Sequence diagramming using UML

In this lab you will our UML diagramming tool to create a sequence diagram for the use case in your rapid prototype. You can use a sequence diagram as an aid to help your team to design an interface with the end-user. As with the other UML labs, experience with modeling will be applicable to potential internships or careers.

What I Want You To Learn

- The basics of using a UML tool to create a sequence diagram
- Create a sequence diagram that could be used to interact with an end-user

NOTE: You may use a different UML tool than Papyrus if you want.

Download and install the Papyrus tooling for Eclipse

In Eclipse:

- Go to Help->Install New Software...
- Choose Add
- Type Papyrus for the name and this for the URL:

http://download.eclipse.org/modeling/mdt/papyrus/updates/releases/luna

- Check the boxes to accept license and install the Papyrus tools

Read through Papyrus tutorials (on Sakai):

- TutorialOnPapyrusUSE_d20101001.pdf
- PapyrusTutorial_OnSequenceDiagrams_v0.1_d2010100.pdf

Create a sequence diagram

- **Create a Papyrus project and copy your SimpleSimonGame.java file into the project (or whatever class/classes you created that extend the ApplicationAdapter or implement your core logic for your Simple Simon rapid prototype).**

- **In a block comment at the beginning of your SimpleSimonGame.java (or other java file), write out one user story / use case for your simple simon experiment representing a successful play of the game. This should be a short paragraph in English describing how the user would perceive one play through of the game with a "success" result.**

You should now create a sequence diagram of this user story / use case for your rapid prototype. Your diagram must have at least 3 actors and lifelines:

- User
- Application
- libGDX (this is the "system")

You may feel free to add additional lifelines by splitting the Application lifeline into multiple smaller components if you created your program with several classes or screens.

Criteria for assignment

1. State the name of the use case (simple simon experiment)
2. Illustrate and label actor(s) and the system
3. Illustrate interactions between the actor(s) and the system. Label the messages.
4. Use conventional UML sequence diagram notation

Grading Rubric

Criteria	Done Well	Need Improvement
Illustrate the actor(s) and the system; name the Use Case	Actor(s) and the system are shown and labeled; use case is named	Actor(s) and the system are missing
Illustrate the messages between the actor(s) and the system with appropriate labels	Events from the actor(s) and messages from the system are shown and labeled	Messages from the actor(s) and system responses are missing
Use the UML sequence diagram notation as shown in the tutorial	The diagram follows conventional UML sequence diagram notation	The diagram does not follow the conventional UML sequence diagram notation