

UNIVERSITY OF ST ANDREWS

CS4099: MAJOR SOFTWARE PROJECT



A Tactical RPG Engine

Bilal Syed HUSSAIN

Supervisor:

Dr. Ian MIGUEL

Second Marker:

Dr. Alex VOSS

March 25, 2012

Contents

Abstract	4
Declaration	4
1 Introduction	5
1.1 Project Baseline	5
1.2 Project Success	5
2 Context Survey	7
2.1 Evolution of Tactical RPGs	7
2.2 Overview of Game Engines	7
3 Requirements Specification	8
3.1 Project Scope	8
3.2 Requirements Overview	8
3.3 Overview Description	8
3.3.1 Product Perspective	8
3.3.2 Product functions	8
3.3.3 User characteristics	8
3.3.4 Constraints, assumptions and dependencies	8
3.4 Specific Requirements	8
3.4.1 Security Requirements	8
3.4.2 User Interface Requirements	8
3.5 Technology Architecture	8
4 Software Engineering Process	9
4.1 Methodologies Used	9
4.1.1 Evolutionary Prototyping	9
4.1.2 Test Driven Development	9
4.2 Version Control	9
4.3 Standardised Formats	9
5 Ethical Considerations	10
6 Design	11
6.1 Engine	11

6.2	Game Progression	11
6.3	Units	11
6.4	Editor	11
6.4.1	Exporting	11
6.5	View	13
6.5.1	Tilemap	13
7	Implementation	15
7.1	Engine Development and Testing	15
7.1.1	Maps	15
7.1.2	Units	15
7.1.3	Events	15
7.1.4	Algorithms	15
7.2	View Development and Testing	15
7.2.1	Map Rendering	15
7.3	User Interface	15
7.4	Data Formats	15
7.4.1	XML	15
7.4.2	Custom Classes	15
7.5	Editor Development and Testing	15
7.5.1	Overview	15
7.5.2	Map Editor	15
7.5.3	Unit Editor	15
7.5.4	Event Editing	15
7.5.5	Exporting	15
8	Scripting	16
8.1	Language Choice	16
8.2	Data Exposed	16
8.3	Action	17
8.4	Winning Conditions	17
8.5	Unit Events	17
8.6	Tiles Events	18
8.7	AI Events	18
9	Evaluation and Critical Appraisal	19
9.1	Results of User Testing	19

<i>Contents</i>	Bilal Syed Hussain	3
10 Conclusions		20
11 Testing		21
11.1 Feedback		21
A User Manual		22
B Questionnaire		23
B.1 Editor Usability Scale		25
B.2 Playing a pre-created game		25
B.3 Questions		26
C Future Work		27

Abstract

In odio velit, semper quis mattis eu, varius et felis. Donec vulputate aliquam purus id feugiat. Fusce vel ante neque, vitae placerat sem. Nam a tortor purus. Aenean laoreet volutpat consectetur. Proin sit amet lorem orci. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi quis tempus lacus.

Donec feugiat ultrices porta. Vivamus laoreet odio sed augue ultrices vitae consequat nibh pharetra. Nam et fringilla est. Sed dolor lorem, luctus aliquet lacinia vitae, mollis vel tortor. Vestibulum aliquam mi eget neque semper aliquam. Duis accumsan sapien tristique tellus fringilla convallis. Nulla odio augue, eleifend sit.

Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is NN,NNN* long, including project specification and plan. words

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the Declaration University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

1 Introduction

An RPG (Role Playing Game) is a game where a player assumes the role of a character. An RPG is usually story driven and the character usually has a quest to complete. In the course of the game the player will go to different environments such as town and dungeons. In these environments the player will have to fight opponents in battles. Combat in RPGs is normally a simple turn based system where players and their opponents take turns to attack each other using various skills.

A Tactical RPG is a sub-genre of an RPG that focuses on the combat side of the genre. A Tactical RPG is series of battles, which take place in various environments intertwined with an over-arching story.

Each battle is grid based (like chess) where each player has a number of units(pieces). The players take turns to move their units. Each unit has attributes associated with it such as



Figure 1: **Tactics Ogre**[?] a classic Tactical RPG

strength, and hit points that affect all the actions in the game. Like chess there are different kinds of units which affects how the unit moves and what action they can perform. A unit can attack other player's units, the goal of the battle is usually to defeat all the opponents units.

The aim of this project is to create an engine which will take resources such as graphics, sounds and rules of the game to create a runnable Tactical RPG.

1.1 Project Baseline

No previous work was used for this project. All of the project was created during the course of the academic year.

1.2 Project Success

- Finished all the all primary objectives, and nearly all secondary and Tertiary objectives.

- Very useable
- Cross platform

2 Context Survey

2.1 Evolution of Tactical RPGs

Notable TRPGs

- Bokosuka Wars - probably the first TRPG
- Fire Emblem: Ankoku Ryu to Hikari no Tsurugi - First popular TRPG. Characters are unique
- Tactical Ogre:
 - First TRPG with isometric graphics.
 - Character battle order is determined by the character's 'speed' rather, each player moves all their units when its their turn.
 - First to have a branching plot and the player's choice effecting the game.
 - Associated the genre with the word 'Tactics', used by many later games
- Final Fantasy Tactics, widely popular, based on Tactics Ogre.
- Disgaea: Hour of Darkness: Allows the player to play random generated maps. The latest in the series is one of the few TRPGs that contain a map editor.
- Recent game, have mostly mix aspects from other genres, for example Valkyria Chronicles features FPS like shoting when attacking.

2.2 Overview of Game Engines

- Sim RPG Maker 95, one of the few tactical RPG's engines
- RPG Maker which it is based off.
- Mention engines such unity which used to make TRPGs?

3 Requirements Specification

3.1 Project Scope

The aim of the project is to allow the user a highly customisable Tactical RPG. There are three main parts to the project the *engine*, the *GUI* and the *editor*.

The engine will contains all the logic of the game including the progression as well as the battle system. The GUI, will be an isometric view of the game (see Section 6.5.1).

The editor will allows the user to customise the input to engine. This includes visual map making as well as specify all the attributes of the units and weapons. The editor also allows the user to export the game as a standalone application.

3.2 Requirements Overview

A complete listing of Requirements is in section ???. The main requirements are to create a engine allows a high degree of customisability, a isometric view and exporting the game as a standalone application.

3.3 Overview Description

3.3.1 Product Perspective

3.3.2 Product functions

3.3.3 User characteristics

3.3.4 Constraints, assumptions and dependencies

3.4 Specific Requirements

3.4.1 Security Requirements

3.4.2 User Interface Requirements

3.5 Technology Architecture

4 Software Engineering Process

4.1 Methodologies Used

4.1.1 Evolutionary Prototyping

4.1.2 Test Driven Development

Test Driven Development (TDD), has four main phases

- Write a Test.
- Write the code to pass the test.
- Run all tests.
- Refractor

The major benefits of TDD are the system will be well tested. It also has the added benefit of allowing prevents new features from breaking the system. Combined with version control as discussed in the next section, it make very easy to find since the unit tests can be used to find *when* the code broke as well as *which* piece of code was the root cause.

TDD was used when designing and implementing the model's algorithms such unit movement.

Although TDD has many benefits it can effently used to test user interfaces, since a unit test can not simulate how a user would react to different aspects of the user interface. Resulting from this I did user studies as well usability studies to defects in the user interface.

4.2 Version Control

4.3 Standardised Formats

5 Ethical Considerations

- Collection of data from questionnaire.
 - Just result of questionnaire, no personal data.
- Asking users to create a game.
- Asking users to play the created game.

6 Design

6.1 Engine

6.2 Game Progression

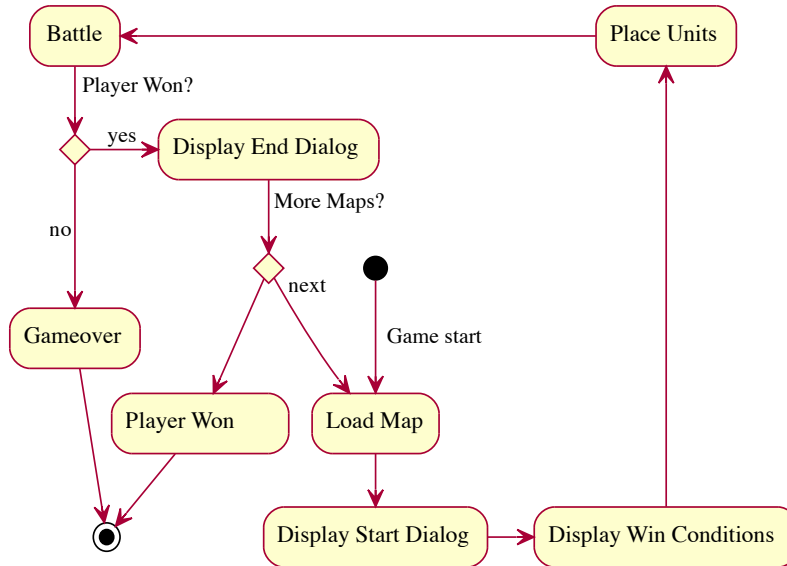


Figure 2: Activity diagram which a high level of overview of the engine

Figure 2 shows the overview of how the created game progress. Each game has a number of maps where a battle take place. After the map is loaded any relanent dialog is display, along with winning conditions. The player's units are then placed are the map¹. And the battle (which will be discussed below takes). If the player's loses the battle, a gameover screen is shown and the game end. In contrast if the player wins he/she advances to the next map, if there one.

6.3 Units

6.4 Editor

6.4.1 Exporting

The editor can export a project as a complete package, either as a Mac OS X application or as jar. These application don't requires any external resources, apart from a recent version of java².

A notable feature of the editor is that jar will work on any java enabled platform, since the jar contains all required libraries for each platform. The OS X application can even be export on other platforms.

While most of the testing was done on OS X³, it also works well on Linux⁴. It even has

¹While the engine support allowing the user's to choose where the their units are placed, the GUI does not due to time constraints. The editor does support specify the starting location for the player's units

²specifically Java 1.6+

³Mac OS X 10.6 Snow leopard

⁴Science Linux x.y

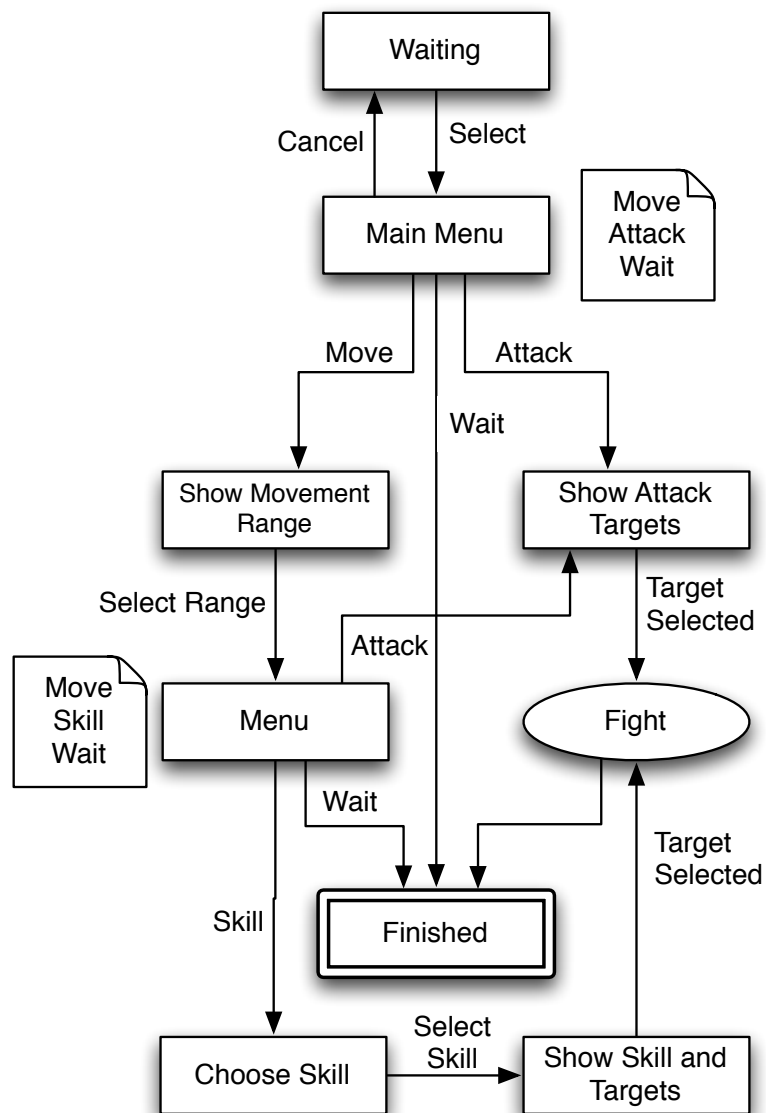


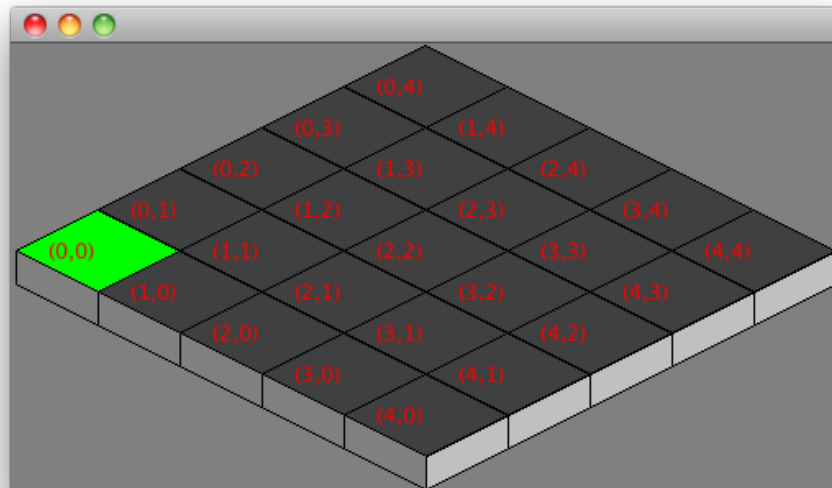
Figure 3: The State diagram of a single turn of a player's unit

limited compatibly with Windows⁵ (apart from some minor graphics issues).

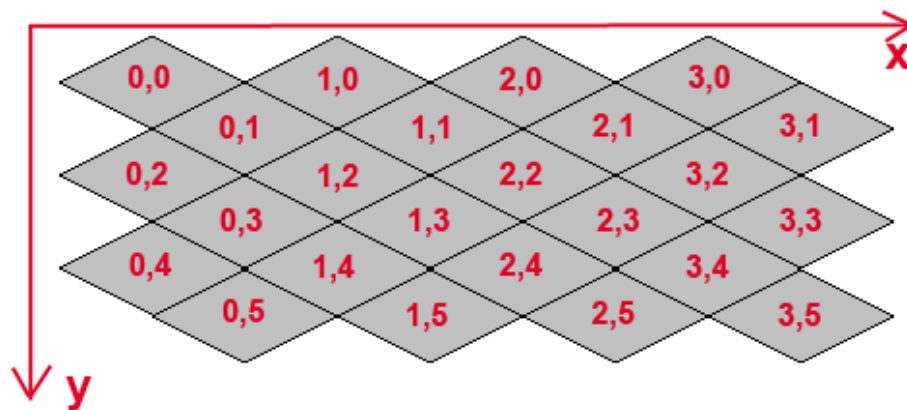
6.5 View

6.5.1 Tilemap

There were two main choices for the isometric tilemap, a ‘Diamond’ map or a ‘Staggered’ map [?], examples of both are shown below.



(a) Diamond Map



(b) Staggered Map

Figure 4: The two main types of isometric tilemaps

The ‘Staggered’ Map has following advantages:

- The map fill up the screen with very little wasted space, so the user can more of what happing on the map.

The ‘Diamond’ map was chosen for the following reasons:

⁵Tested on Windows 7 32 bit

- ‘Diamond’ map look nicer then ‘Staggered’ maps because it has no ragged edges.
- Since that maps are large (at lest 15×15) the space wasted at the edges of the map does not matter as much.
- Simpler to think about, since a ‘Diamond’ map is just a rectangular map rotated.

Maths about isometric tilemaps?

7 Implementation

7.1 Engine Development and Testing

7.1.1 Maps

7.1.2 Units

7.1.3 Events

7.1.4 Algorithms

7.2 View Development and Testing

7.2.1 Map Rendering

7.3 User Interface

7.4 Data Formats

7.4.1 XML

7.4.2 Custom Classes

7.5 Editor Development and Testing

7.5.1 Overview

7.5.2 Map Editor

7.5.3 Unit Editor

7.5.4 Event Editing

7.5.5 Exporting

8 Scripting

Scripting allows the user to customise aspects of the game. This includes customising the opponent's AI, custom winning conditions and user defined events.

8.1 Language Choice

There were three main choices using Javascript, using JRuby⁶, or building a 'domain specific language'.

Creating a 'domain specific language' was considered initially, this would have the following advantages:

- Provides more abstraction, and allow the complex details to be hidden.
- Easier to validate since the languages contains a very few constructs.

but was rejected because:

- of the time to create and test the new language.
- of the cost of creating tools for the new language, there are already source code highlighters and debuggers for Javascript and Ruby.
- of the loss of efficiency, the Javascript parser in the JDK as well as JRuby is very efficient and provides advance features such as 'just in time compilation' ⁷ which would not be possible to implement for the new language within the time constraint of the project.

JRuby has the following advantage:

- Easier syntax for interacting with Java then javascript.
- Easy to use with the embedding API in the JDK.

Javascript was chosen over Ruby as a scripting language for the following reasons:

- Javascript embedding is build into the JDK, so the user does not have install anything extra. It also has the advantage of being cross platform.
- Javascript is easy to learn, and average user is more likely to have used it before as compared to Ruby.

8.2 Data Exposed

Events can be attached to units, tiles in a battle, globally in a battle and to the AI. All events are passed a mapinfo object which contains the following as read only data:

- A hashtable of the players unit and a hashtable of the enemies units. For each unit this includes

⁶A Ruby implementation written in java

⁷A method to improve the runtime performance, by translating the interpreted code into lower level form, while the code is be run

- all the unit's attributes such as the location, and hit points.
- if the unit has been defeated.
- The leader unit of each side if there is one.
- The number of turns taken.

The `mapinfo` object contains the following methods:

`win` The player wins the battle.

`lose` The player loses the battle.

`dialog` The player is shown the specified dialog (to show the user some the plot). Can be directed from a specify unit, or a global message.

`action` Executes the specified action.

This allows the user to make complex events without them changing the model to much.

8.3 Action

A `action` is a set of unit defined actions. For example a poison action could reduces the a units 'hit points' by 10%

8.4 Winning Conditions

The user can specify the winning conditions based on what occurring in the battle, examples include

- If opponent's leader's hp < 50% then `win()`.
- If <character> dies then `lose()`.
- If number of turns > 20 then `lose()`

8.5 Unit Events

Unit events get passed the specified unit as well as the `mapinfo`. the event can be specified to execute when:

1. The unit finishes its turn.
2. The unit is affected by magic.
3. The unit is attacked.
4. The unit attacks.

Example: When <unit> attacked counter attack.

8.6 Tiles Events

Tiles get passes the specified tile as well as the Unit. The event can be specified to execute when

- A unit moves to a tile.
- A unit moves though a tile.

Example: On unit moving though `action(posion)`

8.7 AI Events

The behaviour of AI can be customised, with commands such as:

- Attack the player's unit with highest/lowest hp.
- Attack the player's leader unit.
- If player's leader's hp < 20% `heal(leader)`.
- Attack player's characters of class `<class>`.

The AI events `mapinfo` has addition methods including:

`attack` Attack the specified unit.

`follow` Move as close as possible to the specified unit.

`heal` Heal the specified unit.

`move` Move to the specified location.

`wait` Do nothing this turn

The commands themselves can be conditional, as example

Listing 1: Conditional AI Event

```
If opponent's leader's hp < 20% then
    heal(leader).
else If player has a leader unit then
    If player's leader's hp < 20% then
        Attack the player's leader unit
    else
        Attack the player's closet unit with the lowest hp.
    end
else
    wait
end
```

9 Evaluation and Critical Appraisal

9.1 Results of User Testing

System usability scale (SUS) was used [?]. This works by giving even numbered questions a score of $(5 - value)$ and odd numbered questions a score of $(value-1)$. Questions that contributed a high score show that the system is usable.

10 Conclusions

11 Testing

11.1 Feedback

- It was hard to see which unit was selected. This was fixed by displaying 'Current' in selected unit's info. The info window of the selected unit was also lightened to make it more obvious.
- It was hard to see which are my units. This was fixed by displaying the player's unit's info in green and the enemy's unit's info in red.
- Some users could not figure out the key bindings of the game. This was fixed by displaying a list of all key binding at the start of the game.

A **User Manual**

B Questionnaire

Task

The task involves creating a single level of a Tactical RPG (Each level is grid based (like chess) where each player takes turns to move and/or attack the opposing player).

Weapons


Name	Weapon Type	Strength	Icon
Long Bow	Ranged	30	
Black Spear	Spear	20	
Ice Sword	Melee	10	

Skills


Name	Type	Range	Area	Strength
Air Blade	Ranged	2	0	25
Thunder Flare	Ranged	4	1	15


Units


Agrias		
	Weapon	Long Bow
	Strength	20
	Move	3
	Skills	
	Air Blade	


Elena		
	Weapon	Black Spear
	Strength	30
	Move	5
	Skills	
	Thunder Flare	

Map Enemies

Mustadio		
	Weapon	Long Bow
	Strength	20
	Move	3
	Skills	

Druksmald		
	Weapon	Ice Sword
	Strength	30
	Move	5
	Skills	

Zalbaag		
	Weapon	Ice Sword
	Strength	25
	Move	5
	Skills	

Ajora		
	Weapon	Ice Sword
	Strength	20
	Move	5
	Skills	

Map

Figure 5: The map to create

Win Condition

Defeat Specific Unit – Elena.

Start Dialog:

Text You can not Win!

Speaker Kyou

End Dialog:

Text How did I lose?

Speaker Elena

Music:

Background Music 3-15 Faraway Heights

B.1 Editor Usability Scale

© Digital Equipment Corporation, 1986.

1. I think that I would like to use this system frequently.

← strongly disagree agree completely →

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

2. I found the system unnecessarily complex.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

3. I thought the system was easy to use.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

4. I think that I would need the support of a technical person to be able to use this system.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

5. I found the various functions in this system were well integrated.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

6. I thought there was too much inconsistency in this system

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

7. I would imagine that most people would learn to use this system very quickly

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

8. I found the system very cumbersome to use

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

9. I felt very confident using the system

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

10. I needed to learn a lot of things before I could get going with this system

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

B.2 Playing a pre-created game

1. I found the game intuitive

← strongly disagree agree completely →

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

2. The game had a appropriate level of difficulty.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

3. I enjoyed playing the game.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

4. Please share any comments about the game :

B.3 Questions

5. Have you played a Tactical RPG before?

6. Did Engine have features you like to create in a game?

7. How easy to use was the Engine?

8. What particular aspects of the Engine did you like?

9. What particular aspect of the Engine did you dislike?

10. What features would you like to see added to the Engine in the future?

11. Please share any other comments:

C Future Work

- Improvement to levelling up. Usually a unit does not have access to all of its skill at begining, but gains access to them when levelling up. This would make the produced game more balanced, since only skill appropriate to the unit stats could be used.
- Implementation of an overworld map with a battle happening at each location. This would allow the user to choose which map to play. A good use of this would be a branching storyline where the plot is changed depending on which maps the player plays.
- Better Ai
- Scripted Events