

It is often the case that, in the first iteration, the customer asks for more than can actually be delivered in the allotted time frame. It appears this is what has happened with Iteration 1 of this project. Typically, requirements are revisited in subsequent iterations and 'brought down' to something that is more manageable. It is in this spirit and based on what I observed in the demos of Iteration 1 that I have developed the following requirements for this second and final iteration of your project.

The submission for this 2<sup>nd</sup> iteration is due **April 8<sup>th</sup> at 8AM**. No extension is possible. Demos will take place April 9<sup>th</sup> and/or 10<sup>th</sup>.

In terms of models, you are asked to supply a Word document (**and** its pdf equivalent) that includes:

- the functional requirements you have **successfully** addressed in your game (captured using a unique ID and a short description for each req)
- for each of your functional requirements, a description of how it was tested (by referring to **actual** JUnit tests or test procedures)
- any assumption and simplification you had to make
- a use case diagram **and** corresponding use cases
- an unbound UCM for each of your UCs and for the UC diagram
- an explanation of your architecture and design decisions including:
  - o where and why you use some design patterns
  - o what functionality resides on the client side, and what resides on the server side

The following requirements are organized in increasing order of complexity.  
To "support xxx" means you can set up and run a short test plan that demonstrates xxx works correctly.

1) In order to obtain a passing grade (of 50 out of 100) for Iteration 2 you **MUST** implement the following:

- 2\* You must display the supplied board and **it must be scrollable**
- This board must correctly refresh upon the execution of each action of a character
- 3 You must support a minimum of two characters (Amazon & Captain) while **offering a list of characters from which to select.**
- 4 You must support the placement and display of initial dwellings **and ghosts**
- 2 **You must support a cheat mode that allows:**
  - o The sound and warning **chits of a tile to be selected** from relevant lists (that consider what has already been used) rather than randomly assigned. This selection must occur each time a character ends his turn in a tile whose counters have not yet been assigned. The GUI must distinguish between such counters and those that some player has already assigned.
  - o **The roll of the dice to be fixed** each time such a roll is required at any point in the game.

- 4 **You must allow a character to start at different dwellings**
  - You must support a character moving from clearing to clearing over the same or several distinct tiles
  - You must support a character hiding
- 5 **You must support **one** round of character-to-character combat** between 2 players WITHOUT running away but with 2 \* limit. Each character selects i) a fight counter and attack direction, ii) armors, and iii) a move counter and a defense direction. Once the 2 characters have made their choices, combat is resolved: 1 death, 2 deaths, or combat stops. Ignore fatigued and wounded counters.
- 1 **You must support networking** (ask the TAs for help and advice)
  - In addition to the above, teams of 3 must support swordsman, dwarf and black knight (with relevant special abilities)
  - In addition to the above, teams of 4 must support berserker and white knight (with relevant special abilities)

2) In addition to the previous requirements, in order to obtain a grade between 51 and 69 (out of 100) for Iteration 2 you MUST implement as many of the following:

- You must have a separate display to show what is in a clearing that has many characters and treasures in it.
- You must support alerting weapons and using their times in combat
- You must support fatigued and wounded counters, and resting
- You must support treasures that are random amounts of gold between 10 and 50, a great treasure being a treasure worth 40 to 50 gold. Treasures also have random amounts of notoriety (from 1 to 20) and fame (from -5 to 30).
- You must support treasure appearing, being discovered, being looted.
- You must support victory points (except for learnt spells)
- You must support searching for hidden paths, secret passages, hidden enemies, etc. This may require assigning values to sound and warning chits.
- You must support **multi**-round of character-to-character combat for 2 or more players WITHOUT running away. Killing a character creates a treasure pile out of his/her possessions.
- You must support Lost Castle and Lost City
- In addition to the above, teams of 3 must support:
  - o Cloak of mist, magic spectacles
- In addition to the above, teams of 4 must support:
  - o Deft gloves, handy gloves, shoes of stealth, 7 league boots

3) In addition to the previous requirements, in order to obtain a grade between 70 and 85 (out of 100) for Iteration 2 you MUST implement as many of the following:

- You must support random placement of chits and random rolls
- You must support move restrictions (i.e., caves and mountains).
- You must support blocking (by monsters and other characters)
- You must support selling/buying with natives at dwellings

- You must support monsters appearing and blocking. Every monster roll should activate some monster. You need NOT implement ALL monsters: start with a Giant that is activated by any monster roll.
- You must support **multi**-round of 1 character-to- 1monster combat.
- In addition to the above, teams of 3 or 4 must support 1 character versus several (possibly identical) monsters:
  - o A team of 3 must support at least 3 distinct monsters (each activated by 2 distinct possible monster roll)
  - o A team of 4 must support at least 6 distinct monsters (each activated by a distinct possible monster roll)

4) In order to obtain a grade between 86 and 100 for Iteration 2 you **MUST** have implemented **all the previous requirements relevant to your team size** and offer something more such as:

*minor improvements*

- support the requirements listed for larger teams
- support several boards without hardwiring global clearing coordinates
- support carrying capacity and treasures, weapons and armor weight
- support many more treasures and/or monsters ☺
- a dazzling GUI (with a zoomable map and/or 3D views, etc...)

*medium improvements*

- support horses and other more complex concepts (e.g, treasures in treasures)
- support multiple character/monster combat
- support players playing simultaneously

*major improvements*

- support magic: minimally enchant tiles and cast the “Lightning Bolt” and “Roof Collapses” spells
- support the hiring and use of natives (possibly even in combat!)

Please note that for this last category of scoring, team size will be taken into account.

Some recommendations:

- unless you have already implemented it, forget about recording the activities of a character before performing them when this character plays his/her turn. Instead, assume a player specifies and immediately carries out activities once his/her turn starts.
- forget about building or verifying the paths of a board

This is the first draft of these requirements. I reserve the right to modify this document in the first week of March after reviewing it in class on March 3<sup>rd</sup>. So please check carefully the version as I will increment it if I repost.

Finally I reserve the right to give more than 100 to a team that will have significantly distanced itself from most other teams.