Zackary Morelli

1/17/2022

[zackmorelli@gmail.com](mailto:zackmorelli@gmail.com)

Tiamat is an automatic plan creation script that I had a large vision for when I set out to make it after the department upgraded to Aria version 16 and thus made write-enabled scripting available to us. It is named after the Babylonian goddess of creation. It was supposed to be a large program that would use write-enabled scripting to automate parts of the planning process, mostly by setting up beam geometry based on anatomical selections by the user.

However, it was decided (and I agreed) that it was easier to automate treatment planning, and beam setup particularly, through the use of Eclipse plan templates. It was simply a matter of the department making a set of standardized plan templates and using them, which was not new in version 16, but had not been done yet at the point when we were going through this process.

So, I modified Tiamat so that it is only used to add imaging setup beams to a plan, which is useful, but all the code for doing all the other things that I had originally planned on is still there. If you are interested in ESAPI’s tools for programmatic beam creation and optimization objectives, I’m sure the code and set of classes I built out will be useful to you.

To be clear, Tiamat is currently installed on the clinical system and it is write-approved. But it only adds imaging setup beams. It follows my standard structure of a script that launches in a ScriptExecute file that takes information from the ScriptContext and passes it to a method that starts a WinForm GUI. The user makes selections on the GUI. When the user presses an execute button, an execute method in the GUI then parses through the selections made by the user and passes them, along with the plan info, to a method kept in a separate class called TiamatExecute, which is where most of the program takes place. TiamatExecute then calls many different methods to do various things that are stored in a backend TiamatClasses file.