Alejandro Reyes Bautista – Q13134663

Robustness and Sequence Diagrams Presentation: SCRIPT

Slide 1:

This presentation is about how Robustness and Sequence Diagrams are being used to improve both use cases and domain model of the Athletics club Demonstration Program.

Uses cases and Domain Model are the start point. After that, robustness diagrams have helped us to discover new steps, extensions and classes. In the same manner, the distinct sequence diagrams have helped us to discover methods, classes and attributes.

Slide 2:

The case of study chosen as example is the use case “**Add an event”**. The use case has a first part with the data needed to contextualize it. Name, description, who uses the use case, preconditions required, actors, triggers and state of the system after the execution.

At the second part we develop the step of actions, the extensions branch and variations. The steps enumerated belong to the most common case or ‘happy path’.

We observe that, at the moment, we only have 4 steps, in red, and 2 extensions branches, in green.

Slide 3:

Now we develop the robustness diagram. At the beginning we are in home page, so we do not have the fields to insert the required data. We discover that the step ‘User selects add and event’ and the step in which the system displays that screen were omitted. Other alternative to that could be include as precondition that the user is in front of the interface for add an event, but we chose the first option.

After the user has inserted the required data (type, sex, age group, etc.) and it has been validated (not empty fields and format matches), the step 3 of the original use case says that the system creates the event. But if the event is duplicated the system could throw an error or we should have redundant data within the system, so the best guard would be to check the event does not already exist, another missed step. Finally, when the system notifies that the event has been added successfully in a confirmation dialog, we should give the use the possibility to come back to the main page, so we added the step 8 and 9. We can see on the top left the use case and on the top right the new extended use case.

Slide 4:

Subsequently we develop the sequence diagram for the man path.

The distinct boundaries in the previous robustness diagram compose the distinct screen of the View or GUI. The distinct tasks developed by the controls are encapsulated in the controller, which interacts with both GUI and entities. This is the best moment to discover new methods. And sometimes entities and attributes too. In this example, both in the robustness diagram and sequence we realize that for check if a concrete event already exists, we need an entity that were a list of events or contains the list of events. In the same manner, we discover two methods, one is the method that checks if an event already exist, and the other is the one that add the new event to the list of events (so the next time a new event is going to be created, it will also be checked with the last event created).

Slide 5:

The sequence diagram has highlighted some new features that we have to add to the domain model. We do not need a new class for the entity that represents a list of events. It can be a static List inside Event class. Also we add the new methods to the Class Diagram, keeping the name, parameters and return values as we have defined before.