Software Engineering Large Practical 2016-17

GR**A**BBLE
Proposal document

Designed and planned by
Radoslav Kirilchev – s1452923

Table of Contents

1. Summary 3
2. Functional Requirements 4
    1. Compulsory (core) features 4
    2. Setting and narrative 4
    3. Player accounts, affiliations, and factions 5
    4. The All-Dictionary, and collecting letters 5
    5. Ash 6
    6. Progression 7
    7. Hypothetical features 7
3. Non-Functional Requirements 8
    1. VCS: Git 8
    2. Developer's Android: 4.4.4 8
    3. Online components: C#/Pending 8
    4. Temporal and power constraints 8

1. Summary

The official GR**A**BBLE logo uses the Adobe Caslon Pro typeface (license pending), with the "A" enveloped in a square border, vaguely resembling a Scrabble tile.

GR**A**BBLE (to be depicted with a bold "A") is the name of the Android game for this course, developed solely by me.

This document shall serve as a proposal for the project, as well as a summary of all the features and considerations we strive to ship with the finished product.

These functional and non-functional requirements shall be separated into categories in the document. Refer to the table of contents on pg. 2 (right above) for particular sections; and to the descriptions and summaries of each particular feature.

Note that this isn't a complete SRS document. It will instead employ a broader stroke in the description of all features and requirements; exact details shall be left to the design document that will appear later.

Concept created by Radoslav Kirilchev
Zero Vector Studios, 2016
www.zerovector.eu

2. Functional Requirements

Functional requirements are all the "technical" features of the system. We shall present them in order, from the compulsory ones detailed in the coursework description, through a more detailed list of all bonus features that are being considered, to a list of features that are present only as ideas and are unlikely to make an appearance.

1. **Compulsory (core) features**
    1. Letters are distributed around the central campus of the University of Edinburgh, according to a KML map for each day of the week.
    2. Collected letters shall be formed into seven-letter words, which shall be found in the *"Official Grabble Dictionary 2016"*.
    3. Letters have score values based on their rarity. Collected words shall be assigned scores totalling the sum of the scores of all letters in the word. *(See section 2.5 for more details.)*
    4. Individual letter instances can be collected by individual players once per day.
    5. Additional details can be found in the coursework description.
2. **Setting and narrative**
    1. For this game, I've drawn strong inspiration from Roger Zelazny's novel *"A Night in the Lonesome October"*, which is in turn inspired by the works of many famous authors, including H.P. Lovecraft, Edgar Allan Poe, and Sir Arthur Conan Doyle.
    2. Players of the game take the roles of members of secretive organisations, vying to complete an ancient ritual, and either destroy the world, or preserve it. This is determined by their *affiliation*: the Openers seek to open a rift to the Far Realm (a place beyond human understanding) and unleash eldritch abominations upon the Earth; whereas the Closers, who see the current world as being in adequate equilibrium, wish to prevent this from happening. *(See section 2.3)*
    3. This ultimate goal is achieved by completing an ancient artefact, known as "the All-Dictionary". Collected words are added to the All-Dictionary *(see section 2.4),* and the first player (or *faction* of players) to complete ALL words in the dictionary wins the game for their *affiliation*. *(See section 2.3)*
    4. Scores granted by words and letters serve more purposes than just ranking players. Instead, these scores are represented by *Ash*, which acts as a currency in the game. *(See section 2.5)*
    5. Players may pick a word they're collecting at the moment. Letters that don't fit the word, or are not needed at the moment, are stored as a limited *reserve*. Excess letters may be converted directly into *Ash*; *Ash* may be used to create new letters. *(See section 2.5)*

3. **Player accounts, affiliations, and factions**
    1. Players need an account to participate in the game. An account needs to hold the following information:
        a. E-mail address;
        b. Password;
        c. Username;
        d. Faction name;
        e. Affiliation.
    2. Upon registering (creating an account), players shall select their *affiliation*. The two choices are:
        a. Openers; and
        b. Closers.
    3. Upon registering, players shall also select their *faction name*. In the context of the game, a *faction* is a group of players who share game progress.
        a. A player can invite other players to join his or her faction, and can in turn be invited to join other factions. An invited player may decide whether or not to accept or decline the invitation.
        b. All players in the faction must of the same *affiliation*.
        c. Only the faction creator (the player who entered said faction name upon registering) may kick other players from his or her faction.
        d. Upon joining a faction, all words collected in the player's dictionary are added to the faction's joint dictionary. As long as the player is a member of the faction, the player collects words for the faction's joint dictionary.
        e. A player kicked from a faction returns to his or her personal dictionary which is the same as it were before joining the faction.
4. **The All-Dictionary, and collecting letters**

    The "All-Dictionary", in essence, is the set of all words in the "Official Grabble dictionary 2016". Players aim to complete this collection, thereby winning the game.

    1. Except "All-Dictionary", there shall be one alternative name per *affiliation*:
        a. *Codex Maleficarum*, for Openers; and
        b. *Sacred Tome*, for Closers.
    2. The All-Dictionary contains some 23869 entries, as listed in the coursework description. The game is won when a player (or a *faction*) possesses a "full" dictionary (containing all 23869 words).
    3. Winning the game is a global event; all players of the winning *faction*'s *affiliation* win the game; all other players lose it.
    4. Any prizes awarded to winning players are to be defined at the discretion of the developers.
    5. To complete their All-Dictionary, players need to collect letters and form words.

6. At any point in time, players may work towards only one word. To complete the word, players need to collect the letters in the word.
7. Letters are collected by approaching them whilst carrying the mobile device the game is running on.
8. The range at which a letter is considered "collected" is ten (10) metres (may be changed after testing, depending on results and feedback).
   a. Should said letter not be present in the current word; or should its place have already been filled with a collected letter, the letter is added to the player's *reserve*.
   b. Should the player's reserve be full for this particular letter, the letter may not be collected, or may be converted directly into Ash instead. *(See section 2.5)*
9. Each letter has an *Ash* value, and players may hold up to five (5) instances of each letter in their *reserve*. This may be subject to improvement. *(See sections 2.5, 2.6)*

5. Ash

1. Letters have different Ash values, as detailed in the coursework description:

A	B	C	D	E	F	G	H	I	J	K	L	M
3	20	13	10	1	15	18	9	5	25	22	11	14
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6	4	19	24	8	7	2	12	21	17	23	16	26

(Table 2.5.1 – Ash values per letter. This amount of Ash is needed to create a letter anew.)

2. A collected letter may be converted into Ash at the *Crematorium* (a menu where the conversion takes place). Converting a letter to Ash ("burning") is made at an inefficient rate to incentivise actual letter-collection, as per the following table.

A	B	C	D	E	F	G	H	I	J	K	L	M
1	4	2	2	0.2	3	3	2	1	5	4	2	2
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	1	4	4	1	1	0.4	2	4	3	4	3	5

(Table 2.5.2 – Letter "sell" prices. This amount of Ash is gained upon burning a letter.)

   Note that Ash is always an integer value, meaning that in the case of E and T, a player may convert five E's at once and gain 1 Ash, or 5 T's and gain 2 Ash. The usual rate is a little worse than 5-to-1, with the exception of the letter A, which is 3-to-1, and a few others, which, due to rounding, are closer to 4-to-1.
3. Accumulated Ash may be used at the player's discretion to create new letters at the *Ashery* (a menu where this conversion takes place). To create a letter, a player needs to pay its full value in Ash, as per table 2.5.1. This allows players, when in a dire need, to complete their collections.
   a. Letters may not be created if the current word will not accept them, and/or if the player's *reserve* for this letter is full.

4. In addition to everything else, every six (6) successfully collected letters grant the player 1 Ash, as a means of having some way to generate this resource. The exact amount of letters needed to generate 1 Ash may be changed depending on testing and feedback.
5. Completed words do not grant the player any additional Ash, but do contribute to the player's *rank* with *experience*. *(See section 2.6)*

6. Progression

1. Every player shall have a *rank* in addition to all of their other stats, as a means of facilitating progression. All players, upon registering, start the game at rank zero (0).
2. To increase their rank, players collect *experience*. Every point of Ash collected grants a point of experience to the player; a completed word grants experience points equal to the sum of the Ash values of each letter. *(See table 2.5.1)*
   a. Whilst in a faction, a word completed by a faction member grants no experience to other faction members.
3. Rank increases shall require arbitrary amounts of experience gained. These amounts shall be determined after preliminary testing.
4. Aside from getting the player closer to the final goal of the game, completing the All-Dictionary, ranks shall provide benefits to players. These benefits shall be determined during the development process, but may include increasing the capacity of the player's reserve, or getting better letter-to-Ash conversion rates at the Crematorium.

7. Hypothetical features

These features are listed more in the quality of ideas rather than something to be implemented. Whether or not they see the light of day shall depend on the specifics of the development process.

1. *Player classes:* Upon registering, a player may select one of several pre-defined classes that provide a unique benefit or change to the default play-style. For an example, an "Oracle" player may get a gradually expanding letter-detection range whilst standing still; an "Alchemist" may get to collect twice as many letters for ten minutes per day, and so on.
2. *Player-versus-Player (PvP) aspects:* The presence of factions and affiliations naturally suggests a competitive background for the game. Having factions (from opposing affiliations) declare war on each other, and having players "fight" each other in some way (forming words such as "STRIKE", "ATTACK", "PARRY", "DEFEND", and so on out of their letter collections) whilst in close proximity to each other, and having the winner in the fight gain some of the Ash of the loser, would create a curious PvP side to the whole ordeal. It is, however, most likely too difficult to implement in a short timespan for a University project. For a real game, yes.

3. *Leaderboards, daily quests, timed events:* Leaderboards, displaying the faction that currently has most words accumulated, or the players who've collected most words for the given day, should be easy enough to include, but are listed here simply because they're not a priority. Daily quests, where players get different tasks every day ("collect 10 Z's", "complete 3 words", and so on, with Ash and experience as rewards), or timed events ("The first faction to collect 20 A's in five minutes wins") could also be valid features, but I doubt I'll get to implementing them.

3. Non-Functional Requirements

Non-functional requirements are any other considerations that might be given thought to.

1. **VCS: Git**
   I'll admit to not having a lot of experience with VC systems, and what experience I have, it's been with Mercurial. However, as per the requirements of this coursework, I shall utilise a local git repo. (A good way to get some experience with it, too, I guess.)
2. **Developer's Android: 4.4.4**
   My phone runs Android 4.4.4, so this is the minimum version I'll be targeting. Testing shall be made using emulated Android v4.4 or later versions.
3. **Online components: C#/Pending**
   The functional requirements specify a less-than-trivial (and less-than-very-complicated) need for an online component. That shall most likely use relational SQL databases, and shall be written either in C# or in something else. The details are pending the clarification of the needs of the project.
4. **Temporal and power constraints**
   As the development of the project progresses, I'll look into which features need the most power, or hinder performance the most. Such features might need toning down, or to be completely removed.
   On the same topic, any feature that might end up taking more time to implement than expected might get "the axe" and be cut from the final product. Nothing can be set in stone in an agile environment.