Introduction to the Project

The main goal of this project is the development of teamwork dynamics in the context of a complex programming assignment (by means of its implementation difficulty, but also due to rigorous and demanding deadlines) that demands its development within a team. More specifically, the project is a conversational adventure defined in a modular way by a set of elements, and the actions/operations that the player can perform on/with them. The basic modules of the project are:

1. ***Space*** module. Abstract Data Type (ADT) determining the different possible localizations in the adventure. A Game has a map composed by a set of Spaces that can be explored by the player, according to the decisions taken and subsequent actions.
2. ***Object*** module. ADT representing the characteristics of an object, i.e., position, description, mobility property, etc.
3. ***Player*** module. ADT related to the player (the user) of the adventure, and that describes all the information regarding the player status (location, owned objects, etc.).
4. ***Game*** module. ADT with all the information needed to create and handle a game. In this regard, this ADT contains the information about a player and the lists of objects, spaces, links...
5. ***GameLoop*** module. It is the main program of the game, which is responsible for initializing it at the beginning, updating it according to the rules and user commands, and displaying status, and releasing all the resources at the end.
6. ***Command*** module. It implements the ADT necessary for the user to perform all interactive operations that cause changes in the state of the game (such as picking up an object from a space and leaving it elsewhere).
7. ***GameReader*** module. Module for reading the file or files with the program settings, i.e. to initialize the game (creating the Player, Spaces, Objects, etc.).
8. ***GraphicEngine*** module. ADTs and functions for handling the graphic interface. Conjunto de TADs y funciones para el manejo de la salida de la interfaz de la aplicación.
9. ***Screen*** module. Module responsible for managing the different areas of the screen and printing on them.
10. ***Set*** module. ADT that gives support to the rest of modules. It is a structure that contains a list of unique numerical identifiers.
11. ***Die*** module. Module to get random values in a range as obtained when a die is rolled.
12. ***Inventory*** module. Every user has an inventory of his/her own, which is the set of objects belonging to him/her.
13. ***Link*** module. Module to connect Spaces inside a map. This connection could represent a door, a window, etc.
14. ***GameRules*** module. This module determines how the Objects and Spaces' properties can change apart from the user commands. The set of implemented functions provides a more dynamic game, so that it just not depends only on initialization files and commands of the Game.
15. ***GameManagement*** module. Its aim is to save and recover the game status.
16. ***Dialogue*** module. This module allows a more natural like interaction with the user.

The hierarchical structure of the modules is shown in Figure 3.

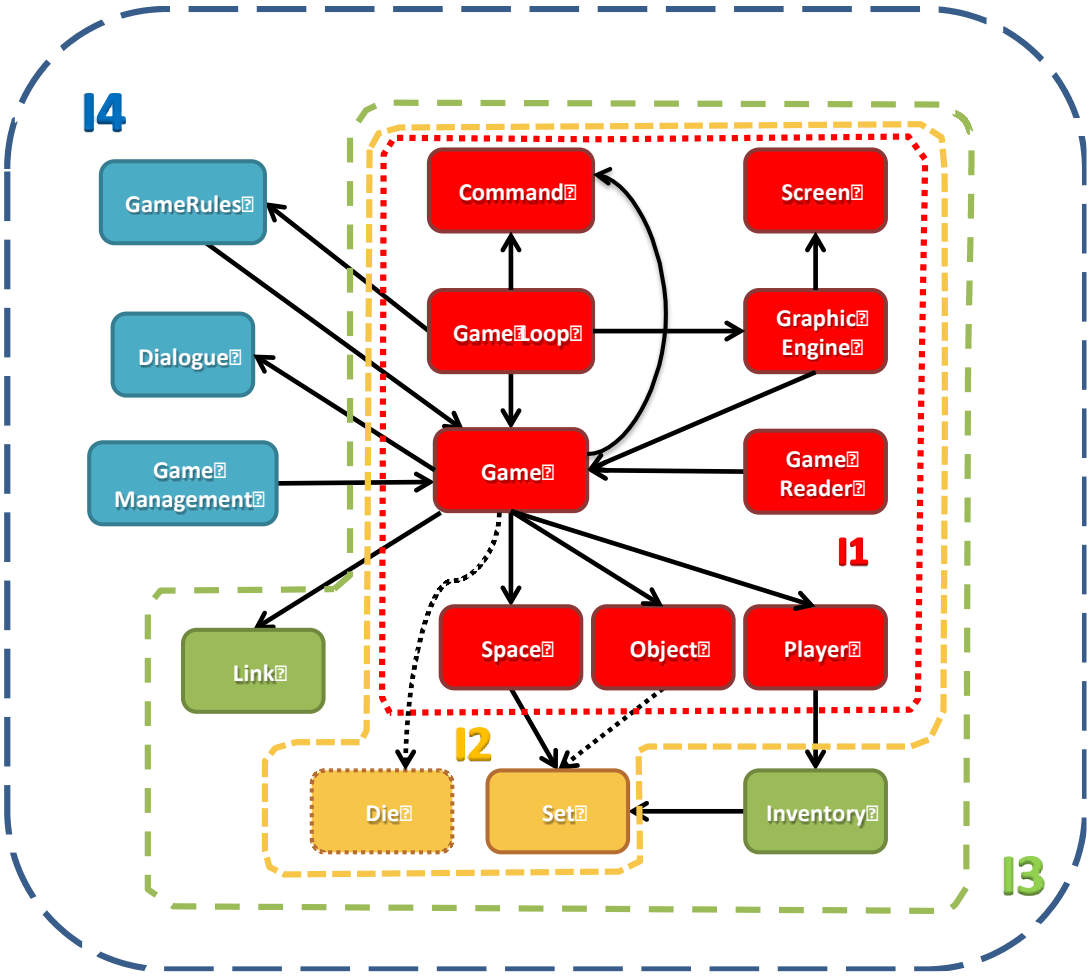

Figure 3. The hierarchical relationship between the different modules that make up the project is displayed, as well as how that will be implemented in successive iterations. In red, the modules to be implemented since the first iteration (I1). Orange modules, those started in the second iteration (I2), which also extend and use the red ones. The compulsory modules in red and orange are re-expanded and used in the third iteration (I3), where the green modules are also added. The mandatory modules are completed in the fourth iteration (I4) for supporting an original text adventure, with a few blue desirable optional modules.

Requirements may change depending on the score to which you are applying and the team size. The following table provides guidance on project modules. The second column indicates whether the module is required to pass the course. The third and subsequent columns define the extent to which modules are required to complete each of the four iterations[1]. **All modules** must be accompanied by an application to **test its functionality**, as well as the indicated documentation and the necessary ***Makefile*** for compilation and linking.

[1] When computing the estimation of the total hours needed for the project, it should be considered that the time must be divided between all the members in the team and the different iterations involved. On the other hand, it is not necessary to do all the optional modules for obtaining a maximum grade.

Module	Compulsory	Required compulsory version for Iteration			
		I1	I2	I3	I4
Space	Yes	Basic	Improved 1	Improved 2	Complete
Object	Yes	Basic	Improved 1	Improved 2	Complete
Player	Yes	Basic	Improved 1	Improved 2	Complete
Game	Yes	Basic	Improved 1	Improved 2	Complete
GameLoop	Yes	Basic	Improved 1	Improved 2	Complete
Command	Yes	Basic	Improved 1	Improved 2	Complete
GameReader	Yes	Basic	Improved 1	Improved 2	Complete
GraphicEngine	Yes	Complete	Complete	Complete	Complete
Screen	Yes	Complete	Complete	Complete	Complete
Set	Yes	No	Complete	Complete	Complete
Die	Yes	No	Complete	Complete	Complete
Inventory	Yes	No	No	Complete	Complete
Link	Yes	No	No	Basic	Complete
GameRules	No	No	No	No	Complete (if submitted)
GameManagement	No	No	No	No	Complete (if submitted)
Dialogue	No	No	No	No	Complete (if submitted)