

Appendix 1. Kulayinjana, Overview-Design-Details (ODD)

The description of the model is based on the updated version (Grimm et al. 2010) of the ODD protocol that was originally proposed by Grimm and his colleagues in 2006. The model was uploaded on <https://www.openabm.org/model/5221>.

Purpose

The purpose of the game is to bring players to incarnate farmers in a virtual environment mimicking their reality, and to collectively learn about the use of the landscape through cattle herding.

Entities, state variables and scales

Cells are the elementary spatial entities of the model. The whole space is divided in two Zones, the first one representing Communal Land and the other representing Forest. Some forage is available in both zones.

In the model, a Household is either controlled by a human player (played household) or connected by the computer to a played household (computerized farm; it then behaves as a clone). In the communal land, Households manage Farms. A farm is made up of an aggregation of 6 cells: 5 Fields adjacent to 1 Kraal (enclosure for cattle). Played households own Cattle, an initial herd of 5 Cows that can be guarded by a Herdboy while grazing. Each Cow has a status (*thin, medium or fat*) that changes over time according to how the cow fed. On their Fields, played Households can grow 2 varieties of Maize (short-term and long-term). When harvested, crop leftovers, called Machanga, represent a source of food for the cows.

GrazingAreas (also called paddocks) represent the management units for cattle herding in the model. Grazing areas are aggregates of cells. There are 4 grazing areas in the Communal Land and 9 grazing areas in the Forest. The level of forage of each grazing area is *null, depleted, medium or good*. It changes according to the load of cattle, the season and the rainfall. Some grazing areas have Waterpans. When these water pans are not dried out, they are used by the cows located there.

Wildlife (lions and elephants) are likely to cause some disturbance to the cattle and the crops of the households.

An overview of the overall structure of the model is provided by a UML class diagram (see figure 1).

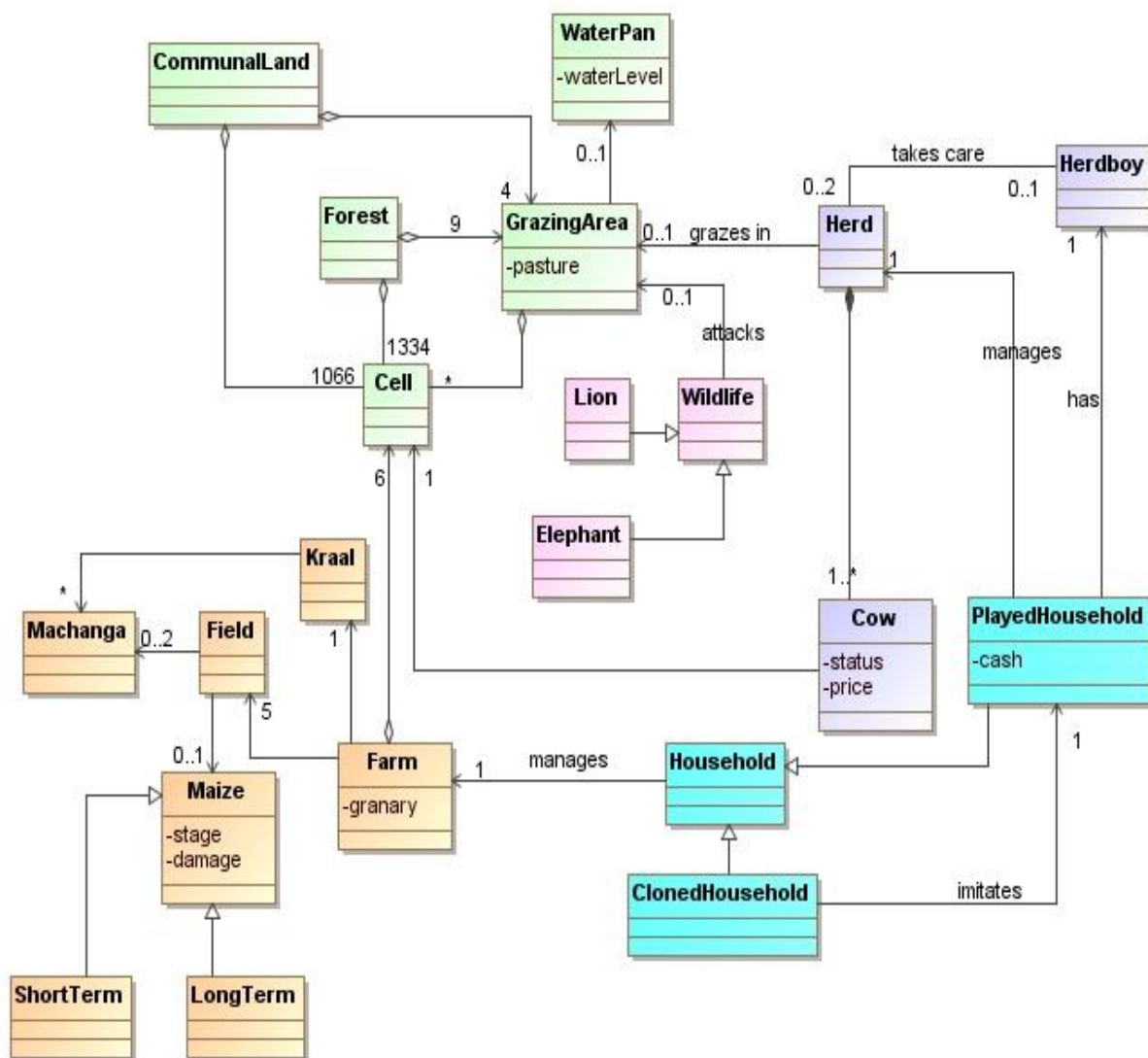


Figure 1. Class diagram of the *Kulayinjana* model.

Players have two objects of decision making at the beginning of each month: their cattle and their fields. Concerning cattle, they have to choose in which grazing area they will graze, and if they will be guarded or not, that is if they put a herdboy with the herd or not¹. At the beginning of the month, players can also decide to buy or sell cattle. Concerning their fields, at the beginning of each month players can decide to plow or harvest their fields. Fields can only be harvested if the maize is either mature or dry, but in the first case, harvesting implies building a granary for the maize to dry up.

¹ In the area, cows are usually grazing during the day, and gathered in the kraal at night we're the Kraal serves as a protection. In the model (and in the game), only active phases of cattle herding are considered, and nights are not simulated.

All of these actions obey specific cost/benefit rules. These rely on tokens that are distributed to, or given back by players. There are two types of tokens, small ones (ST) and big ones (BT). A big token equals six small ones. The balance of tokens is stored in households' *cashboxes*. The main parameters of the model are listed in table 1. The way these parameters affect the various processes is explained in the “details” subsection (last part of the ODD protocol).

| Entity | Parameter | Value | Unit |
|--------------------|-----------------------------------|--------------|-----------------|
| GrazingArea | cattleOverloadThreshold | 10 | cow |
| | protectionAtNightAgainstElephants | 3 | small token |
| Cow | fatteningThreshold | 6 | satiation index |
| | wastingThreshold | 3 | satiation index |
| | marketPrice_Fat | 18 | small token |
| | marketPrice_Medium | 12 | small token |
| | marketPrice_Thin | 6 | small token |
| Cattle | herdboyCost | 1 | small token |
| | grazingAreaCrossingCost | 1 | small token |
| | wateringCost | 1 | small token |
| | damageMaize_Cattle | 25% | expected yield |
| Wildlife | damageMaize_Elephant | 50% | expected yield |
| Maize | damageMaize_Climate | 100% | expected yield |
| | establishmentCost | 6 | small token |
| | yieldIncome | 12 | small token |
| | Machanga | 2 | small token |
| | machangaFeedingCapacity | 1 | satiation index |
| Household | initialCashbox | 48 | small token |
| Farm | granaryCost | 1 | small token |

Table 1. Model parameters.

The time step of the model² represents one day. The game covers two agricultural years, from the beginning of October (year₁) to the end of September (year₂). Players' decisions are done only at the beginning of the month. To complete a game session, 24 rounds of decisions have therefore to be achieved. The virtual environment (Fig.2) referred to as “the map” is a grid of 60*40 square cells. The size of a cell was defined by the design team so that one cell represents one “*umfallow*”, that is the surface that one farmer can plough in one day. Therefore, the cells' area is 0.5acre (45m*45m). The total surface covered by our virtual environment is 2400 cells, that is 1200 acres.

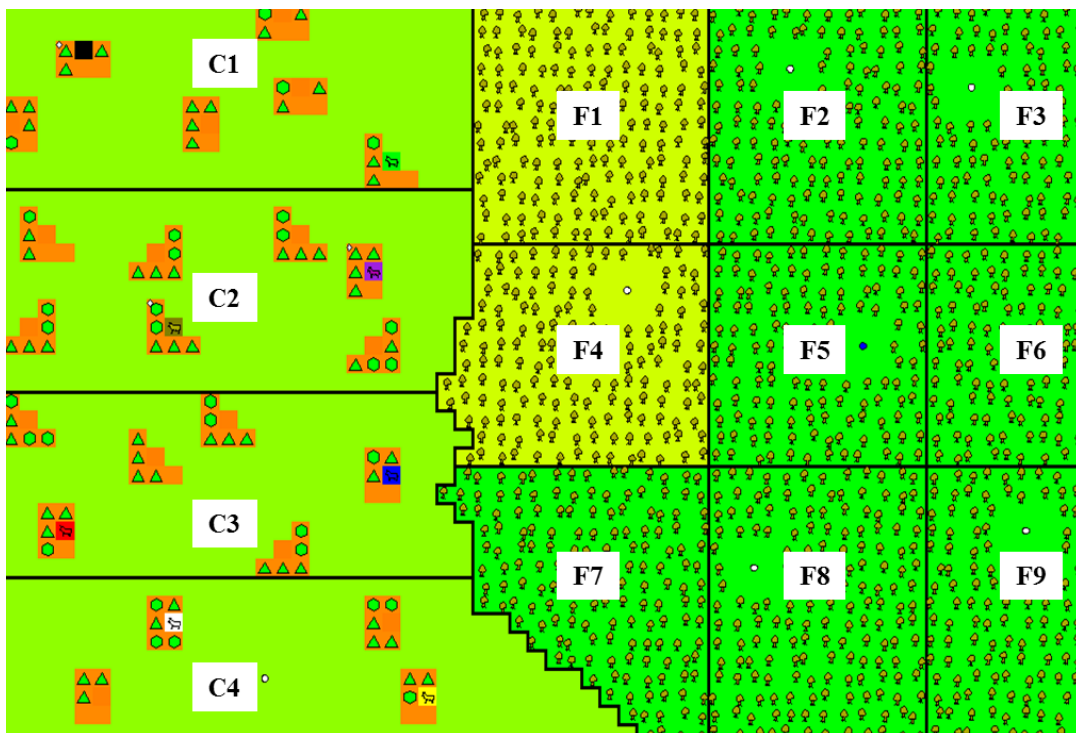


Figure 2. The virtual environment. The virtual environment is divided in 13 *GrazingAreas* numbered C1 to C4 in the communal land, F1 to F9 in the forest. Each played *Farm* has its *Kraal* colored according to the player in charge (2 played *Farms* by communal *GrazingArea*), the 5 fields appearing in orange (5 orange cells around the *Kraal*). *Farms* with no *Kraal* are clones managed by the model. The green entities seen on fields represent growing *Crops* (triangles are short term maize; rounds are long term maize). The figure shows three of the four different forage levels: “poor” (F1 and F4); “medium” (C1 to C4). and “good” (F2, F3, F5, F6, F7, F8 and F9). These levels change during the game according to players' actions.

Process overview and scheduling

² The playing time step is a month, that is that players make their decisions at the beginning of the month. The model supporting the game has a daily time step. In other words, the model does 30 time steps between each round.

The model presented here supports a role-playing game and as a result, is not run “continuously”, but is stopped and resumed for players to make their decisions (Fig.3).

At the beginning of the game, players choose from predefined locations which farm they want to manage. Then the simulation is scheduled by month.

At the beginning of the month, a weather forecast is given for the first week. Players notify with pawns if they want to plant Maize. Players put a pawn on each field they want to plow, knowing that each field can only have one growing Maize at a time. There are two types of pawn, corresponding to the two type of Maize. If some of the Maize is ready to be harvested, the players can choose to harvest. In that case, they have to notify if they leave the crop residue in the field (can be used by any Cattle), or if the crop residues are stored within the Kraal. To do so, they use pawns that they directly put on the game board. Players also decide in which GrazingArea they want their Cattle to graze daily for the next four weeks, and if they will be guarded by a Herdboy³.

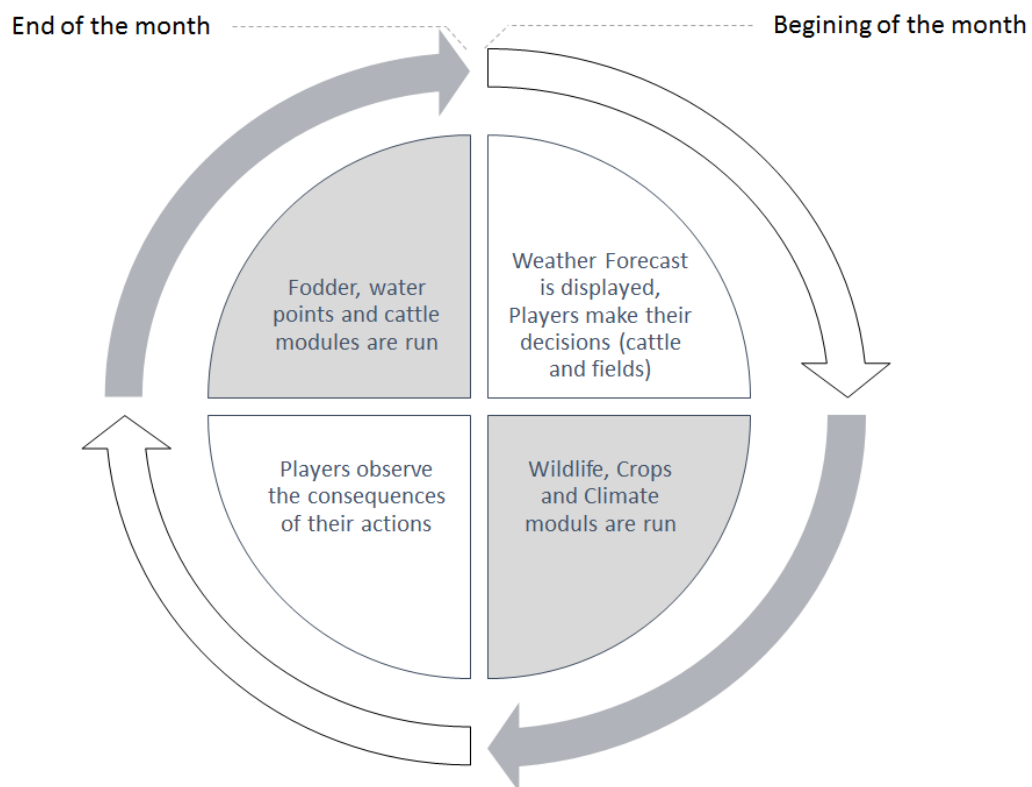


Figure 3. Sequential mobilization of the modules during the playing session. A round of playing, that is a month in the model, is done in four steps. Grey phases represent moments were the model is run, white phases are moments when the model is paused. The model is constituted of different modules, controlling specific dynamics (see next paragraphs). These modules are mobilized at specific steps of the month, as showed by the figure.

³ See foot note 71.

Once all the decisions are entered by the computer operator (Fig. 4), the model is run for 30 time steps (a month).

During the month, the climate is updated weekly. Players have access to a table displayed in the game's interface where the amount of rain is retrospectively shown (Fig. 5). Maize entities are updated daily (Fig. 8) by the computer (stages, failures).

At the end of the month, the Cattle damage module is run (cattle entering fields); the Wildlife module is run (lion attack and elephant crop raiding); Cattle statuses are updated according to the forage level of the GrazingArea they were using; the forage of every GrazingArea is updated. The number of cattle owned by players is updated along with cattle conditions. So are the sizes of Waterpans. Finally, the cashboxes of the 8 played households are updated.

The screenshot shows a software interface for player decisions. It includes a status message, a section for elephant protection, a table for cattle transactions, and a detailed table for farming and cattle management.

Cattle transactions table:

| Seller | Buyer | Purpose | Quantity | Amount |
|--------|--------|---------|----------|--------|
| B | Market | #medium | 1 | 12 |

Farming and Cattle table:

| | Granary | Fields | | | | | Cattle | | | |
|------------|--------------------------|--------|----|----|----|----|--------|----------|----------|------|
| | | F1 | F2 | F3 | F4 | F5 | Status | Location | Herd boy | Cost |
| Black [B] | <input type="checkbox"/> | ST | ST | | | | 5m | C2 | G | 2 |
| Green [G] | <input type="checkbox"/> | LT | | | | | 5m | C2 | | 2 |
| Olive [O] | <input type="checkbox"/> | LT | | | | | 5m | C3 | P | 2 |
| Purple [P] | <input type="checkbox"/> | ST | ST | ST | | | 5m | C1 | | 2 |
| Red [R] | <input type="checkbox"/> | ST | | | | | 5m | C2 | | 2 |
| Blue [U] | <input type="checkbox"/> | | | | | | 5m | C4 | | 2 |
| White [W] | <input type="checkbox"/> | | | | | | 5m | F1 | | 4 |
| Yellow [Y] | <input type="checkbox"/> | | | | | | 5m | F1 | | 4 |

Figure 5.4. Interface used by the computer operator to enter players' decisions at the beginning of each month. The "protection against elephants" can be filled with the initials of the players paying to protect a communal GrazingArea (in this case no one protected any). The "cattle transaction" box deals with cattle sales and purchases (here the Black player sold a medium cow to the market and earned 12ST). The "Farming and Cattle" relates to crops and cattle herding. Here for instance, the black players decided to plant two of his fields with short-term maize and to put his Cattle -guarded by the Green herdboy- in C2.

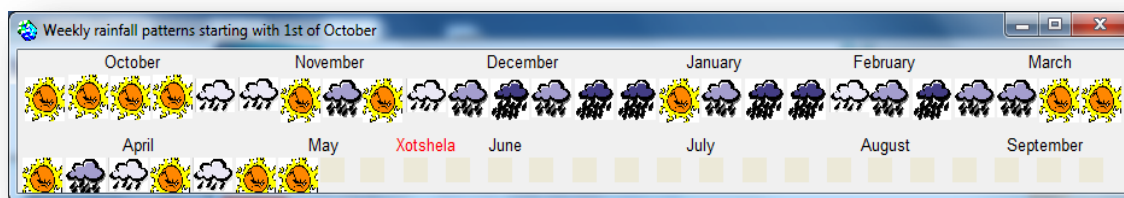


Figure 5. The weekly rainfall calendar. *The sun corresponds to a dry week, a light cloud represents 5-20mm of rain, a grey cloud represents 20-40mm of rain, the dark cloud represents +40mm of rainfall. The first of June is indicated as xotshela, a Ndebele word meaning “to push”, that corresponds to the historical traditional date when cattle are released freely in the fields to eat the crops leftovers.*

Design Concepts

The game was co-designed by researchers and representatives of the local community studied, with the objective of proposing a role-playing game mimicking local for players to reproduce their actual practices. The agents’ behavior is not programmed, but is left open for players to make choices. The consequences of players’ decisions are public. The players can therefore learn and adapt.

We assume that players make, to a certain extent, their decision following the same rationale they would use in the real life. The extent to which they reproduce their actual practices is assessed through a post-playing questionnaire.

Uncertainty is part of the model supporting the game. Wildlife actions for instance are based on probabilities, and players must consider risks while managing their cattle and fields. Similarly, when they have to make decisions, players do not know in advance the rainfalls of the coming month, except for the first week that is announced with a “weather forecast”. Assessing the way players deal with such uncertainties is one of the objectives of the game.

Being implemented as a role-playing game, interactions among agents are central. Players are free to talk during the gaming session, including talking together to advise or seek advice, congratulating or mocking, coordinating or working together. In particular, they can make

agreements to share the cost of guarding their cattle and the cost of guarding their communal paddock at night to prevent crop raiding by elephants.

The played households have similar characteristics at the beginning of the game. The players manipulate or own the same entities (5 fields, 5 cows) and have the same initial number of tokens. Nevertheless, we assume that a diversity of strategies and objectives will be exhibited during the gaming sessions, reflecting heterogeneity in the decision-making processes and objectives among the participants. Eliciting player's strategies and objectives and relating them to their strategies and objectives in their "real-life" is the heart of the post-game debriefing.

Some stochasticity is found in the Wildlife and the Cattle damage modules. The same predetermined sets of "random" events are used for each gaming session. standardizing the randomness is needed to ensure the comparability of the playing sessions.

The observation of the gaming sessions is supported by the use of the computer. Every playing decision is recorded, along with environmental parameters. Furthermore, supplementary information can be extracted from playing sessions through the replay function of the simulation platform used (CORMAS). Additionally, a member of the facilitation team records social interactions by taking pictures and collecting minutes later on organized in snippets of conversation relevant to various themes to be discussed during the post-game debriefing. After playing, questionnaires are administered to all participants.

Implementation details

The model was developed through several participatory workshops. The computer part was implemented with the Cormas simulation platform. The game was played 4 times and the playing sessions involved a total of 22 players.

Initialization of the simulation

The initialization of the simulation was always the same. The model is initiated at the beginning of October. All the farms, played (8) and cloned (15), are located by default. There are two played farms in each communal GrazingArea (Fig.6). Thanks to a name-drawing system, each participant in the playing session is asked to choose which Farm he wants to manage. Each players receives an initial cashbox of 48 small tokens (ST) that he will use to play (plant, drive his cattle, etc). Finally, each player starts with a Cattle herd of five medium

Cows. At the start, communal and forest GrazingAreas all have “medium” forage. All the herds are in their respective Kraals, and all Waterpans are empty, except the one located on F5.

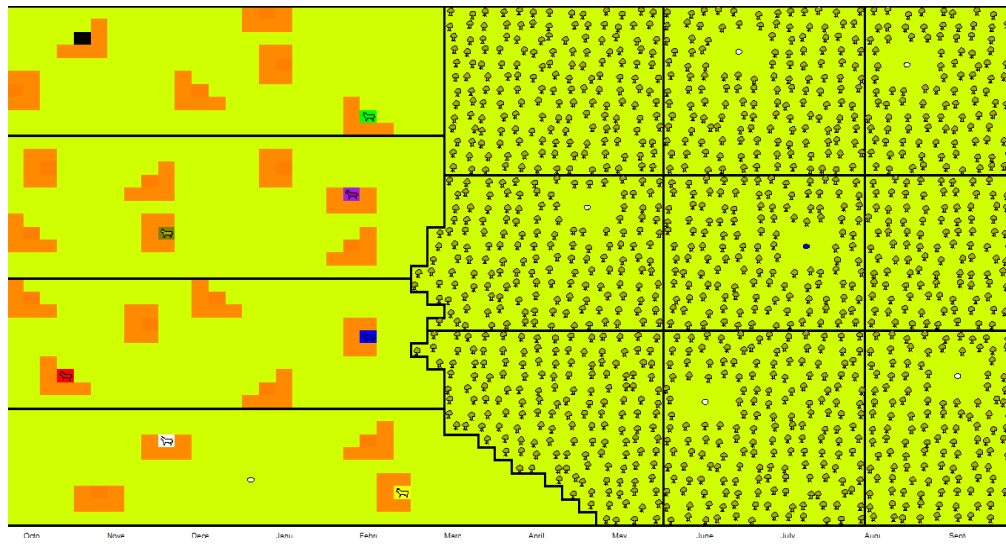


Figure 6. The virtual environment at the initiation of the model/playing session.

Input data

The model used weekly rainfall input data. Rainfall records were obtained in the study area. Two contrasted climatic years (Fig. 7) were used to produce a continuous 2-year dataset: a first “good year” (2012-13), measured by ourselves in the study area, is followed by a “bad year” (data from 1920-21), measured in the study area by the Rhodesian meteorological services. As showed in figure 5, the two sets of empirical data used for the rain sub-model propose very contrasted climatic conditions. The “good rains” year is characterized by abundant rainfall throughout the rainy season, with a total of 733mm, while the “bad rains” year only offers 531mm through erratic and low rainfall. This weekly rainfall was transformed into four types of week (Fig.5.5): “dry” (<5mm); small rain (<20mm); “medium rains” (20 to 40mm); “big rains” (>40mm).

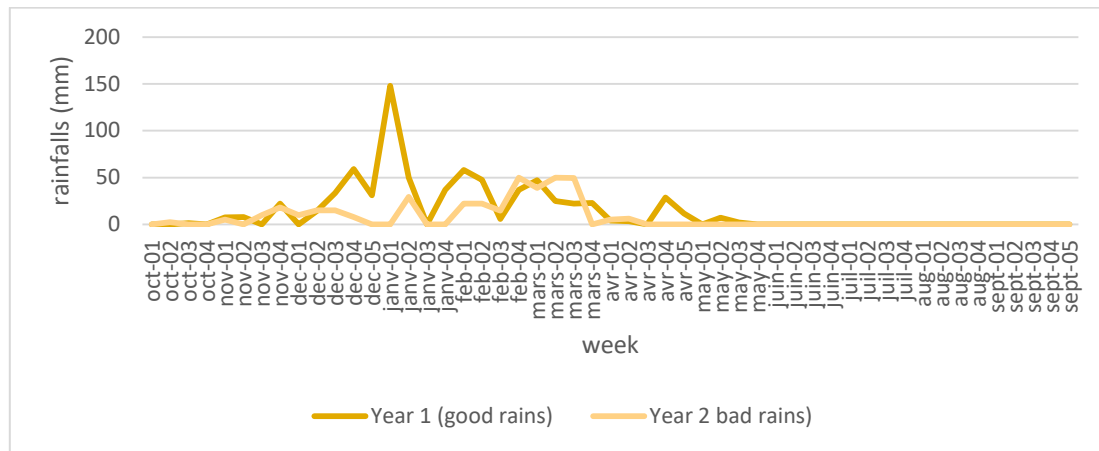


Figure 7. Weekly rainfall data. *The figure shows the two sets of data used for the rain module.*

Details for crop sub-model

This module controls crops dynamics and weather-related failures. There are two types of Crop. They have the same productivity, but differ by their growing dynamics. Once planted (stage: *seed*), both short-term and long-term types germinate after the first non-dry week (stage: *germinated*) and remain in that stage –prone to rain washing– for two months, until they evolve to the “shooting” status. Crop types differ by the time needed to change from the stage *shooting* to the stage *mature*: 1 month for the short-term variety against 2 months for the long-term variety. The transition diagrams are represented in figure 8. Players can harvest their crops when they are either “mature” or “dry”. Nevertheless, when players harvest “mature” crops they must build a granary, unless they already have one.

Crops are sensitive to drought. Between the moment they are planted and the end of December, three consecutive weeks without rain cause a loss of 100% of every growing short-term crop. Long-term crops are more resistant: they need four weeks of drought to be destroyed. On the other hand, during the *germinated* stage of development, crops are also sensitive to floods, and both types will be destroyed (100%) by three consecutive weeks of heavy rains.

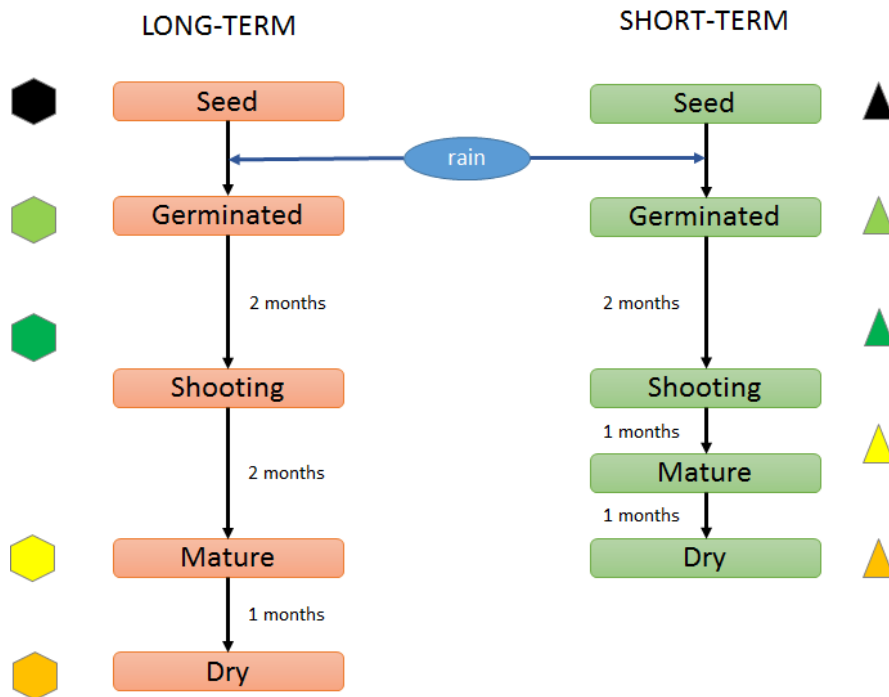


Figure 8. Crops transition diagrams.

Details for elephant damages

The module related to damages made by wild herbivores (only elephants are represented in the model) to crops is run at the level of grazing areas. Each month, grazing areas that are not protected by players and containing at least one field with a mature maize have 94/100 chances of being attacked by elephants. When elephants attack, all the farms are not impacted the same way. The first line of farms (Fig.9) will have up to three fields damaged, whereas the other farms of the grazing area will only have one field attacked. An attack will cause a loss of 50% of the yields. By protecting their grazing area, players decrease the chances of having elephants attacking to 6/100. Attack of Elephants are represented on the visual interface of the game (Fig. 9), and losses caused by elephants are represented by a blue diamond on the corner of each attacked field.

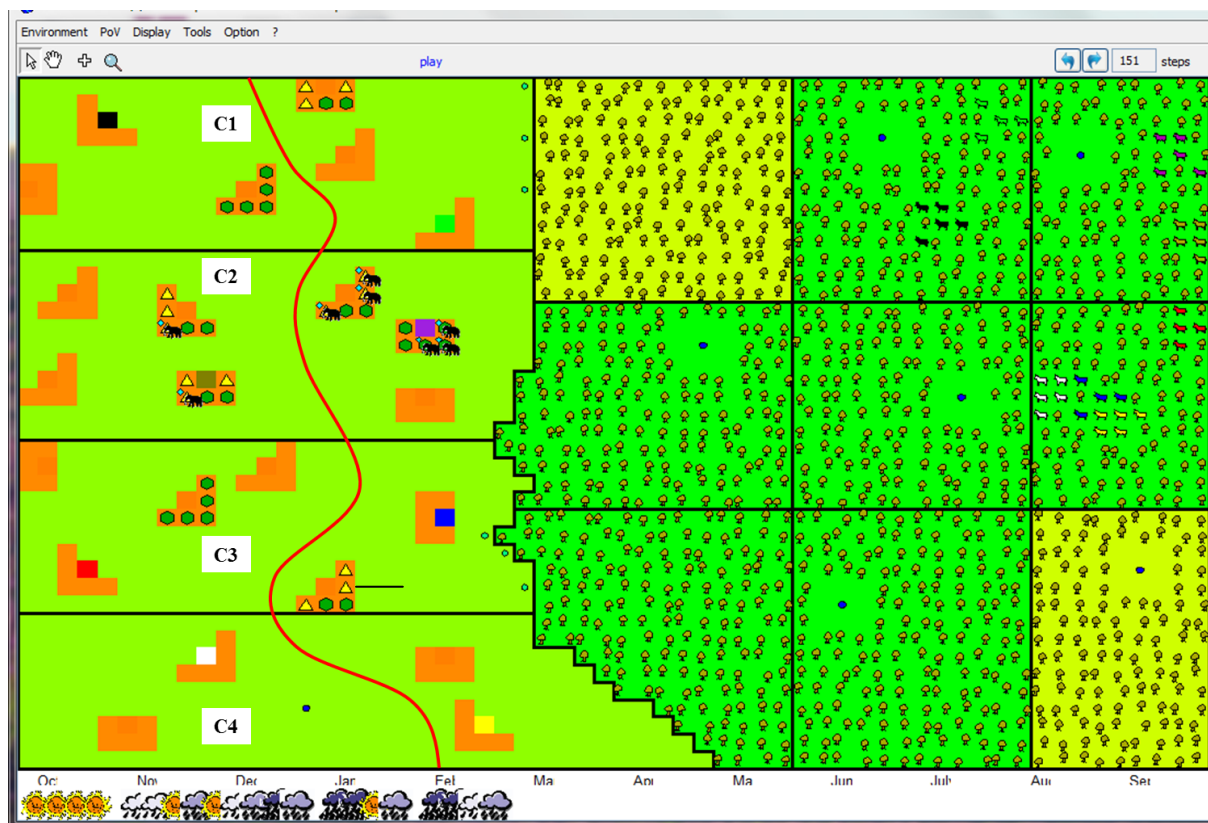


Figure 9. Elephants' damages. Grazing areas with mature maize (here C1, C2 and C3) are prone to crop raiding by elephants. C2, which was not guarded (no small green circles at the edge between the communal land and the forest) was attacked by elephants. Farms located on the right side of the red line are considered as the first line of farms. These farms are more heavily impacted by the attack.

Details for cattle damages

In the paper, we described the rule according to which cattle cannot be released freely in the village before a particular date: *Xotshela* (usually the 1st of June). Nevertheless, it was collectively agreed that the game would contain no enforcing mechanisms, but that the players themselves would be free to discuss, apply and respect this rule or not during the game. Therefore, in the model supporting the game, cattle can graze anywhere anytime and when grazing in the communal area during the agricultural season, cattle entities can enter

fields and cause agricultural losses if these contain crops. Unguarded cattle (without herdboys) will enter a randomly picked field with growing maize, and cause 25% loss of the yield the owner would have get from the maize entity growing there. Repeated losses caused by cattle or wildlife will ultimately destroy maize entities. A given cattle can enter only one field per month. Once the module is run, the visual interface is updated: a message is displayed on the game's interface to inform the players: "*Player...your cattle entered the field of player...*"; the cattle entity is displayed on the field it entered and the losses are represented on the field (Fig.10); the loses are stored and the yields the player will get when he harvests are modified.



Figure 10. Cattle entering a Field. *The cattle of the purple player were left alone to graze in this communal paddock containing growing fields. They entered a growing field of the olive player. The white lozenge on the corner of the field of the olive player signifies a loss of 25%.*

Details for cattle status dynamics

Every month, the level of satiation (0, 1 or 2) of a cow is determined accordingly to the level of forage of the grazing area where the cow was located. When cows are feeding on crop residues (*machanga*), the satiation is set to the number of *machanga* eaten (max. 2). The status of a cow depends on how it fed during the last three months: when its accumulated

satiation is 6, its status increases, whereas when the accumulated satiation of a cow is less or equal to 3, its status decreases.

Details for cattle predation by lion

The predation module was entirely re-designed during workshops. This module is applied at the level of a grazing area. To allow comparison of the role-playing game sessions, the randomness assigned to this process was eliminated by pre-defining the occurrences of cows being killed by a lion in the forest. Numbers are given in table 2.

| GrazingAreas in the forest | Occurrences of presence of cattle in the subarea area at which a kill will be considered |
|-----------------------------------|---|
| Nearby forest (F1, F4, F7) | 3 8 14 |
| Middle forest (F2, F5, F8) | 2 5 9 12 16 |
| Deep forest (F3, F6, F9) | 1 2 4 6 7 9 11 12 14 |

Table 2. Predefined timing for cattle predation by lion.

In the nearby forest, every 5-6 times some cattle will be there, a kill by the lion is considered. In the middle forest, the periodicity is decreased to every 3-4 times, whereas in the deep forest, the periodicity is set to every 1-2 times. The frequency of grazing and the distance to the forest’s edge are therefore the two factors taken into consideration to trigger the event “lion’s kill” in the forest. Once such a kill is considered, its actual realization depends on the presence of herdboys. When all cattle are guarded by a herdboy, the probability that the kill fails is 2/3. Otherwise, the kill will occur and it is 30 times more likely to happen to a non-guarded cow than to a cow being guarded.

Details for forage level in grazing areas and water levels in water pans

These two modules rely on tabulated functions. The forage level of each grazing area is updated at the end of each month according to: (i) the month, (ii) the rainfall of the current month, and (iii) the number of cows that grazed during this month. Furthermore, a specific

function was designed for each climatic year (Fig.11 and Fig.12). An additional feature was decided during the co-design of the model: the four communal grazing areas (C1, C2, C3 and C4) would never reach the “good” level of forage, but would remain “medium” at best. The size of the water pans is also updated at the end of the month. It is not influenced by the number of cows drinking, but relies on a tabulated function designed according to monthly rainfall (Fig. 11 and Fig.12).

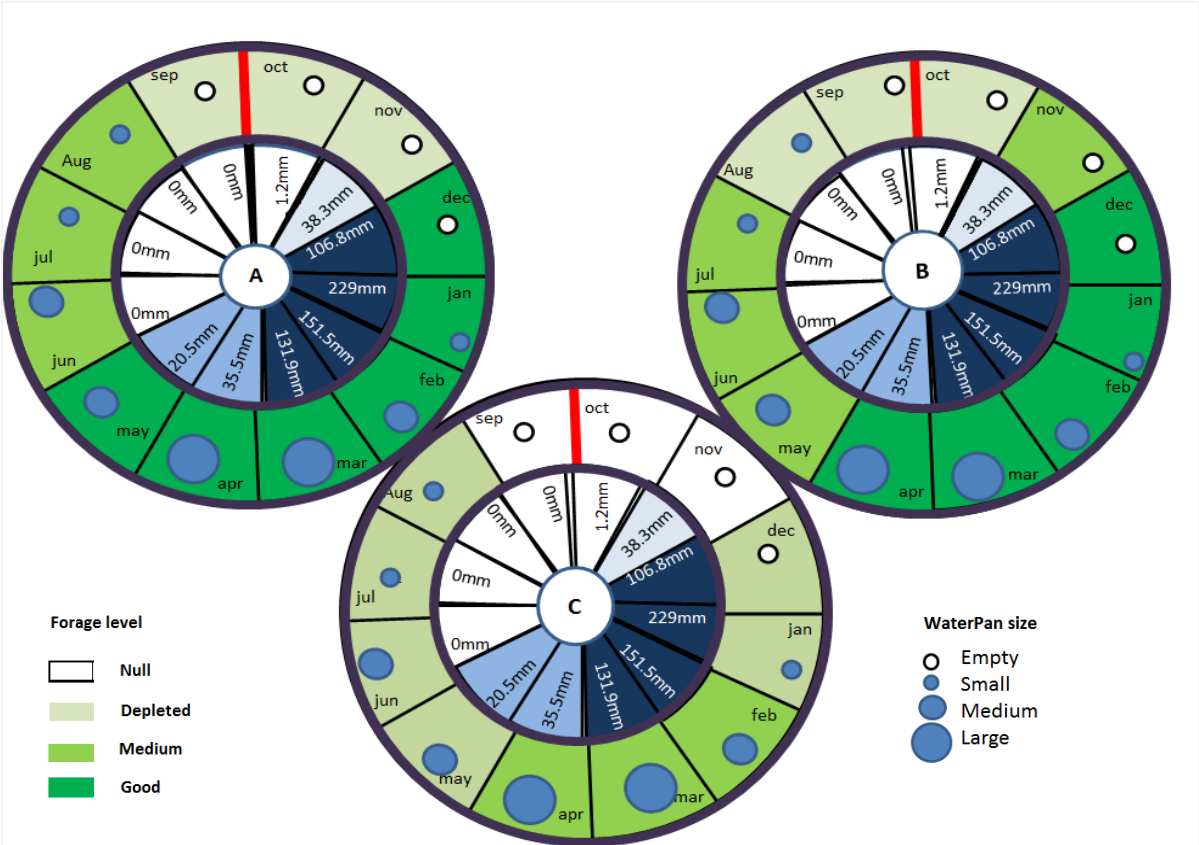


Figure 11. A representation of the tabulated function controlling the forage level of grazing areas and the size of water pans for the “Good Rain” year. The three diagrams represent the forage level at the end of each month in a given grazing area according to the number of cows grazing during the month. A: no cow grazing; B: between 1 and 10 cows; C: more than 10 cows. The circles in each month represent the size of water pans. Monthly rainfall is given on the inner wedges.

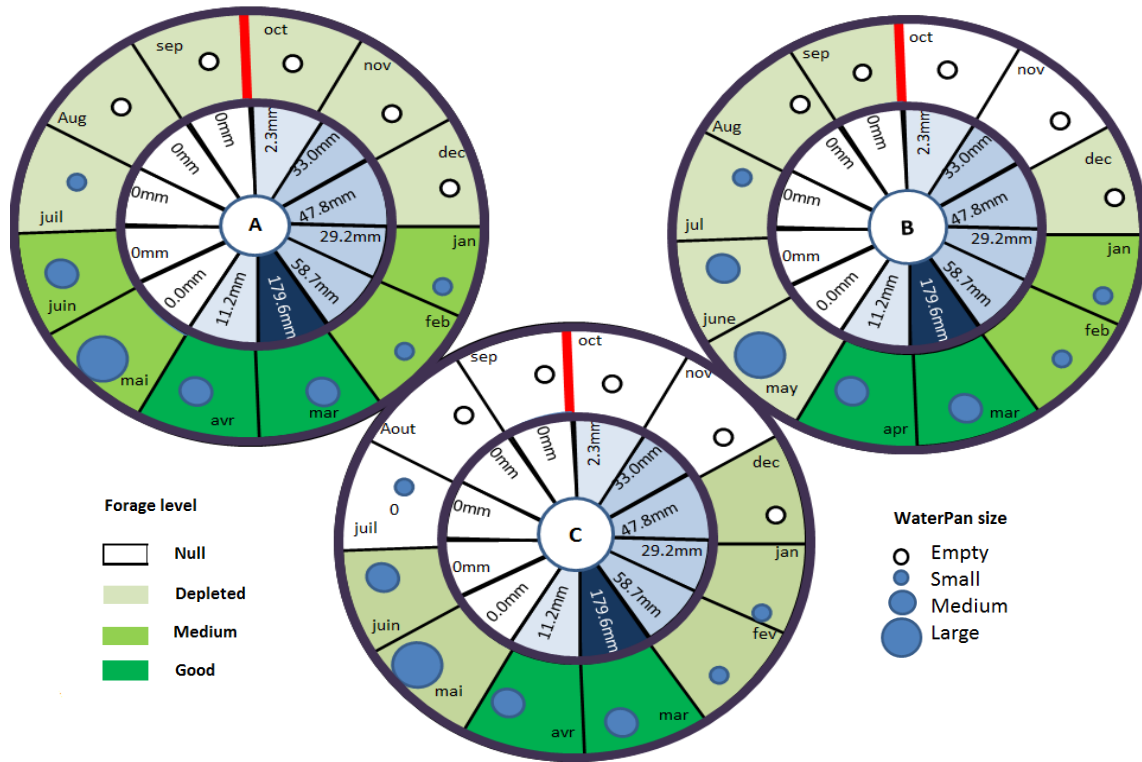


Figure 12. A representation of the tabulated function controlling the forage level of grazing areas and the size of water pans for the “Bad Rain” year. The three diagrams represent the forage level at the end of each month in a given grazing area according to the number of cows grazing during the month. A: no cow grazing; B: between 1 and 10 cows; C: more than 10 cows. The circles in each month represent the size of water pans. Monthly rainfall is given on the inner wedges.

Choice, design and parametrization of the modules

All modules were either collectively designed, or proposed by researchers and modified/validated by the other members of the team. The testing of each module was done through the co-design process. Each version of the game was tested (played) by the team, collectively discussed and modified/validated.