# Algorithms and Complexity for Video and Computer Games
## *Project Specification*

André Nyström & Axel Riese

February 12, 2015

## 1 Introduction

Our chosen general subject is algorithms and complexity in computer games, specifically the 2004 puzzle game "Lumines". Our project will concern the computational complexity of one or several goals related to playing the game.

The subject is of interest because Lumines is to our knowledge not very well researched when it comes to computational complexity. Similar puzzle games, such as Tetris, Bejeweled and Puyo puyo has been more thoroughly examined in recent years, and it seems interesting to us how Lumines will compare. While these findings may seem to be of lesser importance, they may teach us more about the nature of computational complexity, or aid in reductions to other problem instances.

## 2 Problem statement

The essence of Lumines is to create as many $2 \times 2$ monochromatic squares (blocks) as possible, using a particular sequence of $2 \times 2$ dichromatic squares (pieces).

The problem we want to answer is "Is Lumines NP-complete?". More specifically, we want to focus our attention on the problem of maximizing the number of blocks one can create using a priorly known sequence of pieces and an initial gameboard.

## 3 Approach

Our approach to the problem is at first to simplify the game mechanics, for example we will probably consider the "sweep mechanic" of the game to be instantaneous instead of moving. We will also consider the offline version of the game, where every piece involved in a game sessions is known beforehand, rather than the actual online version, where the game pieces are generated probabilistically.

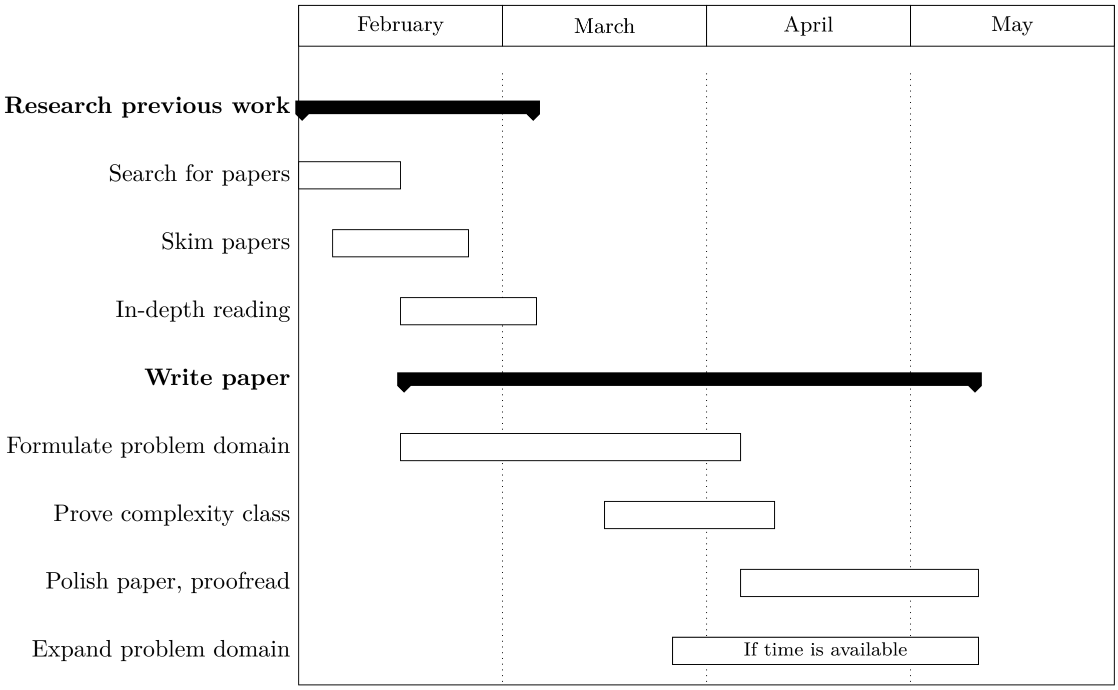
We will then start to define the rules of the game in a rigorous manner. Once we have a mathematical system, we can proceed to formulate a decision problem of interest in that system. After this is done, we can try to prove that the problem is in fact NP-complete. This is done in the standard way; we will first try show that the problem is in NP, by proving that an answer is verifiable in polynomial time, and then try find a suitably reduction from a known NP-complete problem to our problem at hand.

Lastly, if time permits, we can change the rules of our system to be more true to the actual game. For example we could consider continuous sweeps, piece drop speed simulation and online versions of the game. We will then try to prove our previous findings in this new system, which will hopefully tell us more about the original game.

# 4 References

So far we have identified three papers which we consider relevant for our project. One these is a description of Lumines strategies [1], the remaining two regards computational complexity in Tetris [2] and Match-three puzzle games [3]. The Tetris paper is especially of interest. The paper describes a good method for how to approach reasoning about computational complexity in a video game context, such as formalizing the rules of the game in a mathematical system. The fact that Tetris and Lumines share some characteristics, but are not fully alike, makes it possible for us to be inspired by the paper, but not to the point of plagiarism.

# 5 Time plan



*Time plan as a Gantt chart*

# References

[1] Greg Aloupis et al. “LUMINES strategies”. In: *Computers and Games* (2007), pp. 190–199.

[2] Erik D. Demaine, Susan Hohenberger, and David Liben-Nowell. “Tetris is Hard, Even to Approximate”. In: *CoRR* cs.CC/0210020 (2002). URL: `http://arxiv.org/abs/cs.CC/0210020`.

[3] Luciano Gualà, Stefano Leucci, and Emanuele Natale. “Bejeweled, Candy Crush and other Match-Three Games are (NP-)Hard”. In: *CoRR* abs/1403.5830 (2014). URL: `http://arxiv.org/abs/1403.5830`.