Version: 1  
  
Date 22/3-16

Authors: Jonatan Gustafsson, Robert Palm, Gustav Grännsjö, Johan Blomberg  
  
This version overrides all previous versions.  
  
1 Introduction  
   
1.1 Purpose of application

The aim of this project is to realize a strategy game with gameplay inspired by games like “[Duke](https://en.wikipedia.org/wiki/The_Duke_%28board_game%29)” and “[Fire Emblem](http://fireemblem.wikia.com/wiki/Fire_Emblem:_Rekka_no_Ken)”.  
  
1.2 General characteristics of application

The application will be a desktop, standalone (not online), single player vs AI (Artificial Interface), with a graphical UI (User Interface) for Windows, Linux and OS X.

Game is played using six to seven buttons, four to move the cursor around, one to select/confirm actions, one to deselect/abort actions and lastly one to directly show more info about thing marked/selected by the cursor.

The application will be turn based.

A turn consists of the movement of the active player’s units around the board (a player can chose to not move any units or to move them to the same space), when the active player moves a unit next to a unit of the other player the units can “fight” (a player can chose not to “fight”), units can have different “weapons” which works in rock, paper, scissors “triangle”, units have different stats which impact different aspects of the game.

A turn ends when the actual player either has no moves left or when they chose to.

Turns then alternate between the actual player and the AI until one of them wins.

There is no time restriction to a turn (but the AI will finish as fast as possible).

The game ends when a win condition is reached.

1.3 Scope of application   
The game will lack any kind of narrative, focusing entirely on the gameplay. Sound design (background music and sound effects) will also be omitted. Player-versus-player modes will not be supported. The graphics of the game will be simple. The game will lack mouse input/control.

1.4 Objectives and success criteria of the project   
It should be possible to play a full game, controlling and attacking multiple characters in combat against computer-controlled units, and winning/losing depending on certain conditions as defined by the game.

1.5 Definitions, acronyms and abbreviations

* Turn - The period of time in which the computer and player performs their available actions before passing control to the other.
* Unit - A character controlled either by the player or the computer. A unit can move and attack. The game ends when the player or computer is out of usable units.
* Stats - Statistics. The health, attack, speed/movement, weapon(-type) values of a unit.
* Move, Movement - The action of a unit that change the position of a unit on the game board.
* Weapons - Units can have different types of weapons, which defines how well they perform in combat against other units.
* Game board - The board the game is played upon. Moving outside the boundaries of the board is impossible. Also referred to as a “map”.
* Tiles - The game board consists of a square-grid of tiles.
* Terrain - Tiles can have different kinds of terrain, which can impact combat and movement.
* Combat - When two units fight. Meaning they remove their respective attack from each other's health.
* Cursor - The cursor is the visual representation of where the action marker is.
* Marked - The space where the cursor is.
* Selected - When a unit is marked and then the action button is pressed.
* Move Buttons - Up, down, left and right. Moves the cursor.
* Action Button - Selects and confirms actions/units.
* Abort Button - Deselects and abort actions/units.
* Info Button - Show detailed information about the marked and/or selected unit/tile.
* Map - A specific type of game board.

2 Requirements  
  
In this section we specify all requirements  
  
2.1 Functional requirements  
  
Create a list of high level functions here (from the use cases).

The player can:

move units by selecting them and issuing a move command, they fight the AI’s units by issuing a fight command.

2.2 Non-functional requirements   
  
Possible NA (not applicable).  
  
2.2.1 Usability   
  
2.2.2 Reliability   
  
2.2.3 Performance   
  
2.2.4 Supportability  
  
2.2.5 Implementation   
  
2.2.6 Packaging and installation

Mhm  
  
2.2.7 Legal   
Yup

2.3 Application models  
  
2.3.1 Use case model   
  
UML and a list of UC names (text for all in appendix)  
  
2.3.2 Use cases priority  
High priority:

* Unit movement\*
* Unit Combat\*
* Unit health & dying\*
* Unit selection\*
* Turn Cycle\*
* Cursor movement\*

Mid priority:

* Tooltips\*
* Status screens\*
* Experience/levels\*

Low priority:

* Consumable items\*
* Multiple maps\*
* Shopping
* Unit skills
* Camera panning\*
* Save/load games\*
* Options
* Pre-battle preparations
* Tutorial
* Show enemy movement/attack ranges

2.3.3 Domain model  
  
UML, possible some text.  
  
2.3.4 User interface   
  
Text to motivate a picture.  
  
2.4 References  
  
APPENDIX   
  
GUI  
  
Domain model  
  
Use case texts

1. Unit movement:

Summary: How the player will move their units on the board. UC Attack can be proceeded after this UC.

Priority: High

Extends: -

Includes: Cursor Movement

Participators: Player and Computer

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Player selects unit (see Cursor Movement UC) |  |
| 2 |  | Highlight squares available to move to |
| 3 | Player selects square to move to |  |
| 4 |  | Units moves to selected square |

2. Unit combat

Summary: The player selects an enemy unit to attack with one of their units after it has finished moving.

Priority: High

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Finish moving unit |  |
| 2 |  | Highlight squares adjacent to unit |
| 3 | Select from one of the highlighted squares |  |
| 4 |  | Perform attack between selected enemy and player unit |
| 5 |  | Both units gain experience |

3. Cursor Movement

Summary: When the player moves the cursor on the game board.

Priority: High

Extends: -

Includes: Pressing buttons

Participators: Player

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Player uses arrow keys |  |
| 2 |  | Highlights the selected tile |

4. Unit Selection

Summary: When the player wants to highlight a unit in order to perform a specific action with it

Priority: High

Extends: -

Includes: Pressing buttons

Participators: Player

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Player uses action button |  |
| 2 |  | Display options menu or UC 1 |

5. Turn cycle

Summary: Once the player is satisfied with the commands they have issued, or have used all available commands, their turn ends and the computer’s turn begins.

Priority: High

Extends:

Includes:

Participators: AI and Player

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Player finishes all their available actions or manually chooses to end their turn |  |
| 2 |  | Player turn ends, enemy turn begins |
| 3 |  | Enemies finish all their actions |
| 4 |  | Enemy turn ends, player’s turn begins |

6. Unit health & dying

Summary: When a unit has been dropped to zero health by an attack from an enemy unit.

Priority: High

Extends:

Includes:

Participators: AI and player

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Enemy deals damage >= current health left of an allied unit |  |
| 2 |  | Unit is removed from play |
| 3 |  | Enemy unit gains experience points see UC Experience |
| 4 |  | Enemy continues turn |

7. Experience

Summary: When a unit has ended combat it gains experience based on the unit’s level and the enemy’s level.

Priority: Mid

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Finished combat with enemy unit and no deaths. |  |
| 2 |  | Both users gain experience points, is shown in an experience bar and as a number. |
| 3 |  | Turn continues |

8. Tooltips

Summary: When the cursor is hovering over a tile, information is shown about the unit or the terrain.

Priority: Mid

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Cursor is placed on tile (see UC Cursor Movement) |  |
| 2 |  | Shows information about terrain. If unit is standing on terrain also shows information about unit. |

9. Status screens

Summary: A screen that can be brought up by the player to view detailed info about any unit.

Priority: Mid

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Player presses “info” button while cursor is on a unit. |  |
| 2 |  | Displays a screen containing detailed info and statistics about the unit |
| 3 | Player presses “back” button |  |
| 4 |  | Displays the game board again |

10. Select and Use Consumable Items

Summary: A player can select a unit and choose to use a consumable item. Item will affect unit by for example giving it health.

Priority: Low

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Selects unit (see UC Unit selection) and chooses the Items option |  |
| 2 |  | List of items player currently carries appears |
| 3 | Selects item in list (e.g. health potion) |  |
| 4 |  | Unit is affected by the item (e.g. health increases) |

11. Camera Panning

Summary: For boards large enough to not fit on the screen, the player can adjust their view by panning the “camera” around.

Priority: Low

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Player moves cursor towards the edge of the screen |  |
| 2 |  | Pans the map into view from outside the edge the cursor is close to. |

12. Select Multiple Maps

Summary: Can select multiple maps at start of game,

Priority: Low

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | Selects option “maps” at start of game |  |
| 2 |  | Gridview of all available maps appears |
| 3 | Selects a map |  |
| 4 |  | Game starts with selected map. |

13. Saving and loading games

Summary: The payer can suspend a game and resume it at a later time.

Priority: Low

Extends:

Includes:

Participators:

|  |  |  |
| --- | --- | --- |
|  | Actor | System |
| 1 | The player chooses an option to suspend game |  |
| 2 |  | Saves the state and positions of all units on the map and stores the data |
| 3 | Player begins a new session and chooses to resume a suspended game |  |
| 4 |  | The stored data is loaded and the map is restored to the state it was before the suspension. |