



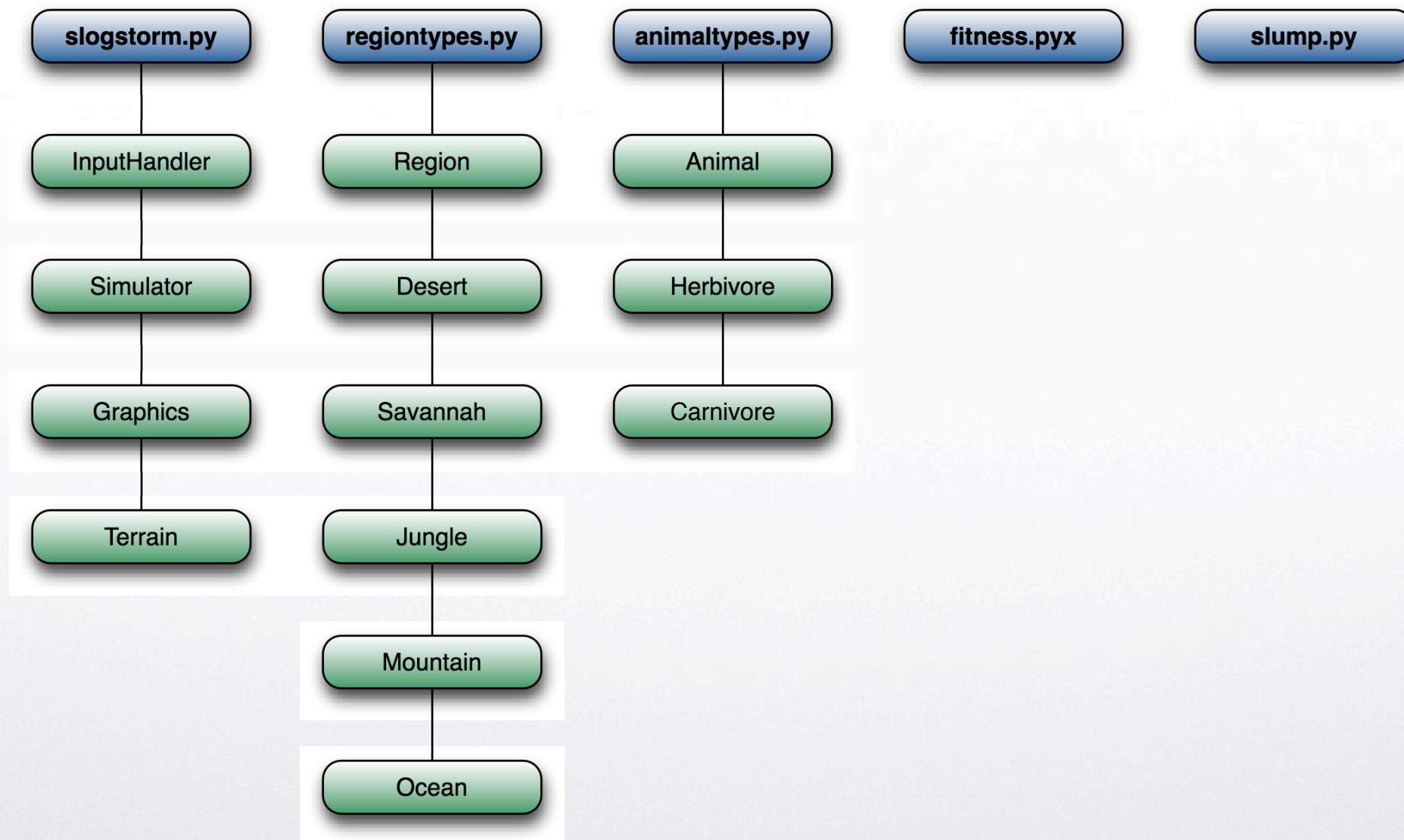
# Modellering av Bjarnøyas Økosystem

---

Team Slogstorm  
Aleksander Hykkerud og Daniel Hjertholm



# Struktur







# Filoppdeling

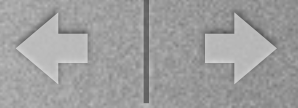
- 5 filer / moduler
- Enklere å få oversikt
- Slump: enkelt å bytte ut elementer



# Klasseoppdeling

- **InputHandler**
  - Har alle metoder brukeren trenger (generere terreng, starte simulering, sette ut dyr, forandre parametre, lage animasjon osv...)
  - Videreformidler til andre klasser
  -





# Klasseoppdeling

- Grafikk
  - Tegner grafikken
- Simulator
  - Kjører simulering i x antall år
  - Mottar grafikk- og terreng-objekter og arbeider med disse



# Terreng

- Her dannes terrengtyper (i array)
- Inneholder metoder som har mulighet til å gå igjennom arrayet
  - growth
  - migration
  - decay
  - animal\_counts
  - terrain\_map
  - terrain\_dimensions





# Terrengtyper

- Superklasse Region
- Underklasser Desert, Savannah, Jungle, Mountain, Ocean
- Inneholder dyrene i lister (to forskjellige)
- Inneholder metoder som behandler dyrelistene



# Dyretyper

- Superklasse Animal
- Underklasser Herbivore og Carnivore
- Metoder som endrer statusen til dyret
- Metoder som forteller om den dør eller flytter på seg.
- Underklassene inneholder spisefunksjoner (overloading)
- Parametere lagret på *klassen* vha. staticmethod





# Årets syklus

- Simulering går igjennom metoder i terreng som representerer hver årstid

```
for self._year in range(self._year + 1, self._year + 1 +
years):
    self._terrain.growth()
    self._terrain.migration(self._year)
    self._terrain.decay()
```



# Årets syklus

- Metoder i terreng går igjennom celle for celle å gjennomfører syklussene i hvert terreng

```
def growth(self):  
    """Perform regrowth, nutrition and breeding cycles."""  
  
    for row in self.terrain_map():  
        for celle in row:  
            celle.regrowth_cycle()  
            celle.nutrition_cycle()  
            celle.breeding_cycle()
```





# Eksempelsyklus

- Et år begynner, simuleringsklassen ber Terrain-objekt kjøre decay-metoden.
- Decay går igjennom celle for celle å ber det aktuelle terrenget i den cellen utføre aging\_cycle-metoden
- aging\_cycle-metoden går igjennom listen med dyr i terrenget ber dyreobjektene bruke metoden aging som øker alderen til dyret med 1 og oppdaterer fitness



# Eksempelsyklus

```
for self._year in range(self._year + 1, self._year + 1 + years):
    self._terrain.growth()
    self._terrain.migration(self._year)
    self._terrain.decay()

decay()
    for row in self.terrain_map():
        for celle in row:
            celle.aging_cycle()
            celle.weightloss_cycle()
            celle.death_cycle()

aging_cycle()
    for animal in self._herbivores + self._carnivores:
        animal.aging()

aging()
    self._age += 1
    self.update_fitness()
```





# Optimalisering

- Python er svært allsidig å kan ta imot veldig mye forskjellig input
- Python må da teste hva den får inn hele tiden, noe som koster i prosessortid
- Ved å deklarere hva inputten skal være og kjøre dette i C istedenfor klarte vi å redusere denne tiden betraktelig



# Optimalisering

- Fitness-funksjonen ble kalt ca. 5 millioner ganger for en simulering på 10 000 år
- Sto for rundt 60% av CPU-tiden
- Endret til fitness-variabel og metode for å oppdatere variabelen
- Reduserte kall til metoden med en faktor 5





# Optimalisering

- Metoden brukte fortsatt rundt 30% av CPU-tiden
- La metoden i egen modul og “cythoniserte” den
- Reduserte CPU-bruken med en faktor 3



# Optimalisering

- Bilder fra profiler før og etter...





# Tester

- Tester de fleste metoder i programmet om de gir forventet resultat

```
def test_bad_mapstr_raises_exception(self):  
    """Ensure that bad mapstrings are handled correctly."""  
    self.assertRaises(ValueError, slog.InputHandler, mapstr="000\n0JO\n0000")  
    self.assertRaises(ValueError, slog.InputHandler, mapstr="B00\n0JO\n000")  
    self.assertRaises(ValueError, slog.InputHandler, mapstr="o00\n0JO\n0000")  
    self.assertRaises(ValueError, slog.InputHandler, mapstr="D00\n0JO\n000")
```



# Dokumentasjon

- Dokumentasjon generert med Sphinx
- Ingen Sphinx-tilpassning i docstrings

[link til dokumentasjon...](#)

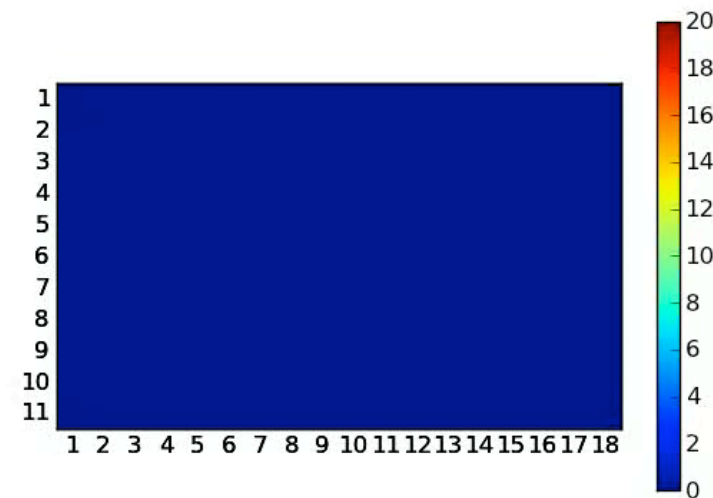
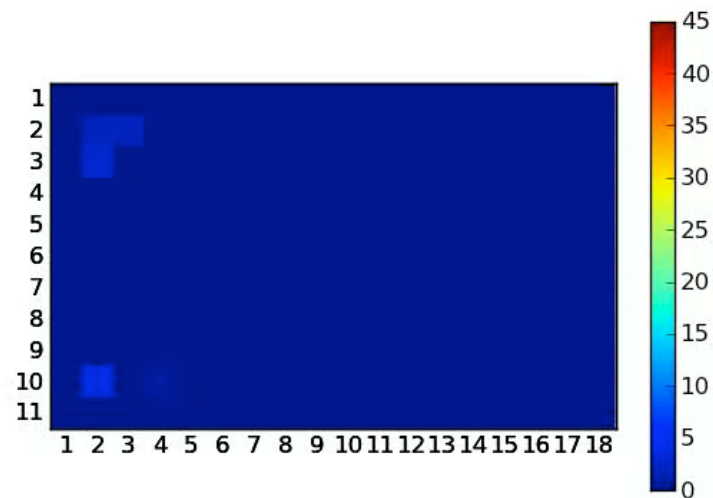
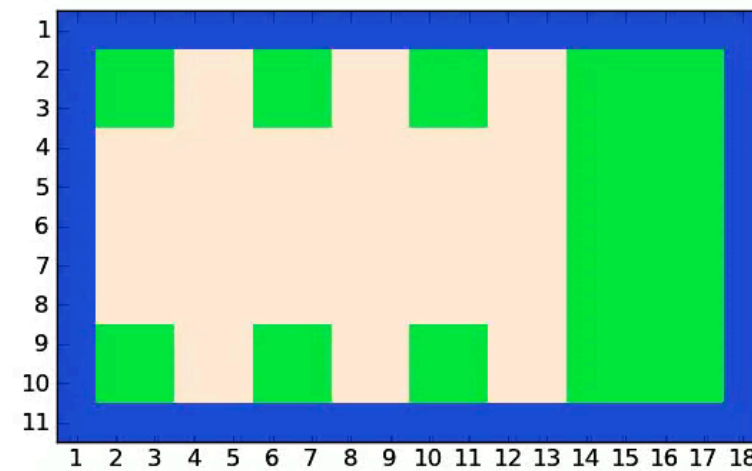
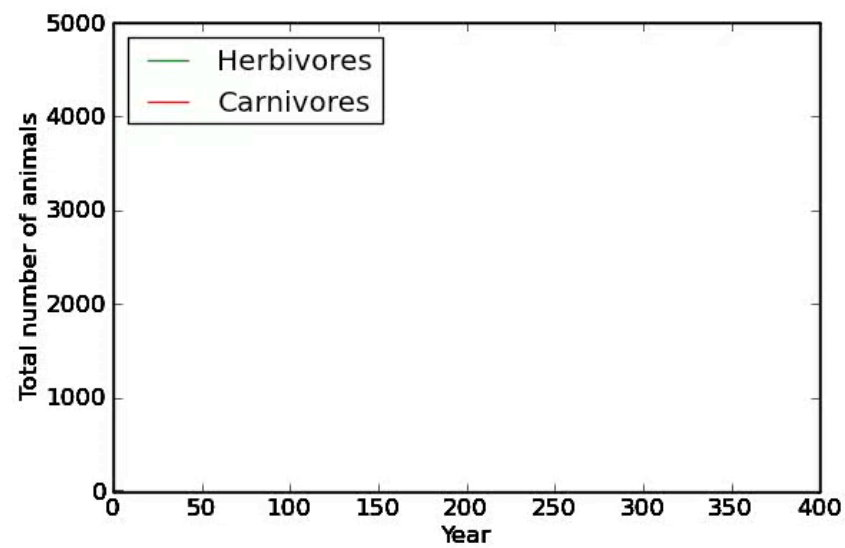




# Demo



# Demo







# Demo