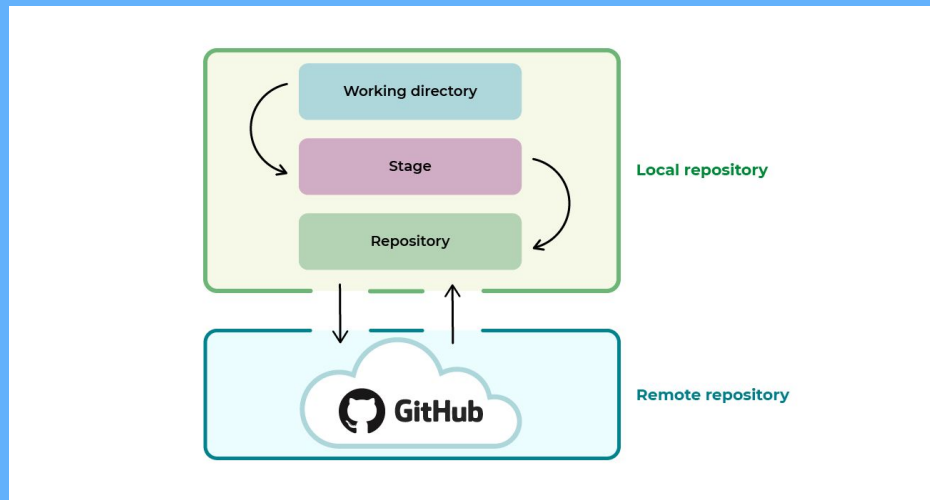
# HEAD

- Referanse til nåværende posisjon i historikken
- Hoppe rundt i historikken
- Detatched head



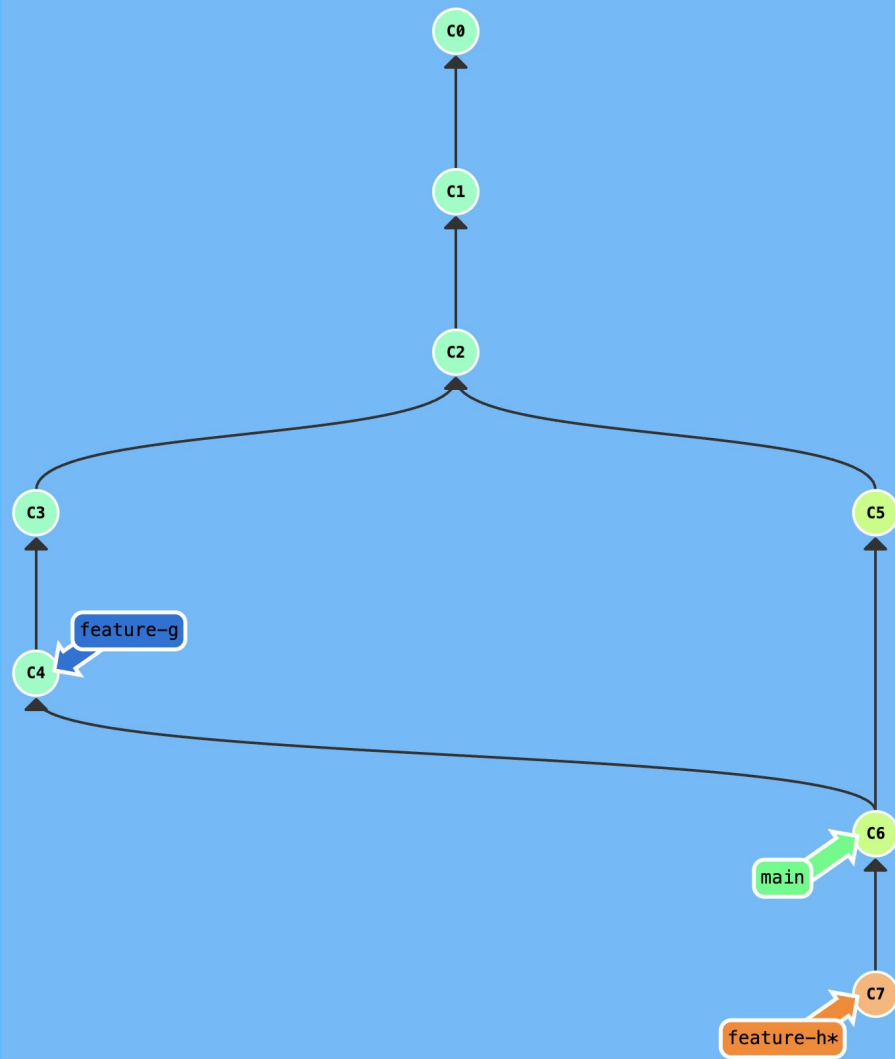
# Remote

- GitHub
- En kopi av repoet på en annen server (deployet)
- Backup
- Deling med andre
- Pull requests og code reviews
- De-facto sannheten
- Synkronisere lokalt og remote manuelt



# Kommandoer

- `git init`
- `git add <file.md>`
- `git commit -m "<my message>"`
- `git commit --amend`
- `git status`
- `git branch <name>`
- `git checkout <branchname>`
- `git merge <branchname>`



# Kommandoer

- `git init .`
- `git add <file.md>`
- `git commit -m "<my message>"`
- `git commit --amend`
- `git status`
- `git branch <name>`
- `git checkout <branchname>`
- `git merge <branchname>`
- `git clone <remote url>`
- `git fetch`
- `git pull`
- `git push`

# Aldri utlært, men kommer langt med kjernen

- tags
- stash
- rebase
- reflog
- reset
- cherry-pick
- octopus-merge
- forks
- ...

Bli komfortabel med git

→ investering som lønner seg

Oblig 3 - teamarbeid

# Drar sammen trådene til obliquen

- Bruke git i utviklingen
- Teste ut ulike løsninger
- Leverer koden på GitHub
- Pushe opp til remote
- 2 brancher
  - Før utvidelsen
  - Etter utvidelsen

# Gode praksiser

- Vi jobber sjelden direkte i main
- Merger PRer inn i main
- Committe ofte
- Små commits
- Gode committ-meldinger
- Pushe ofte




(DEMO)



# Oppgave til fredag



- Sette opp git lokalt - installere
  - GitHub-bruker med ut-e-posten
  - Godta invitasjonen inn i emnet sin GitHub-organisasjon (får e-post med link)
  - Godta oppgaven i GitHub classroom (egen link)
  - Da har du automagisk fått et repo på github
  - Oppgaveteksten
  - Kloner dette repoet lokalt
  - Autentisering mellom GitHub og lokal maskin
  - Pushe opp en liten endring - opprette main.py
  - Da er du klar til å knekke igang med oblig 2
  - Bonus: Søke om GitHub copilot pro-lisens for studenter
- 

Legge ut på Canvas:

- Smørbrødlista
- Guide til autentisering
- Lenker - øve på git-kommandoer

Arbeidslivsrelevans

Anvende  
objektorienterte  
prinsipper

GUI / pygame

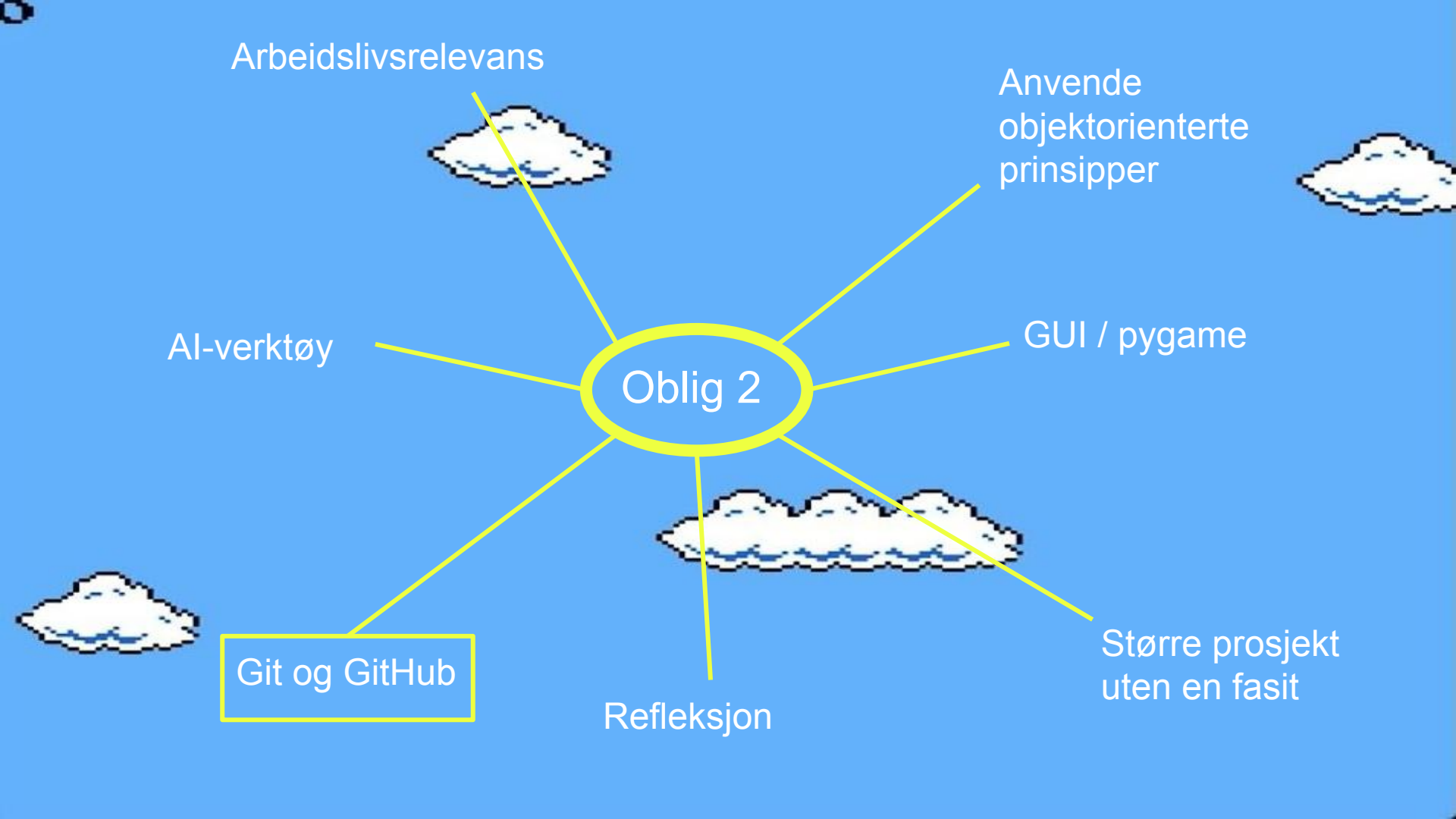
AI-verktøy

Oblig 2

Git og GitHub

Refleksjon

Større prosjekt  
uten en fasit



# Målet med denne forelesninga

- God innsikt i arbeidskravet
- Hva kreves av dere i arbeidskravet
- Git er ikke skummelt
  - Forståelig verktøy
  - Superkraft
- Hvordan jobbe mot en remote på GitHub
- Klar for å sette igang, første oppsett denne uka

# Tilbakemelding

<https://nettskjema.no/a/vibe-check>

