

# Multi-agent based simulation of G. R. R. Martin's Sand Kings

Jakub Ciecierski, Viet Ba Mai,  
Michal Slupczynski and Wojciech Zyskowski  
Faculty of Mathematics and Information Science,  
Warsaw University of Technology  
Plac Politechniki 1, 00-660 Warsaw, Poland

**Abstract**—This document describes the multi-agent based simulation of George R. R. Martin's *Sand Kings* created for the Agents Systems and Applications course at the Warsaw University of Technology. Application's main purpose was to analyse diplomacy between profit-driven entities in hostile environment, described in the novelette above.

## I. INTRODUCTION

**S**AND KINGS is a science-fiction novelette written by George R.R. Martin in 1979. Simon Kress, its main protagonist, is a collector of lethally dangerous and exotic animals. Due to his prolonging business trips they often die during his absence. Eventually, when a need for replacement occurred, he stumbled upon mysterious establishment, in which he found terrarium filled with four Sand King colonies. Each colony consisted of female, immobile Maw and number of insect-like mobiles that are controlled by their Maw through telepathy. Mobile's main purpose is to hunt and collect food for the Maw to digest, which only later they are able to feed on. The shopkeeper informed him, that the colonies would start to wage wars between each others with time. Excited with the vision, Kress bought four Sand King colonies and decided to have them installed in his flat. With parties to show off new pupils, Simon couldn't wait for the conflicts to emerge. As Sand Kings lived peacefully for the days to come, he started to starve them so they would become desperate. From graceful and highly intelligent entities they've turned into wild and murderous creatures that sought only to find more food to grow, which eventually have broke out of the terrarium they were imprisoned in.

The simulating application focuses on the stage slightly after extreme starvation of Sand Kings, when the only goal driving them was to amass extreme amounts of food. In order to do so they would kill the creatures that were thrown into terrarium, but also fight with each other and potentially kill - if that would prove profitable.

## II. TOOLS

Repast Symphony is an open source agent-based modeling toolkit that simplifies model creation and use. Out of wide variety of accessible tools, we have chosen it for the development of the simulating application, mainly due to its ease of use and possibility of run-time dynamic interaction with

the simulation. Additional perk of the Repast toolkit was the extent of tutorial materials and quality of their description.

## III. METHODOLOGY

Small introduction to methodology

### A. Agents

What is an agent. Mobiles, Maw, God, Monsters etc.

### B. Environment

1) *Map*: The map is a 50x50 grid consisting of different shades of yellow and grey to represent the environment described in *Sand Kings*, where the terrarium was filled with sand and rocks.

2) *Food*: In the simulation there are 5 types of food with different weight and calories which are proportional. They affect respectively how many mobiles are needed to carry each and how much maw's strength is increased by eating it. The first four types of food are dropped by the God agent. They are represented by **pizza**, **doughnut**, **grape** and **cabbage** icons, where the first one has the highest calorie value. The last type of food is dropped only by a living agent (either a mobile, maw or monster) when it dies. It is shown as **meat** in the simulation. For balancing purposes meat gives lesser calorie values than any other kind of food and amounts to roughly half of the mobile price. Depending on the agent type different numbers of meat are dropped, according to rule that the stronger initially is the agent the more food it will drop when dying.

### C. Communication

A simply mechanism was constructed to create an inter-agent communication. Each agent can send a message to another by adding a packet i.e. the message, its sender and recipient to the message queue. The queue notified by change will process a given packet, and fire a specific handler for the message type. Based on the sender's needs and the recipient's current (TODO storage ??), a response is sent. The usage of messages ranges from informing other agents of their findings in the environment like food or enemies, to even the combat system.

#### *D. Knowledge Base*

Things about knowledge, communication etc.

#### *E. Scheduler*

Agents' actions are

### IV. SIMULATION

Small intro to simulation. What are we planning to do in this section.

#### *A. Data set*

What do we have in simulation. Environment, maws, mobiles, how many etc.

#### *B. Results*

Results of our simulation.

### V. CONCLUSIONS

Conclusions of this paper.

### APPENDIX

Is it even needed ?

### ACKNOWLEDGMENT

Some thanks to Paprzycki ? hahahahaha for what

### REFERENCES

[1] Source example, for more add sources here