Synopsis of *Making Mario Work Hard*

Aaron Ceross (ac14580)

1 Aims and Objectives

An NP-complete problem is one with a solution that can be *verified* quickly in polynomial time, though an existing solution is not readily known or *identifiable*. It has been proven that certain video games, such as *Super Mario Bros*, are computationally hard (i.e. NP-complete) in that the character could be made to solve an arbitrary instance of a Boolean satisfiability problem (SAT). SAT is a type of NP-complete problem in which the variables of the formula may be consistently replaced by the values TRUE or FALSE in a way that the formula evaluates to TRUE, or 'satisfiable'.

The aim of this project is to investigate the feasibility of the video game medium as a teaching tool to visualise and demonstrate concepts in computational complexity through the evaluation of different instances of SAT. The research is successful when it meets its objectives, which are linked to the schedule of deliverables. These objectives are to:

1. Develop a platform game which is NP-complete; (Met by Deliverables 1 and 2)
2. Develop a level generator which converts a SAT into a playable level; (Met by Deliverable 2)
3. Develop an intelligent agent (IA) which can play these levels by choosing a suitable path using a SAT solver; (Met by Deliverable 2)
4. Evaluate level-generation and IA for efficiency and success against an established criteria or one developed during the project; (Met by Deliverable 3)

2 Deliverables

- **D.1 Foundational Work**, Weeks 17 — 21 (23 February — 29 March 2015):
  - This includes items such as: ***D.1.1 Research Plan***, ***D.1.2 Literature Review***, ***D.1.3 Identification of relevant algorithms and methodologies***, and ***D.1.4 Creation of success criteria for level-generator and IA***. While the dates are specified, the development of these deliverables will be on-going through the life-cycle of the project.
- **D.2 Platform Game Development**, EV1 — SV6 (30 March — 19 July 2015):
  - ***D.2.1 Platform Game***: This will be designed and written in C++, using the Simple DirectMedia Layer development library to manage audio, movement input, and graphics.
  - ***D.2.2 Level Generator***: This creates multiple playable levels, which allow for inputs from both the SAT problem solver and the human user. The level generator may need to be scalable to take into account complex levels that are beyond what a human player would play. This requires adapting the display engine to zoom out and display the graphics in a more simple manner.
  - ***D.2.3 IA***: The function to automatically play-through the generated level. This may include a speed-up/down option in order to reasonably watch a large complex level. The extended version of this will allow a user to pause a play-through and take over from the IA. There may be scope also to compare different SAT solvers and visualise this through the game (see D.3.3).
  - ***D.2.4 Development Tests***: These include (a) testing the game mechanics of the platform game and ensuring NP-completeness; (b) scalability testing of level-generation algorithm(s); (c) testing of IA mechanics; and (c) user-acceptance testing (e.g. qualitative feedback from students and lecturers).
- **D.3 Data Collection, Analysis and Evaluation**, SV7 — SV15 (20 July 2015 — 21 September 2015)
  - ***D.3.1*** Evaluation report on whether the level generator successfully creates an NP-complete level based on the SAT;
  - ***D.3.2*** Evaluation report of the IA's ability to solve the generated level across time and movement efficiency;
  - ***D.3.3*** Comparison report on using different SAT solvers simultaneously in the game environment. Each solver would be uniquely identifiable (e.g. by colour) so the user could see the individual problem-solving paths.

3 Added Value

Computational complexity is difficult yet fundamental subject for students of computer science, mathematics, and related disciplines. The main challenge of the project is to transform a complex mathematical theory or concept into a gamified visualisation. If successful, this could allow for these concepts to be presented in a novel, complementary manner to conventional formulae and diagrams, which may be more approachable and understandable for non-experts and experts alike. Video games allow a user to interact with the concepts in a more practical, hands-on, and entertaining dimension, which could provide an effective starting point for presenting computational complexity to a wider audience. Future work may include evaluating the research deliverables's effectiveness as a teaching tool.

Furthermore, this research could have impact on game design theory and automated content generation in the video game industry. Depending on the results of the research, the level-generation algorithm could be further developed into a more refined tool to generate more complex and challenging video games.

I have previously created a video game in C for the Software Engineering module, which will help in this project's development.