The Giving Game: Design Document

The Giving Game

April 1, 2015

University of Amsterdam

Julian Ruger 10352783

Contents

1. Introduction

I decided to create the simulator with Python(https://www.python.org/). I will be using the newest version of Python, Python 3.4, because this allows me to make use the full potential of Python. Python is a fairly easy programming language. The syntax allows me to write the program in fewer lines of code. With Python I do not have to specify the type of variables and the memory space is automatically allocated and deallocated. Python is also supported by alot of different packages and frameworks which can be used to write the code in even fewer lines. For example I will be using the Qt framework for the GUI. Python will save me alot of time and I only have about six or seven weeks so that is why I have chones Python. The only problem with python I might run into is the performance of the simulator. Because Python automatically allocates the memory space it might be less efficient using Python when I need to use alot of memory space. During the six/seven weeks of programming I will need to pay good attention to this problem to make sure the simulator will not get to slow at executing the tasks.

2. The simulation model

The simulator can be divided into three major parts: the algorithm implementations (Back-end), the visualisation (results and graphs) and the GUI. The simulator can be seen as the following proces:
Picutre of the Proces here!
Input

- Selection rules: random rule, balance rule, goodwill rule etc.
- Parameters: Time, N people, M products, value of products etc.

Output

- Results: Community effect, number of transactions etc.
- Graphs, functions that show the results over time etc.

Rules for the simulator

The simulator must be able to execute multiple selection rules with multiple parameters, be able to handle a large amount of agents (1000-10000) and be able to show results during the giving game.

3. Back-end

Packages: Numpy,
I made the following design choices for the course of the giving game.

- The player will be a node in a peer-to-peer network where every node keeps track of its saldo with the other nodes. To accomplish this I will be using a object oriented structure to create this network.
- For the calculations of the saldo and other variables I will make use of the numpy package.

Code structure

Memory management

A dictionary will be used by every node to store the information about other nodes. For example if we have a node P and a node Q the dictionary of P will look like {'Q' : 'saldo'}.

4. Visualisation

Packages: Numpy, matplotlib, pychart
graphs used to plot the results. Results will be saved in a file.

5. GUI

For the GUI I will be using the Qt framework.