Elk Island National Park

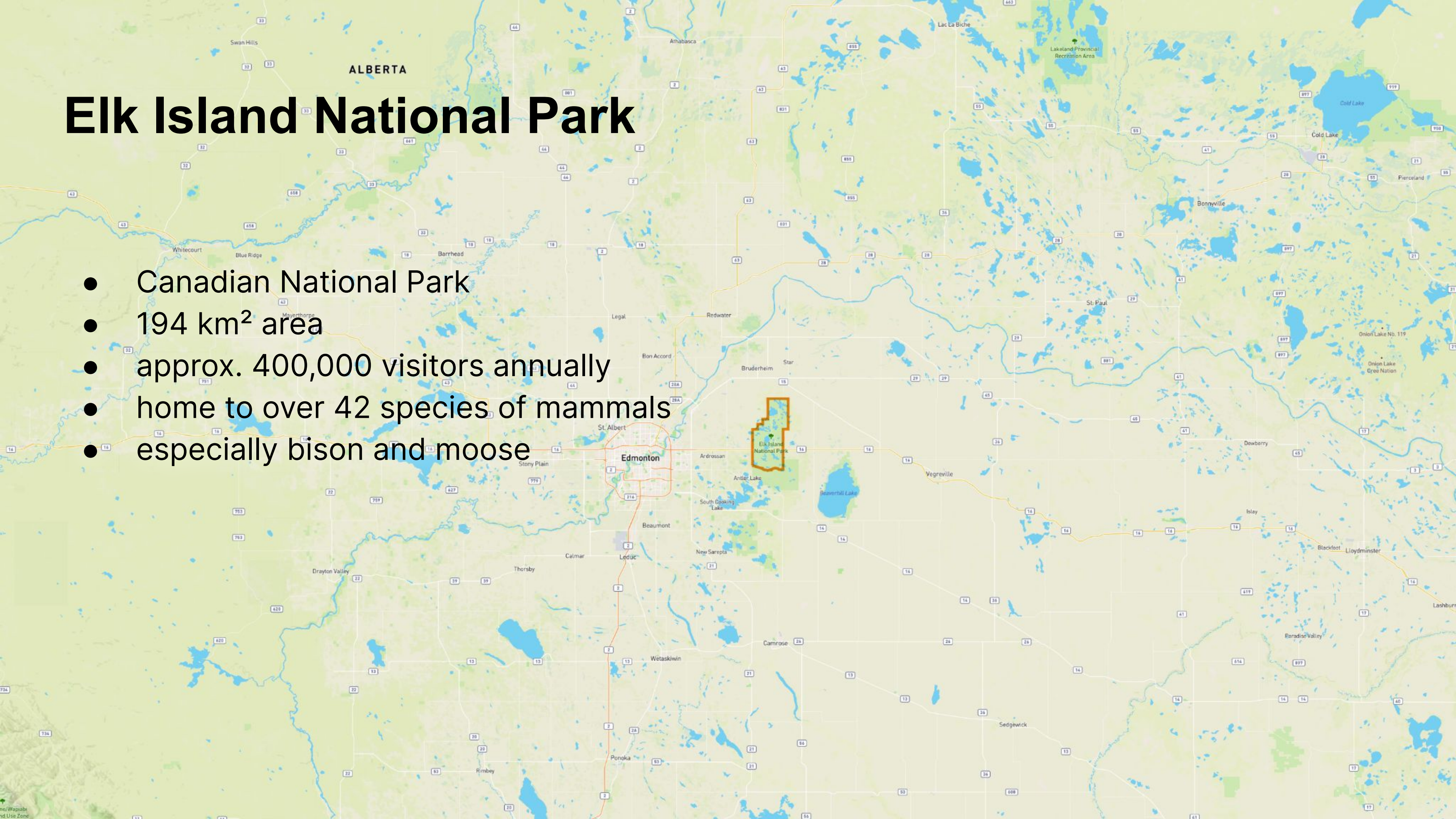
Antoni, Johnny, Olga, Ivoni, Sergio, Lorenzo

Structure

- Project Overview + Park Information
- Research and Data
- Layers
  - Water
  - Perimeter
  - Vegetation
  - Altitude
  - Temperature
- Agents behaviour
  - Spawn files
  - Abstract animal
  - Behaviour for specific animals
- Data Visualization
- Limitations and Problems
- Documentation Overview
- Kepler Demo

Elk Island National Park

- Canadian National Park
- 194 km² area
- approx. 400,000 visitors annually
- home to over 42 species of mammals
- especially bison and moose



Project Overview

- Goals
    - Simulate animal behaviour in the park to be able to better react to situations.
    - Determination of the ***carrying capacity***.
- Project Structure/Status
- Outcome/Deliverables
    - Model
    - Visualization (pictures)
    - Documentation

Animal data and research

- Special repo for animal specific data (such as daily, seasonal, reproductive behavior)
- Sources : https://red-sigma.github.io/einp-model/data_sources.html
- Problem: despite our efforts to collect as much data as possible for our research, we faced a challenge in that much of the data appeared too abstract, making it difficult to directly integrate it into the model.

Geo data search and extraction

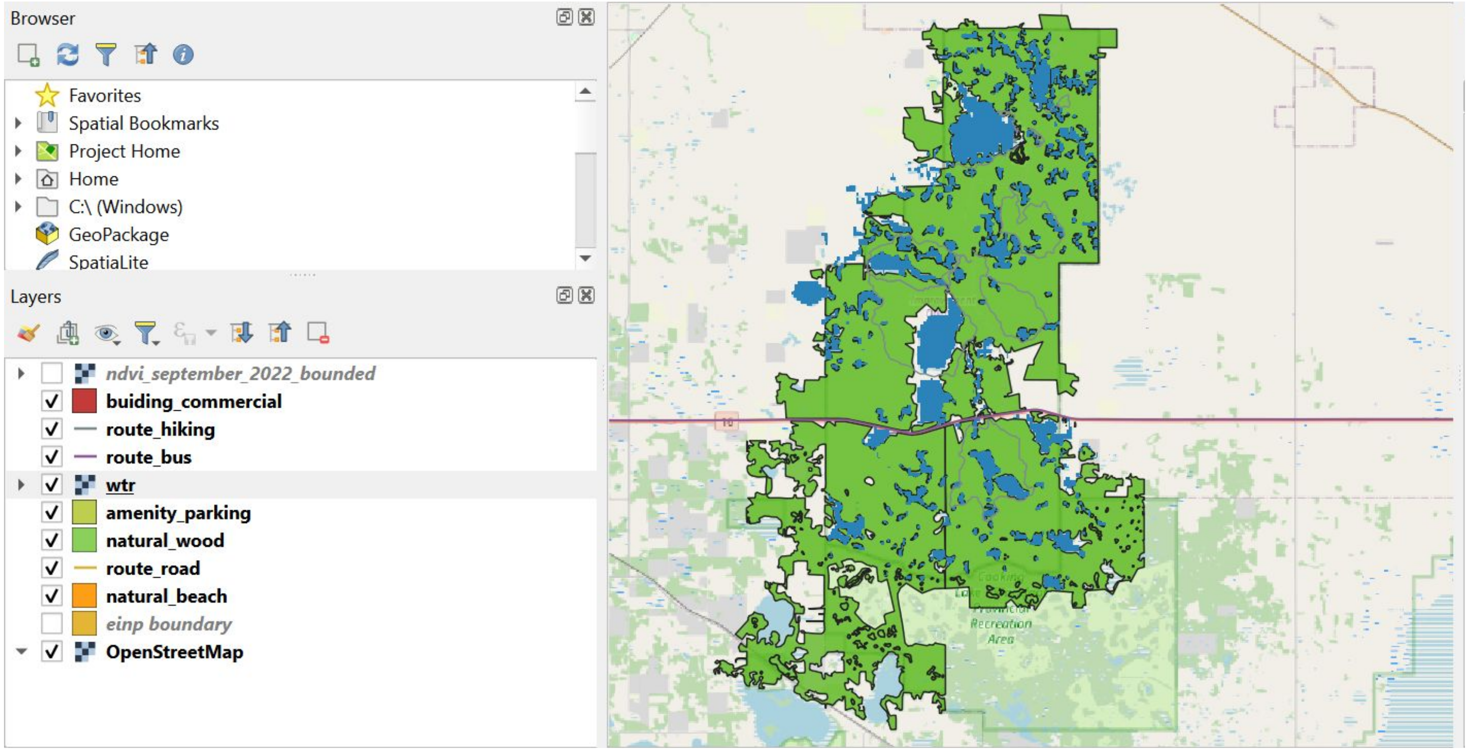
Data Sources and Tools:

- **OpenStreetMap**: Provides a comprehensive and freely accessible database of global geographic data
- **QGIS**: Open-source software for analyzing and visualizing spatial data

Geo data search and extraction

1. Preparing Data with OpenStreetMap
    a. Identifying and downloading relevant geographic area
    b. Filtering and selecting specific data for further processing
2. Data Processing with QGIS
    a. Upon importing data from OpenStreetMap, we utilized such tools as QuickOSM und JOSM Remote for extracting and querying features of the landscape
    b. Data transformation in formats that are supported by MARS framework
3. Data integration and Simulation
    a. Integrating extracted layers into simulation environment
    b. Establishing interaction within the simulation

QGIS

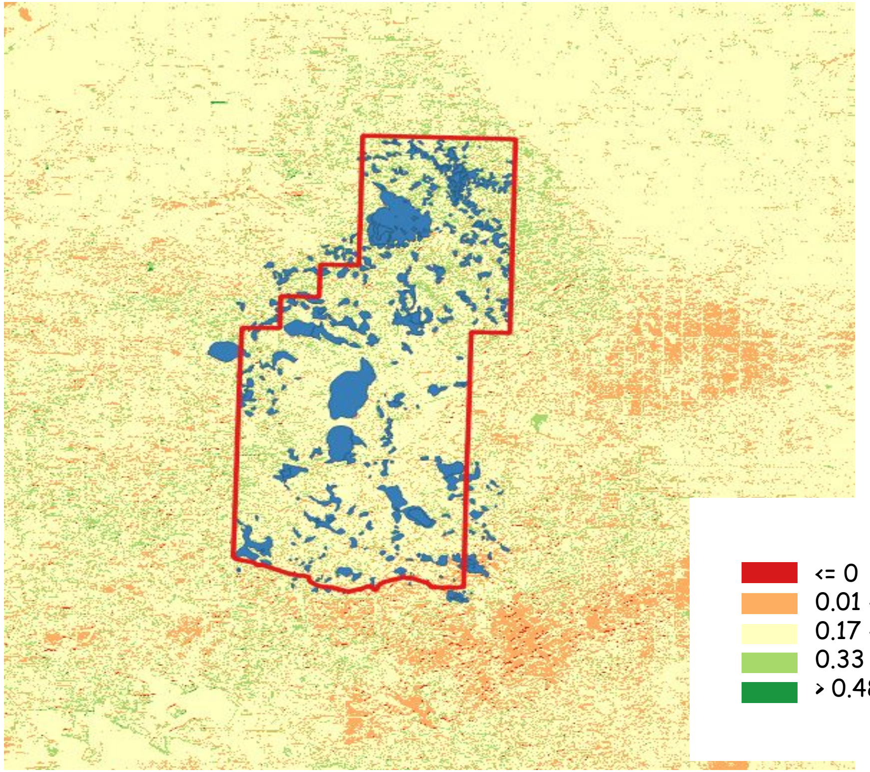

Browser
Favorites
Spatial Bookmarks
Project Home
Home
C:\ (Windows)
GeoPackage
SpatiaLite
Layers
ndvi_september_2022_bounded
buiding_commercial
route_hiking
route_bus
wtr
amenity_parking
natural_wood
route_road
natural_beach
einp boundary
OpenStreetMap
Recreation
Area

Normalized Difference Vegetation Index (NDVI)

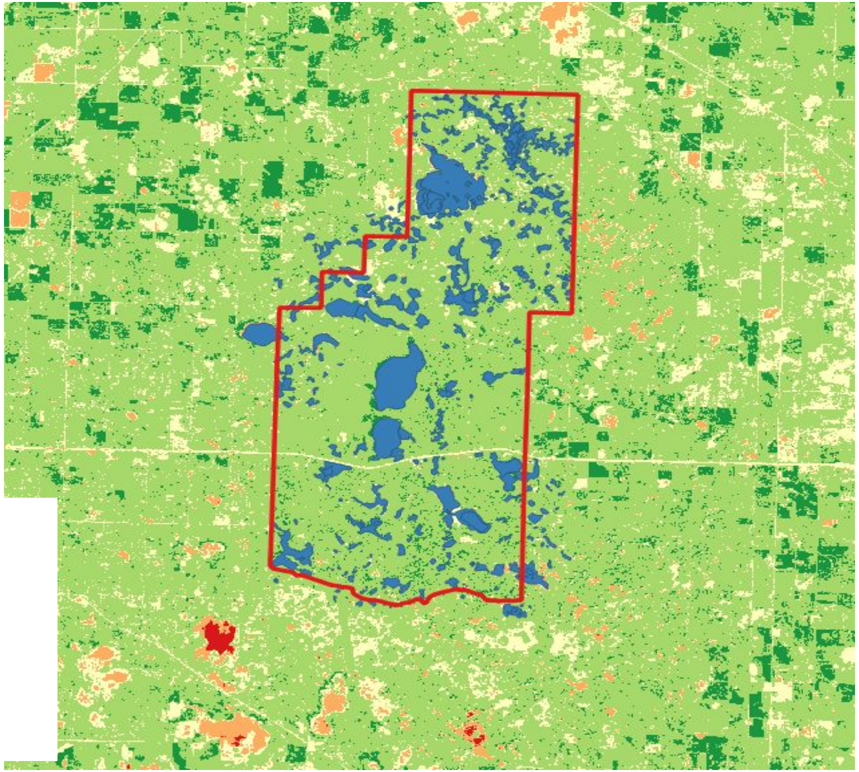
- **NDVI** is a used to assess the health and vigor of vegetation.
- **NDVI** formula: $$\frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}}$$
  - **NIR**: near-infrared reflectance (Landsat 8 Band 5)
  - **Red**: is the red reflectance (Landsat 8 Band 4)
- **NDVI** values range from -1 to 1, with higher values indicating healthier.
- Download Landsat 8 data from USGS Earth Explorer
- Calculate with QGIS

NDVI - Results

January 2023



July 2023



Vegetation Layer

- Uses raster data
- Explore NDVI values within range

ncols	478
nrows	569
xllcorner	-112.951338900000
yllcorner	53.510665400000
dx	0.000360805858
dy	0.000361231283

```
0.1087214350700378418 0.1280758678913116455l 0.12807
0.1598902642726898l934 0.1870967745780944824 2 0.1870
0.1825840473175048828 1 0.2099883705377578735 4 0.2099
0.1516868025064468383 8 0.1527098119258880615 2 0.1527
0.1593287885189056396 5 0.1468069255352020263 7 0.1468
```

```
var nearVegetationSpots = _vegetationLayer.Explore(Position, 20)
                .OrderByDescending(node => node.Node.Value)
                .ToList();
```

Water Layer

- Two different abstractions
  - RasterWaterLayer → On the stop checks
  - VectorWaterLayer → Get nearest water vertix

```
public class RasterWaterLayer : RasterLayer {

    public bool IsPointInside(Position position) {
        return Extent.Contains(position.X, position.Y) && GetValue(position) == 0;
    }

}
```

```
public class VectorWaterLayer : VectorLayer {
}
```

Water Layer/Perimeter

- RandomWalk()

```
protected void DoRandomWalk(int numOfAttempts) {
    while (numOfAttempts > 0) {
        var randomDistance = _random.Next(RandomWalkMinDistanceInM, RandomWalkMaxDistanceInM);
        var randomDirection = _random.Next(0, 360);

        Target = Position.GetRelativePosition(randomDirection, randomDistance);

        if (_perimeter.IsPointInside(Target) && !_gridWaterLayer.IsPointInside(Target)) {
            Position = Target;
            break;
        }
        numOfAttempts--;
    }

    Assert.IsTrue(_perimeter.IsPointInside(Position) && !_gridWaterLayer.IsPointInside(Position));
}
```

Water Layer/Perimeter

- LookForWaterAndDrink()

```
protected void LookForWaterAndDrink() {
    var nearestWaterSpot = _vectorWaterLayer
        .Nearest(new []{Position.X, Position.Y})
        .VectorStructured
        .Geometry
        .Coordinates
        .Where(coordinate => _perimeter.IsPointInside(new Position(coordinate.X, coordinate.Y)))
        .OrderBy(coordinate => Position.DistanceInMTo(coordinate.X, coordinate.Y))
        .ToList();

    foreach (var vertex in nearestWaterSpot) {
        Target =  new Position(vertex.X, vertex.Y);

        if (Position.DistanceInMTo(Target) < MaxWalkingDistanceInM
            && _perimeter.IsPointInside(Target)
            && !_gridWaterLayer.IsPointInside(Target)) {
            Position = Target;
            Hydration += 20;
            break;
        }
    }

    Assert.IsTrue(_perimeter.IsPointInside(Position) && !_gridWaterLayer.IsPointInside(Position));
}
```

Temperature Layer

- Data source: https://open-meteo.com/en/docs/historical-weather-api (CSV Format)
- Simple API: returns geo-localized temperature value in Celsius at a given time
- Based on MARS support for time series data (https://www.mars-group.org/docs/tutorial/data-sources/time-series) → folder of .asc files associated with a time
- Improvable with more precise data

```
public class TemperatureLayer: RasterLayer
{
    public double GetTemperature(Mars.Interfaces.Environments.Position position)
    {
        return GetValue(position);
    }
}
```

```
ncols         1
nrows         1
xllcorner     -112.88721
yllcorner     53.510665400000
dx            0.2
dy            0.3
NODATA_value  -9999

-25.5
```

Altitude Layer

- Uses raster data
- Explore altitude values within range

```
ncols         137
nrows         163
xllcorner     -112.950833333333
yllcorner     53.511250000000
dx            0.001253041363
dy            0.001252556237
```

```
740 739 739 736 737 740 740 740 737 739 739 73
735 731 731 734 733 732 732 735 738 736 736 73
734 734 734 734 731 736 736 736 739 736 736 73
730 729 729 733 733 728 728 735 734 737 737 74
735 725 725 728 733 735 735 733 734 737 737 73
729 733 733 737 727 724 724 724 731 736 736 73
728 731 731 729 734 724 724 722 730 736 736 73
```

```
var height = AltitudeLayer.GetValue(Target);
```

Abstract Animal

- Aspects taken into account for the modelling

```
public abstract Position Position { get; set; }
public abstract Position Target { get; set; }
public abstract double Hydration { get; set; }
public abstract double Satiety { get; set; }
public abstract double Latitude { get; set; }
public abstract double Longitude { get; set; }
public Guid ID { get; set; }
public double Bearing = 222.0;
public LandscapeLayer LandscapeLayer { get; set; }
public Perimeter Perimeter { get; set; }
public VectorWaterLayer VectorWaterLayer { get; set; }
public RasterWaterLayer RasterWaterLayer { get; set; }
public VegetationLayer VegetationLayer { get; set; }

public int _hoursLived;
public AnimalType _animalType;
public readonly int[] _reproductionYears = {2, 15};
public bool _pregnant;
public int _chanceOfDeath;
public int Age { get; set; }
public AnimalLifePeriod _LifePeriod;
public MattersOfDeath MatterOfDeath { get; private set; }
public bool IsAlive { get; set; } = true;

protected bool isLeading { get; }
protected int herdId { get; }

public static Random _random = new ();
private const int RandomWalkMaxDistanceInM = 500;
private const int RandomWalkMinDistanceInM = 10;
public const double MaxHydration = 100.0;
public const double MaxSatiety = 100.0;
public const int MaxAge = 25;
```

Animal Behaviour

- Tick()
  - Increment lifetime
  - Resolve states

```
public override void Tick() {

        _hoursLived++;

        if (_hoursLived % HoursPerMonth == 0 && _pregnant) {
            if (_pregnancyDuration < Bison_DurationOfPregnancy) {
                _pregnancyDuration++;
            }
            else {
                _pregnancyDuration = 0;
                _landscapeLayer.SpawnBison(LandscapeLayer, Perimeter, ...);
            }
...}
```

Animal Behaviour

- Tick()
  - YearlyRoutine()

```
public override void Tick() {
...
        if (_hoursLived == HoursPerYear)
        {
            YearlyRoutine();
        }
        if (!IsAlive) return;
...}
```

```
public override void YearlyRoutine() {

    _hoursLived = 0;
    Age++;

    var newLifePeriod = GetAnimalLifePeriodFromAge(Age);
    if (newLifePeriod != _LifePeriod) {
        if (newLifePeriod == AnimalLifePeriod.Adult) {
            if (_random.Next(2) == 0)
                _animalType = AnimalType.BisonBull;
            else
                _animalType = AnimalType.BisonCow;
        }
        _LifePeriod = newLifePeriod;
    }
...}
```

Animal Behaviour

- Tick()
  - YearlyRoutine()

```
public override void YearlyRoutine() {
...
    if (Age > 15)
    {
        _chanceOfDeath = (Age - 15) * 10;
        var rnd = _random.Next(0, 100);
        if (rnd >= _chanceOfDeath) return;
        Die(MattersOfDeath.Age);
        return;
    }
...
    if (_LifePeriod == AnimalLifePeriod.Adult && _random.Next(100) < ChanceForPregnancy) {
        _pregnant = true;
    }
...}
```

Animal Behaviour

- Tick()
  - Hydration

```
public override void Tick() {
...
        else if (Hydration < HydrationTreshold) {
            if (Hydration < MinHydration) {
                _hoursWithoutWater++;
                if (_hoursWithoutWater >= MaxDurationWithoutWater)
                {
                    Die(MattersOfDeath.NoWater);
                    return;
                }
            }
            LookForWaterAndDrink();
        }
...}
```

Spawn files

- Official data source: https://open.canada.ca/data/en/organization/pc?q=Elk+Island+National+Park&keywords=Alberta&portal_type=dataset&collection=primary&sort=
- Information needed: spawn location, life phase and gender of the animal, herd information
- Unfortunately information detailed to this degree is not available
- CSV Files for Bisons, Elks and Mooses automatically parsed by MARS framework
- Compatible with improved data in the future

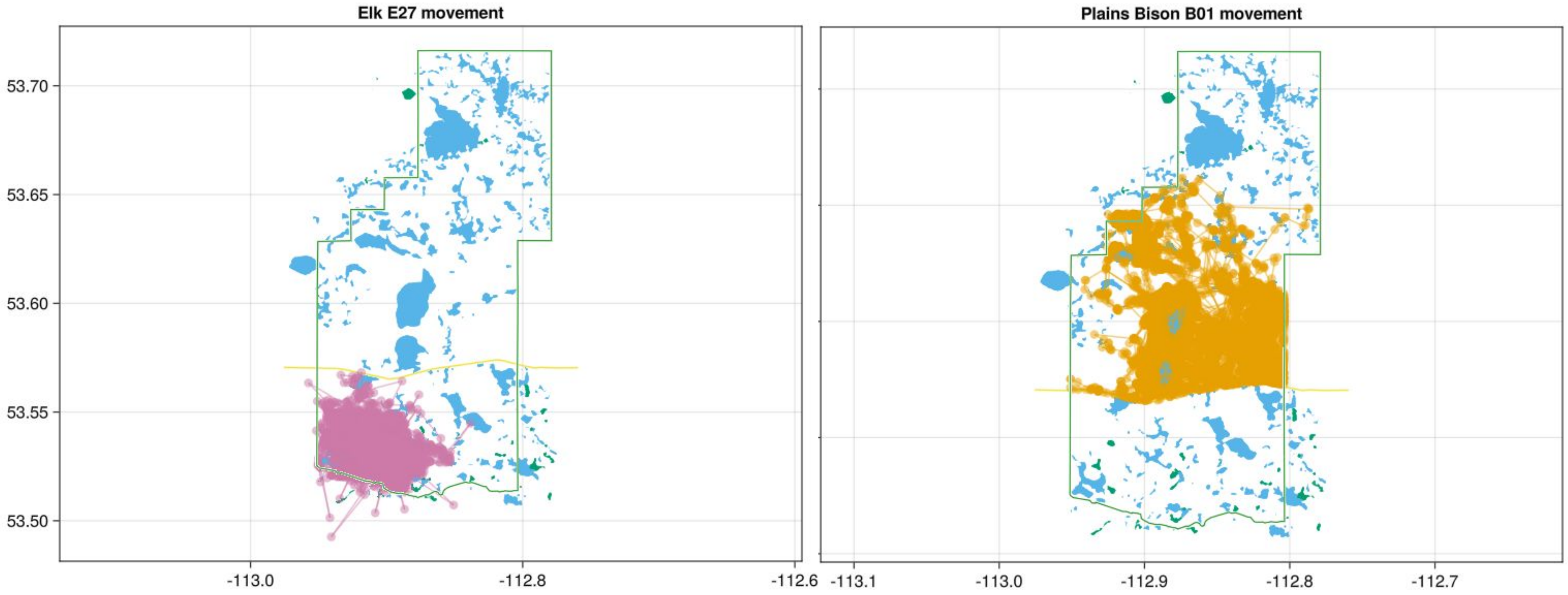
```
Latitude;Longitude;_animalType;BisonType;herdId;isLeading
53.666374258827545;-112.8249086593389;BisonCow;BISON_TYPE1;1;true
53.666374258827545;-112.8249086593389;BisonBull;BISON_TYPE1;1;false
53.666374258827545;-112.8249086593389;BisonBull;BISON_TYPE1;1;false
```

Animal parameters

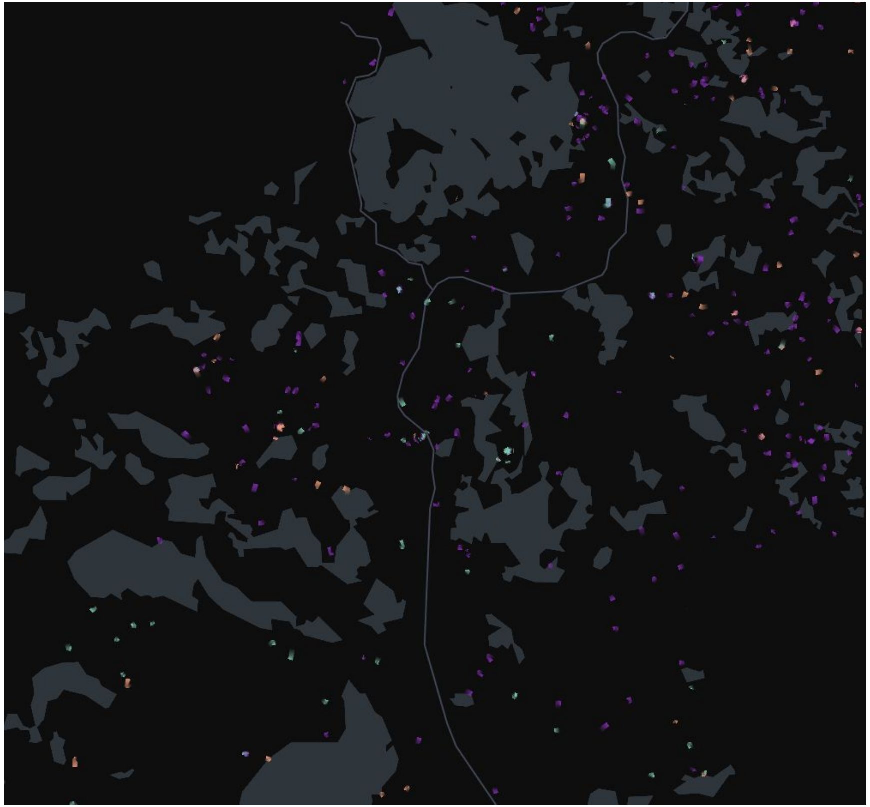
- Using the config.json file, individual parameters for animal behaviour can be defined
- Allows to tune quantity of food and water needed daily → determines changes in movement and behavior

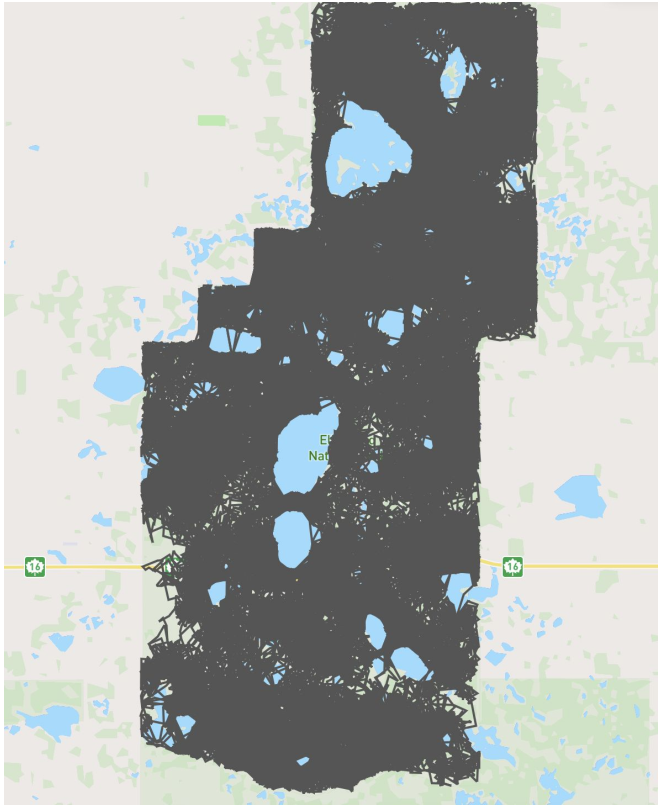
```
"name": "Elk",
"mapping": [
  {
    "parameter": "DailyFoodAdult",
    "value": 9.1
  },
  {
    "parameter": "DailyFoodAdolescent",
    "value": 4.2
  },
  {
    "parameter": "DailyFoodCalf",
    "value": 2.0
  },
  {
    "parameter": "DailyWaterAdult",
    "value": 60
  },
  {
    "parameter": "DailyWaterAdolescent",
    "value": 29
  },
  {
    "parameter": "DailyWaterCalf",
```

Data Visualization

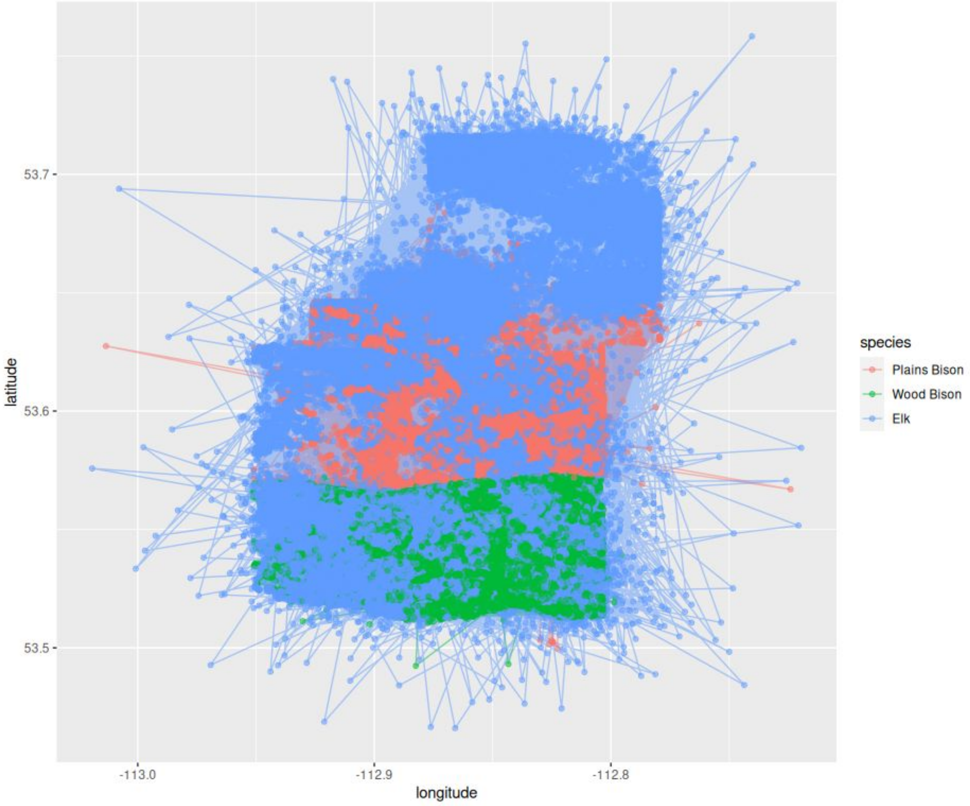


Data Visualization

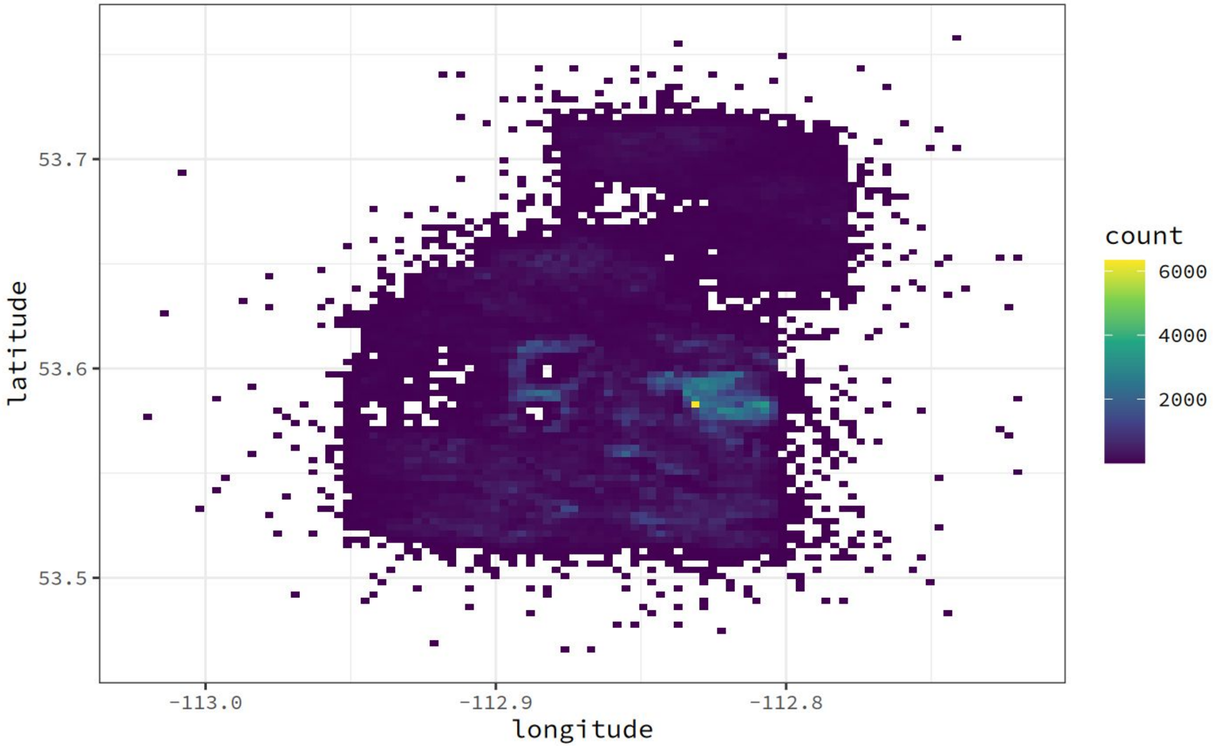




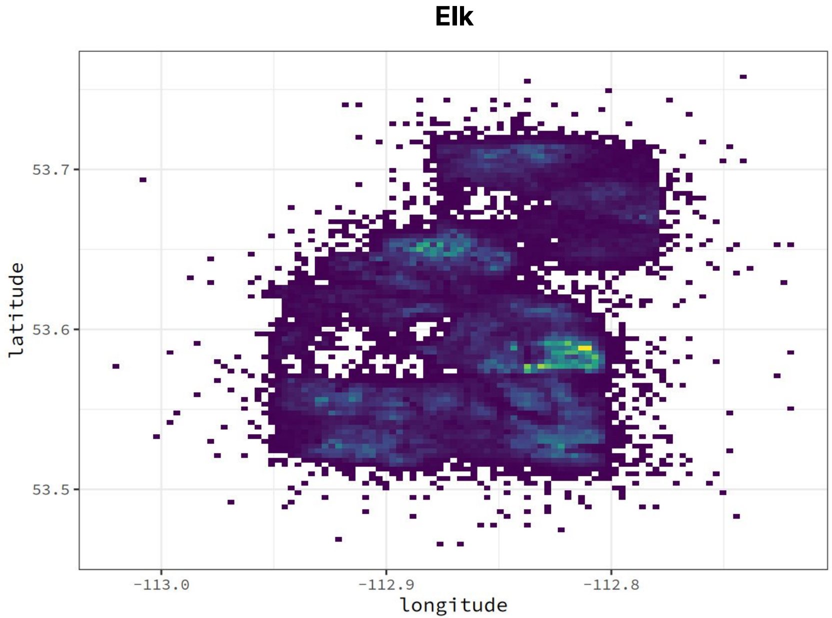
Data Visualization

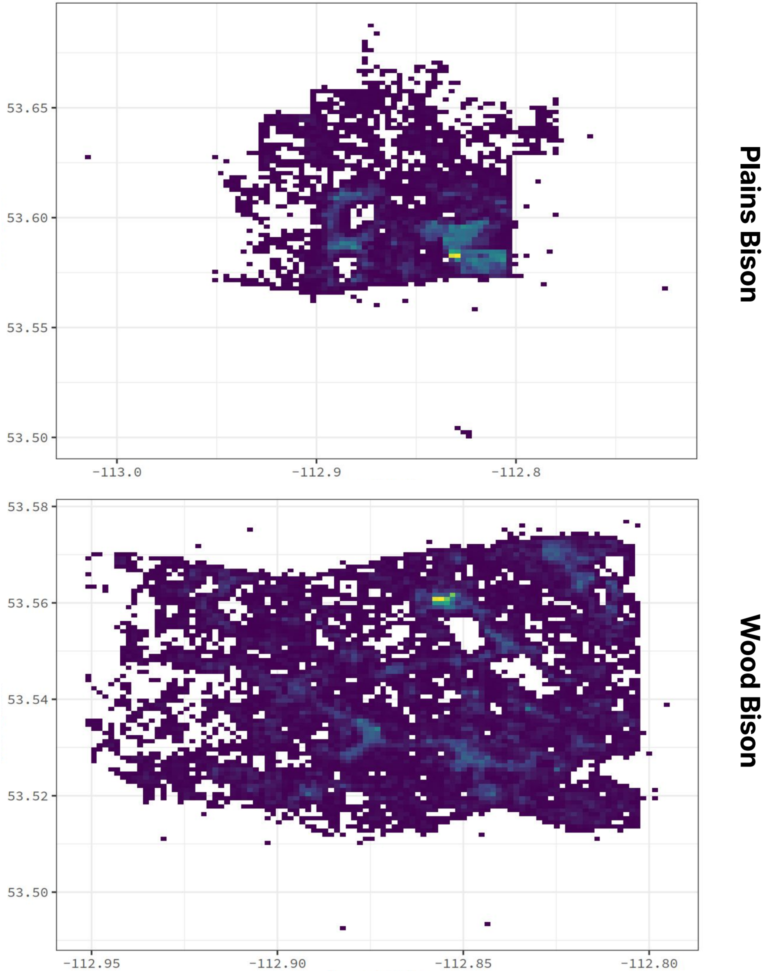


Data Visualization



Data Visualization





Future Development Roadmap

- Herd behaviour
- Altitude Layer logic
- Temperature Layer logic
- Season logic
- Damage Layer for Vegetation
- Initialization with existing herd

Documentation Overview

https://red-sigma.github.io/einp-model/

Introduction

The EINP Model is a simple MARS model that incorporates georeferenced raster and vector data. The model's environment represents the Elk Island National Park in Alberta, Canada. This area is modelled as a grid layer with georeferenced cells.

We simulate the behaviour of Bison, Moose and Elk inside the park, using agents.

Our goal is to provide an initial impression of how the animal populations evolves over time to make an estimate of to total carrying capacity of Elk Island National Park.

> Note: In its current state the model does not represent real animal behaviour accurately, and we hope that more features can be added in the future.

Quickstart

To use the model clone the repository by running:

```
git clone https://github.com/Red-Sigma/einp-model.git
```

Then go into the newly downloaded project and run the simulation:

```
cd einp-model/GeoRasterBlueprint/
dotnet run -sm config.json
```

If you want more details and options on how to run the simulation please have a look at Running & Output

Project Structure

After running `git clone` you should have a folder that looks similar to the one below.

```
einp-model/
├── docs/
├── GeoRasterBlueprint/
│   ├── Model/
│   ├── Resources/
│   ├── config.json
```

Demo

https://kepler.gl/demo