Kokatrix history simulator

Sami Mäkelä

January 23, 2004

Chapter 1

Introduction

Kokatrix history simulator can be used to generate worlds with several communities that have buildings and inhabitants with relationships. Whole history of the world is stored.

1.1 Time

Time is always represented as a floating point value. It's the time in years since the beginning (0.0).

Chapter 2

Elements

Entities can be from different races which determines average age and some other attributes. Entities also belong to countries that determine a great deal of their behavior, like how often they move, how many children they have, on what age they get married etc. Also communities belong to countries. Communities are used to determine the location of entities so that it is easy to implement event scripting and some other things. There are also buildings, that are used for living and working in. Different types of buildings can be defined. Entities also have professions, they are used to determine in what kind of buildings the entities want to live or work in. Finally there are different types attributes which can add events to person's life story. Attributes can also be inherited.

Races, country, profession and other attributes are used together to determine an entity's lifestyle. This means that for example the country or an attribute of a person can change the person's age. The difference is that the average age must always be determined in the race.

2.1 Races

Races are supposed to be things like humans, orcs or elves. They have different physical attributes for males and females. The attribute Sexes tells how many of the born creatures of this race are females or males.

Important attributes for races are:

- Average age is actually a random variable that tells the maximum age of the creatures with this race. In the example I use normal (Gaussian) distribution. I'll need to write a tool that can be used to experiment what kind of values different distribution generate. The editor has following problem: in the menu button, the label is not changed when the edited element is changed. This is a problem with the underlying GUI library and there is nothing I can do about it.

- Mature age tells after which age the creatures can start having children.

- Pregnancy just tells what is the minimum period between getting children. Ignored with males.
- Number of children tells how many children is got when giving birth (ie. how often gets twins). Ignored with males.

To copy the data from males to females or from females to males, use the Edit-menu.

2.2 Countries

I guess these countries are more like cultures than actual countries. Anyway communities have a country they belong into, and persons too. Currently all countries must be predefined, but it's easy to add a feature to make copies of countries.

They have these attributes:

- Child death is the ratio of children that die when giving birth.
- Child period is the time between getting children. If the season modifier is enabled, it can be used to add seasonal variation.
- Children is the number of children.
- Marriage age tells when the people start getting married.
- Marriages tells how many times the people usually get married. Currently there is nothing to handle the case that spouse dies. Perhaps this can be done with the death event attribute similarly to succession.
- Then there are more attributes to control marriages and inbreeding.
- Names includes the name lists.
- Events are events that happen to persons with this country. Most normal events are probably best defined here. See chapter 3 about events for more information.
- Fallbacks if some event doesn't succeed, fallbacks get called instead.

2.3 Professions

Professions are used to determine in what kind of buildings the persons want to live. Also stuff like lords, kings and wizards are professions.

Professions should define following attributes:

- Lives at work tells the ratio of living in the same building as the workplace, for example the king should always live in a castle. Does not always happen because for example spouse wants to move. Attribute Homes adds more control.

- Wealth tells the wealth of the persons with this profession. This is used to determine what kind of buildings the persons can build.
- Events can be used to add special events related to profession.
- Homes tells in what kind of buildings the persons with this profession can live. The first attribute tells the preference, bigger the better.
- Works tells in which buildings these people might want to work. Starting to work might require profession change.
- Build attributes tell what kinds of buildings these people might make.
- Close professions tells into which another profession persons might change.
- Marry event is executed for the spouse. For example the spouse of the king becomes queen.
- Death event is executed after the person dies. For example when the king dies, his successor will become new king.
- Requirements is a requirement that the person must fullfill to have this profession. See section 3.1 about requirements for more info. For example a wizard needs to be talented in magic.

2.4 Communities

Communities are towns or other communities. They belong into a country and they have buildings. Communities can affect to the lifestyle of the persons that live in it in the same way as countries, races and professions. Communities can be predefined, or they can be founded by persons.

Communities can have attributes, which tell what kind of places they are. The attributes can be got from the map, where different colors represent different attributes. The Info/Colors field can be used to define what different colors mean. The attributes can also be added and removed as events (needs to be fixed, currently can only set the whole list).

2.5 Buildings

The creatures in the simulation can build different kinds of buildings and live in them. The features of the buildings are determined with building types.

Building types have following attributes:

- Inhabitants tells how many persons can live in the building.
- Price tells how much wealth the builder must have.

- FAMILIES tells how many families can live in the building. The persons without families are ignored.
- OWNER is a job that can only be done by the owners of this building. (This system probably needs some work).
- JOBS lists the other jobs in the building.

To control how many buildings of certain type can be built into a community, following dependancy attributes can be used:

- PEOPLE tells how many persons the community must have before this building can be made.
- BUILDINGS tells how many buildings of some other type must be in the community.
- ATTRIBUTES tells what attributes the community must have. I'll have to add amounts to town attributes so that one building needs certain amount of a resource.

2.6 Artifacts

Artifacts are important items that the persons can find or make. They can be given to other persons. The attributes GIVE EVENT and GET EVENT are executed when this item is got or given.

2.7 Persons

The lifestyle of a person is determined by it's race, country, profession and community. In addition to these, a person might have special attributes. The important fields for special attributes are:

- INHERIT tells how probably it's for a person's children to inherit an attribute. If both parents have the attribute, it's just tested twice.
- STRENGTH is just an amount which tells how strong this attribute is in a person.
- TYPE. When an attribute belongs to a type, a person cannot have attributes with same type.
- MUTATIONS. When an attribute is inherited, it might be changed to another one.
- GENE STRENGTH tells if the attribute is recessive when used as a genetic attribute.

The initial persons in the world can be created using SEED event. It's also possible to make predefined persons. These persons can then be born into the world with a special event (not yet implemented).

Chapter 3

Events

Predefined events can be added to the world. The target of an event can be a person, building, community or whole world. First part of an event selects it's target. For this selection, *requirements* can be used.

3.1 Requirements

Simplest form of requirement is just to have a *test*. This won't work if you want to select best from several alternatives. Then IF-THEN-ELSE can be used to decrease or increase how good the alternative is.

A test is a proposition in or-and-not normal form. It should be easy to figure out how this works.

3.1.1 Communities

Following atomic tests can be applied to communities:

- INHABITANTS tests if the community has more or as many inhabitants as the specified amount. If the test is negated, tests if it has less inhabitants.
- COUNTRY tests if the community belongs into a country.
- ATTRIBUTE tests if the community has an attribute.
- PERSONS tests if the community has required persons.
- BUILDINGS tests if the community has required buildings.

3.1.2 Persons

Following atomic tests can be applied to persons:

- ADULT.

- Male tests for sex.
- Age.
- Profession.
- Lives at tests in which community the person is living.
- Birth place.
- Home and Work test the types of buildings.
- Country.
- Race.
- Mother and Father make tests for persons parents.

Tests for different elements can be tested by using the Complex editor menuitem, and then button Add item. I'll have to add a tool to test personal requirements for persons.

3.2 Selection

Persons can be selected in the following ways:

1. Single person just specifies the person. This is used for generated events.
2. Multiple persons selects amount of persons using a requirement. The persons that have the best result in the requirement are selected. If some randomization is wanted, it's possible to use the Mod requirement.
3. Town selects amount of persons from a town.
4. Percentage selects a rate of persons from a town.
5. Relatives selects relatives of a person.
6. Named selects named persons relative to the event or person. Not yet implemented.

Following relations are implemented:

- Child. For a female, the children are just the children that have been born to her. For a male, the children are the ones that have been born to his wife while they have been married.
- Parent. Real parents.
- Building relation holds if they live at same building.

- TESTED adds a test to the relation.
- COMBINED combines to relations. For example COMBINED(CHILD,CHILD) returns grandchildren.
- RECURSIVE applies the relation recursively until a limit is hit. For example ttribRecursive(Tested(Male,Parent),8) returns 8 male ancestors of the person.
- BEST uses requirement to select best of the relatives.
- SUCCESSOR implements succession calculation. First parameter tells which parents are used in calculation, and second one which descendants. The test is a test that tells if a person is good successor (for example the successor must be alive).

3.3 Life events

Lifestyles include these kind of events. They have following fields:

- EVENT is the event to execute.
- AMOUNT tells how many times the event is executed.
- START tells when the event is first executed, relative to the birth of the person.
- PERIOD tells the period between rest of the events.

3.4 Person events

Here is a list of events that can happen to persons:

- DIE with a reason.
- MOVE TOWN moves to different town.
- FOUND TOWN moves to a new town.
- MOVE BUILDING
- CHANGE JOB
- BUILD BUILDING
- BUILD HOME
- RETIRE
- FIND JOB

- Independent
- Give artifact
- Change profession to a specified one.
- Get attribute
- Remove attribute
- New country changes country.
- Build a building of specified type.
- Work in a building of specified type.
- Move to a bulding with specified type.
- Find item of type.
- Give item of type.
- Participate event with role.
- Cause event is used to cause global events.
- Choice chooses random event from list.
- Conditional events first execute a test. Based on the result one of the events is executed.
- Composite events have an initiator. For the initiator, an event is executed. Then the composite event can select other persons, for whom events are also executed.

Events can be given names in the Info editor. Then they can be executed using Named event.

3.5 Other events

For communities, there are two important events, Seed and Epidemic. The seeding event addss Amount new persons to the community with specified race, profession and country. The Genome can be used for attributes like skin, eye or hair color. The Epidemic event can be used to add diseases or other similar stuff.

The Disaster event can be used to add events for several people in the specified area. Attribute Duration tells how long time the disaster lasts. Other events for world tell

Chapter 4

Other stuff

4.1 Timelines