Africa Wildlife

Visualizing climate change-induced animal extinctions in Africa

Learning Unit 2, Methods in CS: Analysis
Daniele Solombrino, 1743111

Introduction

Africa Wildlife is an **interactive visualization** tool, realized in **Scala** via **Kojo** SDK by students of **5th year ITIS** school.

The goal of the interactive visualization is to **convey** a strong visual **message**, in order to make students aware of the massive animal extinctions caused by **climate change** in Africa and to promote **in-class discussions** about how scholars could help stopping climate changes.

Learning goals

1. Interdisciplinary

1.1. Make students, as future members of society, **conscious** about the huge threats posed to African fauna by **climate change**
1.2. Consolidate **science** knowledge
1.3. Consolidate **geography** knowledge

2. Computer Science

2.1. Observe how **computational thinking** can be applied to real-world problems
2.2. Consolidate **Scala** knowledge
2.3. Consolidate **Object Oriented Programming** (OOP)
2.4. Understand the importance of adopting the right **data structures** at the right time
2.5. Understand how to work with an **SDK** (**Kojo**)
2.6. Use of external **data** and **assets**
2.7. Usage and production of **documentation**

3. Soft skills

3.1. Creativity
3.2. **Group** work
3.3. **Presenting** work
3.4. **Time** management and optimization
3.5. **Gathering knowledge** from external sources

Learning goals interaction and challenges

Every proposed goal is tied to the other ones, creating multiple constructive challenges to the students.
Here's a description of such "challenging interconnection".

As future members of our society, it's important to inspire students to actively do something about climate change (learning goal 1.1), which is having a heavy toll on African fauna.

To do so, they first need to **understand** what are the components of the African **environment** (1.2), how they **interact** together and how they **affect** survivorship of African fauna (1.3).

Based on gathered knowledge, students can then start to **model** the African environment and fauna (2.1), translating their **creative** vision (3.1) into actual **Scala** code (2.2), using **OOP** and **Kojo SDK** (2.3 and 2.4).

Students at targeted class level have never worked with an SDK, which represents a **challenge**. They'll have to break the ice with SDK-related concepts and the usage of **documentation**, should they need some functionality that has not been covered in class for time reasons.

OOP and data structures always represent a challenge: every problem can be tackled using them in **multiple**, correct and incorrect **ways**.
Students will have to understand what are the best classes and data structures to use, in order to **comply** with the model they came up with.

Cartesian coordinates (1.4) will be useful to model **animal migrations**, which play an important role in the natural fauna extinction prevention and is one possible variable that can be considered in advanced versions of the project.

Pre-requisites

1. Interdisciplinary

1.1. Science of African environment
1.2. Africa geography
1.3. Mathematical functions
1.4. Cartesian coordinates

2. Computer Science

2.1. OOP
2.2. File management in Scala
2.3. Initial familiarity with Kojo SDK

Delivery to students

Pre-requisites listed above will be delivered in class by the appropriate teachers.

Science, Geography and Computer Science teachers will present the project (**co-held** lecture).

Students will receive:

- **Data** regarding African water sources and fauna, stored in Objects placed at the very beginning of the code files (example in appendix 1)
- Expected qualities that must transpire from developed code (e.g. **future extensibility**)

At the end of the presentations from all groups, the Computer Science teacher will present their version (co-held lecture with the English teacher), in order to conduct an **open discussion** about pros and cons of all delivered solutions.

Delivered from students

Students will have **3 weeks** of time to deliver:

- A list of group **participants**
- Full simulation implementation: front (**GUI**, example in appendix 2) and back-end (**code** governing the GUI)
- Accompanatory slides for the **oral presentation** (approx. 15 mins)

Evaluation

Whenever encountered, the word “conceptual” refers to the interdisciplinary topic(s).

Topic	**Points** (up to)	**Topic level**
Conceptual modeling of African environment	+1.5	Basic
Conceptual animal extinction modeling in function of up to 2 variables (for example, drank water and felt temperature)	+1	
Loading data and assets from disk	+1	
Hardcoding data rather than loading it from disk	-1	
Correct adoption of Kojo SDK	+1	
Correct OOP and data structures in “Basic” topics	+1	
Easily extensible code	+0.5	
Conceptual animal extinction modeling in function of 3 variables (for example, animal rivalries)	+1	Advanced
Correct OOP and data structures in “Advanced” topics	+0.5	
Good presentation	+0.5	
Conceptual animal extinction modeling in function of 4 variables (for example, animal migrations)	+1	Proficient
Added elements of dynamicity (count of current living animals, current year, opacity of water according to their level, etc.)	+0.5	
Correct OOP and data structures in “Proficient” topics	+0.5	

Bonuses (max 1 point):

- +1: good **documentation**
- +1: **creativity** and/or **accuracy** of variables considered in conceptual animal extinction function

Constraining on the maximum bonus points achievable aims at enforcing time management and prioritization (learning goal 3.4), very important soft skills in corporate and academia work environments, the two possible paths students can take after last year of high school.

Appendix

1)

```
object LionParams {
  val MAX_FELT_TEMPERATURE = 40;
  val MIN_WATER = 2
  val ICON_FILE_PATH = ICON_FOLDER_PATH + "lion_64.png"
}

object ElephantParams {
  val MAX_FELT_TEMPERATURE = 37;
  val MIN_WATER = 3.5
  val ICON_FILE_PATH = ICON_FOLDER_PATH + "elephant_64.png"
}

object ZebraParams {
  val MAX_FELT_TEMPERATURE = 36;
  val MIN_WATER = 2
  val ICON_FILE_PATH = ICON_FOLDER_PATH + "zebra_64.png"
}
```

2)