ETHlogo

**Lecture with Computer Exercises:**

**Modelling and Simulating Social Systems with MATLAB**

Project Report

|  |
| --- |
| Insert Title Here  … |

Patrick Barton & Lars Beglinger & Liam Brennan

Zürich

17.12.2017

**IMPORTANT**

**You MUST include the ETH declaration of originality here; it is available for download on the course website or at**

**http://www.ethz.ch/faculty/exams/plagiarism/index\_EN;**

**It can be printed as pdf and should be filled out in handwriting.**

**Agreement for free-download**

We hereby agree to make our source code of this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

|  |  |  |
| --- | --- | --- |
| Patrick Barton | Lars Beglinger | Liam Brennan |

# Table of content

[Table of content 4](#_Toc501222869)

[Abstract 5](#_Toc501222870)

[Individual contributions 5](#_Toc501222871)

[Introduction and Motivations 6](#_Toc501222872)

[Description of the Model 7](#_Toc501222873)

[Implementation 7](#_Toc501222874)

[Simulation Results and Discussion 7](#_Toc501222875)

[Summary and Outlook 7](#_Toc501222876)

[References 7](#_Toc501222877)

# Abstract

# Individual contributions

GitHub allowed us to work individually on our project. After sitting together for a few hours, brainstorming and defining upcoming tasks, the code from each contributor has been coding and debugging on their own for the most part.

Lars and Liam started off by some research into the twitch API and how this data is being used already, trying to find a reliable source for our project where we want to reproduce existing, measured data with our model.

In a next step, Liam made a skeleton with the core functions evaluating the viewer and streamer compatibility, focusing on the stochastics to get our desired normal distribution. Lars then extended the core and added more mechanisms like a factor of uncertainty before Patrick introduced the time dimension as well as the individual refresh rate. Additionally, he plotted our results similarly to the format the measured data is available.

From there on, Lars added additional elements influencing the dynamics such as a streamers’ schedule a viewer threshold and viewer refresh inertia. To finish it off, Lars and Liam worked on an efficient and clean way to present the data.

To sum it up, our efforts were well distributed amongst the team members and we all had our field of expertise: Liam focused on the statistical aspect, Patrick on the dynamical part and Lars was responsible for the implementation of real world mechanisms influencing viewership in MATLAB.

On a side note, it should be noted that all three team members are part of a very time-consuming focus project. Therefore, we had to limit the time for this simulation, even though further extending the MATLAB script would be incredibly interesting.

# Introduction and Motivations

Twitch is a live streaming platform mainly focused on video games. Since its launch in 2011, it has grown rapidly and and now provides entertainment for 15 million unique daily users (twitch.tv/about), ranking it in the top 50 of the most frequented websites in the world. (alexa topsites)

Everyone can set up their own live stream, build a community and show off their video game skills and entertain their viewers. Gamers streaming the same game are often competing against each other for viewers. Unlike in traditional television, there is a two-way interaction between the streamers and the spectators through a chat, adding a new layer of complexity to the viewer count dynamics.

We all are frequent visitors on Twitch: There, we seek entertainment, we are following eSports tournaments and we are watching professional gamers play on an extremely high skill level. We often were amazed by how many viewers a streamer can attract, and ultimately by how much money a handful of streamers earns, all that while playing video games. We noticed how some streamers start their broadcast right around the time the most popular channel goes offline, effectively offering an alternative to the sudden lack of entertainment and trying to attract some bored viewers that would never tune in if more popular people are streaming. It is a truly fascinating and complex system and in this course, we saw the opportunity to analyse that behaviour in detail on a scientific level.

Even though we are not planning to start our own Twitch streams, it undoubtedly is interesting to find out how a channel might be able to substantially increase their viewer count by some basic improvements.

# Description of the Model

Undoubtedly, the human mind is

# Implementation

# Simulation Results and Discussion

# Summary and Outlook

# References