Farm It Right

Project Report – Prepared by Deb Polson and Dave Wallace

25 Sept 2008

Farm It Right is the working title for an interactive simulation (prototype) for educating university level archaeology students about sustainable farming practices. Based on an **ecological model** of the **9th century Nordic region**, it allows students to compare statistical and visual data to appreciate the effects of modifying various input parameters to the model.

The user is tasked with simulating the grazing of sheep on an area of grassland. A successful outcome is if the player manages to farm their land in a sustainable way for a full ten years. A failed outcome, for example, may be that all the sheep die before the ten years have expired. Maintaining a sustainable population of sheep is a balance of ensuring enough food is produced (by grazing or hay production) to feed the sheep whilst preventing overgrazing of the grassland (which reduces the grazing potential and in turn the number of sheep that can be supported by the land). The player can adjust various parameters to affect this balance (or imbalance).

The Client

The client was Professor Armin Shmidt from the University of Bradford, England.

The Team

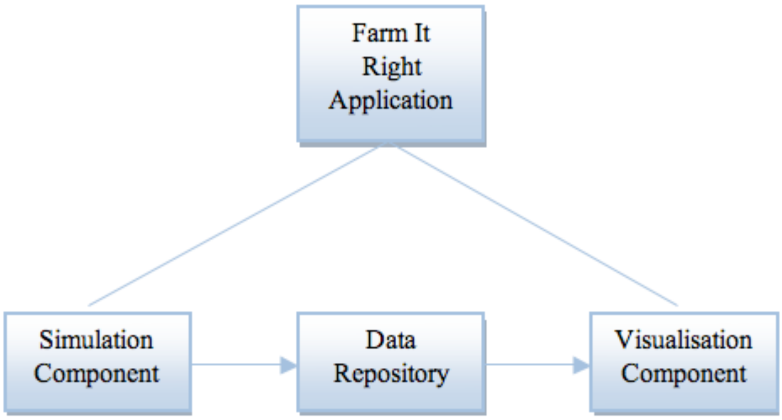
Lead Design: Deb Polson
Programmer: Dave Wallace
Visual Production: Shu-Min Heng

The Proposal

It was proposed to create, in **three weeks**, a “simulation with game elements” as an aid for educating university level archaeology students about sustainable farming practises. It would be based on an **existing scientific model**, and allow the user to see graphically the effects of modifying various input parameters to the model. The budget was small for the nature of the project and the time to complete was very limited. For this reason it was essential to spend time early with the client negotiating the actual scope and feasibility, while still maintaining maximum potential within the project for accurate **data visualisation**, dynamic **user interaction** and **knowledge transfer**.

Design and Development

The application is, in structural essence a simulation and data visualisation, similar to previous projects completed by this team (e.g. SCAPE). The development was modularised along these lines as demonstrated by the following diagram: It was decided to develop the application in flash for the usual reasons: rapid prototyping of GUIs, easily distributable, familiarity etc.



A main “Game Model” doc which had already been partially completed by the client was primarily used to guide development. The final development model was a compromise of features from this based on scoping constraints.

A partial prototype (modelled in Pascal) was supplied as a initial reference. The relative completeness of this existing model afforded a more rapid porting of the simulation logic to the final flash application; therefore most of the development time was able to be spent on enhancing the user interaction with the graphical interface.

It should be noted that the decision to encapsulate the simulation in a game-like context allowed the scope of development to be controlled to some extent because it freed the application of the expectations of detail and accuracy of a full blown scientific model, whilst still being an accurate demonstration of broader concepts.

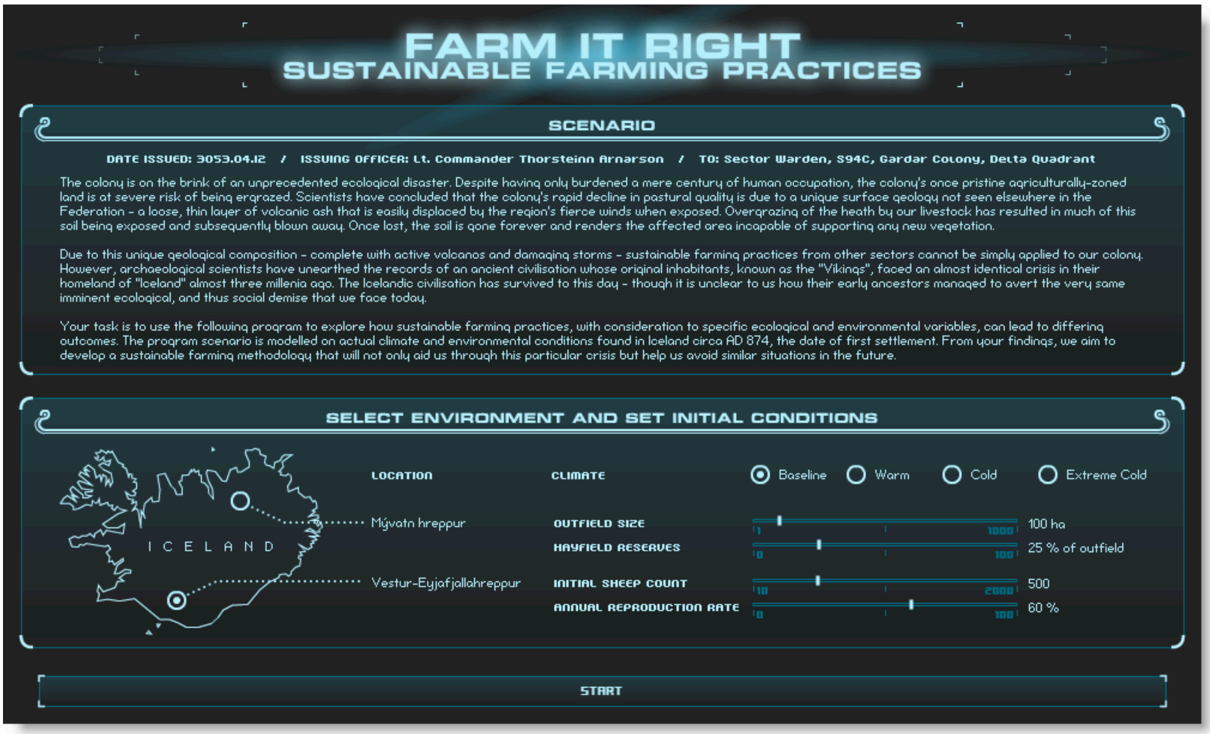
Design Considerations

One of the challenges when designing the interface was attempting to encapsulate a vast amount of visual feedback to the complex equations of the simulation model in as few screens as possible.

At the behest of the client, the simulation was set to model a real period of time around 880A.D. in Iceland (from which the environmental and climatic data sets were derived). In order to nurture the game element, it was decided to create a narrative framework (scenario) around the simulation where by a player assumes the role of a fictional person who has travelled back through time charged with overseeing the colony through remote console controls (in the vein of Star Trek) in an effort to understand what went wrong in the past and to experiment with the possible effects of alternate farming decisions. The aesthetic style of the interface draws inspiration from both these diametric influences (Nordic tradition and futuristic control interfaces).

There are two modes of user input to the simulation; some parameters can only be initialised once at the outset, and others are adjusted throughout the course of the simulation. The interface was divided up into two main screens, based upon these different modes of input.

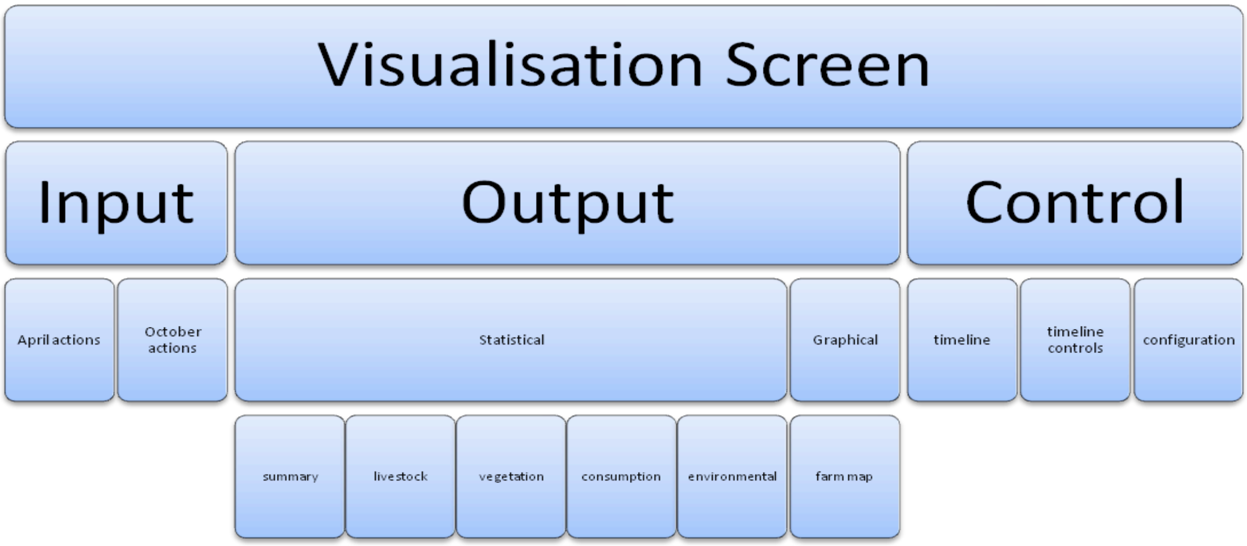
Scenario Screen



The initial screen has a short introduction to set the scene and explain the narrative context of the simulation. It also contains a section for the setting the start parameters (location, climate, outfield size, hayfield reserves, sheep count and reproduction rates) for the first mode of input.

Visualisation Screen

This interfaces is split into two halves, the **statistical** and the **visual** data representations. The player can pause the timeline over the ten years at certain intervals to adjust some settings such as the number of livestock slaughtered. There are a number of 'tabs' players can click on to view and adjust further settings. The visualisation screen abstracts the main mechanics of the simulation. It is logically divided into the following sections:



The most salient component of the visualisation screen is the farm map as the user can quickly tell a lot about the 'health' of the farm by observing it. The state of overgrazing, as well as changes the sheep population is immediately apparent. More detailed statistical feedback about the vegetation, livestock (weights and numbers) and consumption are displayed in tabbed panes on the left. The most important of these are displayed on a summary tab, including the most important of all – the maintenance : consumption ratio (MCR). This is the product of the main equation determining the sustainability of the farm. It can simplistically be described as the amount of feed produced by the land verses the amount of feed required to keep the sheep from starvation. An MCR above 1.0 defines an unsustainable situation where overgrazing and starvation will occur (and will be reflected in the farm map)

Game Play Model

The user is tasked with simulating the grazing of sheep on an area of grassland. A successful outcome is if the player manages to farm their land sustainably for a full ten years. A failed outcome is if all sheep die (for whatever reason) before the ten years have expired.

Maintaining a sustainable population of sheep is a balance of ensuring enough food is produced (by grazing or hay production) to feed the sheep whilst preventing overgrazing of the grassland (which reduces the grazing potential and in turn the number of sheep that can be supported by the land). The player can adjust various parameters to affect this balance (or imbalance). Some of these can only be set once at the outset including:

- Location – north or south of Iceland
- Climate - along with location affects the temperature ranges
- Outfield size –initial area of grassland available for grazing
- Initial sheep count
- Annual reproduction rate

Other parameters can be adjusted at interval throughout the simulation, which correspond to real decisions that might be made on a sheep farm at certain times of the year. These include:

- how much land to set aside for hayfield
- how many sheep to slaughter

- how many sheep to take into the byres and feed with hay versus how many sheep to graze

Climatic events such as storms and volcanic eruptions are included to introduce a degree of randomness to the simulation.

The typical sequence of events for the simulation is as follows:

1. Set initial parameters
2. Run simulation
3. Observe statistical and visual feed back to determine if food production and consumption is balanced
4. If not enough food is being produced to feed the sheep or overgrazing is occurring, take action to reduce the food production/consumption ratio (e.g. by reducing the number of sheep by slaughter or feeding more sheep on hay to prevent further overgrazing)
5. Continue until the timeline has concluded.

Development process

Both the budget and timeline were tight for the nature of the project. Splitting the application into its simulation and visualisation components helped compress the development timeline and make maximum use of our development resources. Once the functional requirements had been specified the simulation was able to be developed independently, in tandem with the design of the graphical interfaces. After these were both complete they were able to be interfaced via a main controlling parent component (see component diagram above) in a relatively straight forward manner.

Development of the first prototype was completed to an agreed level of functionality roughly three weeks after the initial design was locked down. After a period of user testing conducted by the client, further feature additions and improvements were made to the prototype after additional funding was secured. These additions addressed both design ideas that were out of the scope of the initial development as well as feedback from the evaluation process. Many more potential future additions were also identified to increase the scope and relevance of the simulation/game.

Client Relationship

Managing the scope of client expectations in this instance was particularly challenging due in no small part to the fact that all communication with the client was done via proxy to the other side of the world. In particular it became necessary to impart firmly and repeatedly the importance of considering as much of the concept design as possible before the start of development to avoid revisions in the development cycle and stress already tight timelines.

As is the nature of such domain specific projects, a significant amount of specialised domain knowledge had to be assumed by the designers and developers. At the same time, the expertise of the client had to be trusted and interpreted whilst not requiring the client to understand the specifics of the design methodologies or development processes.

Although the client's initial expectations of the outcome of the project were rather too grandiose for the size of the proposed budget, these expectations were managed and ultimately met and in some parts, exceeded (based on client feedback).