Intelligent Game Designer

Project Plan

Team Juliet

January 26, 2016

1 Introduction

During this project, each of the six members of our team should be allocated a portion of the work that will take approximately 40 hours (10 per week). Below the work is broken down into its different components and subcomponents, each of which is listed with the person/people who will be primarily responsible for it. Each part is given an estimate of the time it will take. The majority of the project involves writing code, and all code should be tested (including unit tests), so this will be included in the time estimates.

2 Game Implementation

Team members: Alastair, Artem, Ben, Ciaran, Devan, Dimitris

Approximated time cost: 30-35 hours
Given that all other components of the project depend on the implementation of the game for which we are designing levels, every member of the group will contribute to this component. At the highest level of abstraction, this boils down to implementing the GameState and DisplayedGameState classes mentioned in the specification, and their corresponding makeMove() methods. Of course, this will also involve a plethora of methods for analysing the board for matches, special candies, and so on.

3 Simulated Players

Team members: Artem, Dimitris

3.1 Designing Basic Simulated Players

Approximated time cost: 4 hours (1-2 hours per player type).
This includes implementing low levels of simulated players based on the strategies discussed in the Specification.

3.2 Designing advanced players

Approximated time cost: 9 hours (3 hours per player type).
Improving on the basic players, coding and refining search algorithms.

3.3 Experimenting on more elaborate concepts

Approximated time cost: 2-3 per concept explored
Combining different strategies such that best players can get maximum score while also completing the objective of the level. This includes creating metric functions to measure the difficulty of removing a jelly or to measure the probability of completing a game by making that move.

3.4 Testing and Assessing Simulated Players

Approximated time cost: 8-10 hours
There will be numerous tests written to check that the Simulated Players behave the way they were designed to work. Assessment for the quality of the players and assignment of class will be possible by analysing the games played.

4 Level Design

Team members: Alastair, Ciaran, Devan

4.1 The Genetic Algorithm

Approximated time cost: 25-30 hours
The core 'learning' process of the level design will involve implementing a genetic algorithm. This should maintain a population of level designs, and through the use of mutation and crossover (which are in turn based on the results returned by the simulated players, aesthetic checker and constraints checker), should converge the generations of solutions to satisfy the criteria set out by the user. Ciaran will be focused on implementing this, with the help of Devan and Alastair.

4.2 Level Designer Manager

Approximated time cost: 25-30 hours
Running a large number of simulated players on various level designs and interpreting the results returned involves some form of multithreaded management which should be encapsulated within this component of the level designer. Devan will be focused on implementing this, with the help of Ciaran and Alastair.

4.3 Aesthetic Checker

Approximated time cost: 6-9 hours
The genetic algorithm will rely on some form of evaluation of how aesthetically pleasing a given level is. This will include measuring aspects such as how symmetric it looks, the level of connectivity of jelly-blocks, and so on. Alastair will be focused on implementing this.

4.4 Constraints Checker

Approximated time cost: 2-3 hours
Before level designs can even be considered as a solution, they will need to pass a set of constraints that ensure the design is actually a feasible solution. Alastair will be focused on implementing this.

5 User Interface

Team members: Ben

5.1 Game Interface

Approximated time cost: 5-15 hours, depending on level of detail used: 4 for basic setup, 3 for additional texture design/implementation, 3 for moving tiles, 3 for effects (explosions, sound effects etc.)
When a game of Candy Crush is being played be a human player, it will need to be displayed on the screen, in a visually engaging manner. One part of the overall project will involve creating this display and adding features to make it more aesthetically pleasing, such as animations, visual effects and graphics. (This interface will be used both for human players and for when we wish to watch simulated players play).

5.2 Menu Navigation

Approximated time cost: 3-5 hours
Due to the number of features that could be implemented in such a project, a visual interface will be needed to navigate through and select which of the features to view. This will involve designing the pages and traversing between them.

5.3 Level Creator

Approximated time cost: 2-5 hours
An optional addition to the overall project, a level designer could be useful to both present to others, and to test our simulated players on. This will involve making an interface with which a human user can create Candy Crush levels to be saved/exported in our agreed format.

5.4 Level Load/Store

Approximated time cost: 3-4 hours
For demonstration and testing purposes, time allowing, we wish to store a list of previously made levels, which could be played by humans or simulated players. This interface will include detecting what levels have been stored, and an interface to navigate and select between them.

5.5 Level Requesting

Approximated time cost: 3-5 hours
Used as a front end for the level designer, this interface will allow a human to select particular criteria for the Level Designer to work towards. While the Level Designer is working, the two will communicate so that the screen can display some form of information regarding the progress of the designer, such as the range of predicted difficulty levels generated, and the current best designed level.