

# SimSE Player's Manual

---

## 1. Beginning a Game

When you start a new game, you will see a window pop up that contains a short narrative about the game you are about to play. It is **IMPERATIVE** that you read this narrative very carefully, as it will usually tell you what the goals of the game are, how it will be evaluated whether you meet that goal, and very important hints about how to succeed in the game. At any time during the game, you can come back to this narrative by clicking the “i” in the upper left-hand corner of the user interface.

## 1. Viewing Resources

You can see the status of all of the major entities in the game (your artifacts, customers, employees, projects, and tools) in a few different ways:

1. To view an object's details, you can first click corresponding word to that object's type (e.g., for an artifact, click “artifacts”, for an employee, click “employees”) in upper left-hand corner, next to the SimSE logo, and then all of that particular type of object will appear in the grid to the right. Clicking on one of the avatars will bring that object and all of its attributes into view in the bottom area.
2. For employees, you can also left-click on their avatar in the office, or in the “Current Activities” area on the right. This will bring the employee and his/her attributes into view in the bottom area.
3. You can see a quick “at-a-glance” view of all of the objects of a particular type by clicking on the “ALL” button in the grid after performing step 1. This will bring up a table of all of the different objects and their attributes. You can hide and unhide attributes/columns by right-clicking on the column.

## 2. Interacting with Employees

You can interact with your employees through right-click menus. Right-click on one of their avatars in the office, on their avatar in the “Current Activities” area on the right, or on their avatar in the grid in the upper right-hand portion of the interface, and you will be presented with a menu of options. This is the only way to perform actions in the game.

When you use an employee's right-click menu to perform an action, the employee that you right-clicked on is automatically added as a participant in that action. For example, if you used “Andre's” right-click menu to start the “Create Requirements” action, “Andre” automatically gets assigned to the “Create Requirements” action. Thus, when prompted to choose other participants for this action, “Andre” will not appear as a choice (since he has already automatically been added as a participant).

## 3. Moving Forward through Time

The simulation clock is located in the lower right-hand corner of the user interface. You can move time forward through the game in 2 different ways:

1. Enter the number of clock ticks you wish to advance forward in the clock's text field, and then click the "Advance Clock" button. If you check the box "Stop at Events", the clock will go until either: a) one of the employees has something to say, or b) it has already gone forward the specified number of clock ticks, whichever comes first.
2. Click the "Next Event" button, which will cause the simulation to move forward until one of the employees has something to say. (This is the most common way of stepping through time in SimSE.)

When you tell the clock to move forward, you should first make sure that you have in focus an object of interest. For example, if you have just told your employees to work on requirements, you should bring the requirements document into focus (make it visible in the lower portion of the user interface by clicking on "Artifacts" then the requirements document icon) before you step the clock forward (or just after starting the clock forward). This way you can monitor the action to see how their work is progressing by watching how the size, errors, etc. of the requirements document change.

#### **4. Awareness**

In addition to viewing the status of the simulation objects (artifacts, customers, employees, projects, and tools), there are a couple of other ways to stay abreast of the events that are occurring in your game:

1. The current activities of all of your employees are listed on the right hand side of the GUI.
2. The employees will often inform you of important events, the beginning and ending of activities, hints, and other guidance through pop-up speech bubbles that appear over their heads. It is important that you read these speech bubbles carefully, as they will often give you important information that will greatly help you succeed in the game.

#### **5. Explanatory ("Analyze") Tool**

The explanatory tool allows you to view graphical representations of your game in the form of line graphs, discover insights into some of the rules underlying the game, and create multiple parallel timelines, or game branches, to explore alternative decisions. The explanatory tool can be accessed through the "Analyze" menu in the upper left-hand corner. The "Analyze" menu is enabled throughout the game, allowing you to use the explanatory tool during the game.

##### **5.1 Generating Graphs**

There are three types of graphs that can be generated: object graphs, action graphs, and composite graphs.

An object graph depicts how an object's attribute values changed over time, and can be generated by choosing an object in the drop-down list labeled, "Object Graph", choosing one or more of that object's attributes in the list marked, "Show Attributes", and then clicking the "Generate Object Graph" button. Figure 1 shows an object graph for an employee's energy and mood. Time is represented by the horizontal axis and attribute value is represented by the vertical axis. The title of the graph indicates which object's

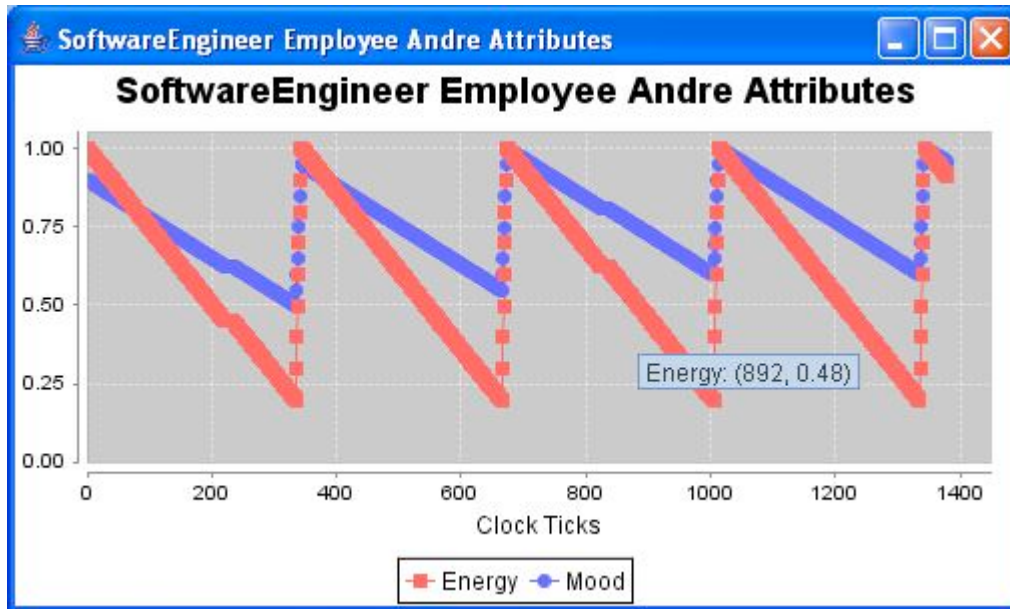
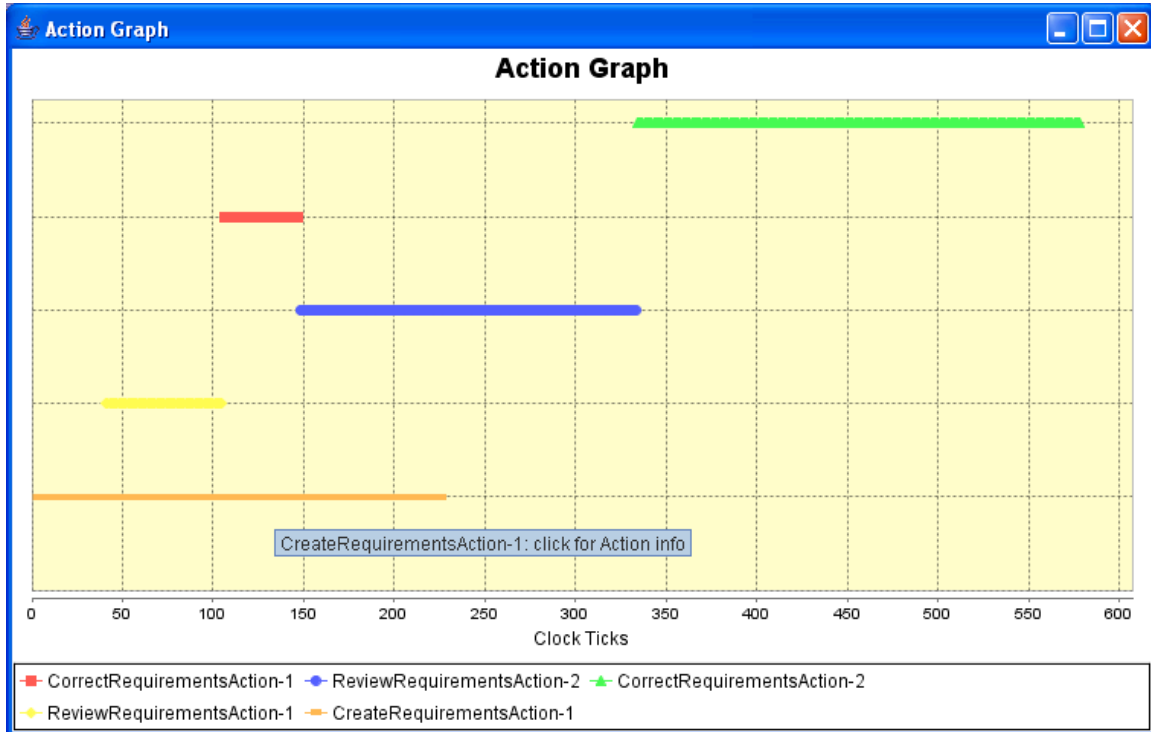


Figure 1: An Object Graph Generated by the Explanatory Tool.

attributes are being graphed—in this case, a “SoftwareEngineer” Employee named “Andre.” The key below the graph explains which data points correspond to which attributes. Any data point in the graph can be moused over to reveal that point’s exact x- and y-values. In Figure 1, the data point for the energy attribute at clock tick 892 is being moused over, at which point the employee’s energy was 0.48.

An action graph provides a trace of events/actions/activities that occurred in the simulation, and can be generated by choosing one or more actions to graph in the “Action Graph” list and then clicking the “Generate Action Graph” button. Figure 2 shows an example of an action graph that includes three different types of actions: “CreateRequirements”, “ReviewRequirements”, and “CorrectRequirements”, with one occurrence of “CreateRequirements” and two occurrences each of “ReviewRequirements” and “CorrectRequirements” (multiple occurrences are indicated by the number at the end of the action label, e.g., “ReviewRequirementsAction-2”). The x-axis indicates time progression, in clock ticks. The y-axis has no semantics, but only serves as a delineator for graphing actions—each action is graphed on a separate gridline on the y-axis. The key below the x-axis indicates which data points correspond to which actions. The data points for an action begin at the time that action was triggered and end at the time that action was destroyed. For example, in Figure 2 the “CreateRequirements” action, represented by the orange (bottom) line, began at clock tick 0 and ended around clock tick 230.

Mousing over a data point will display the name of the action and a reminder that the data point can be clicked on for more information, as shown for “CreateRequirementsAction-1” in Figure 2. When a data point in an action graph is clicked on, the details and effects of that action are displayed in the form of the screen shown in Figure 3. There are two tabs in this screen: Action Info and Rule Info. As their names indicate, the Action Info tab contains information about the action and the Rule



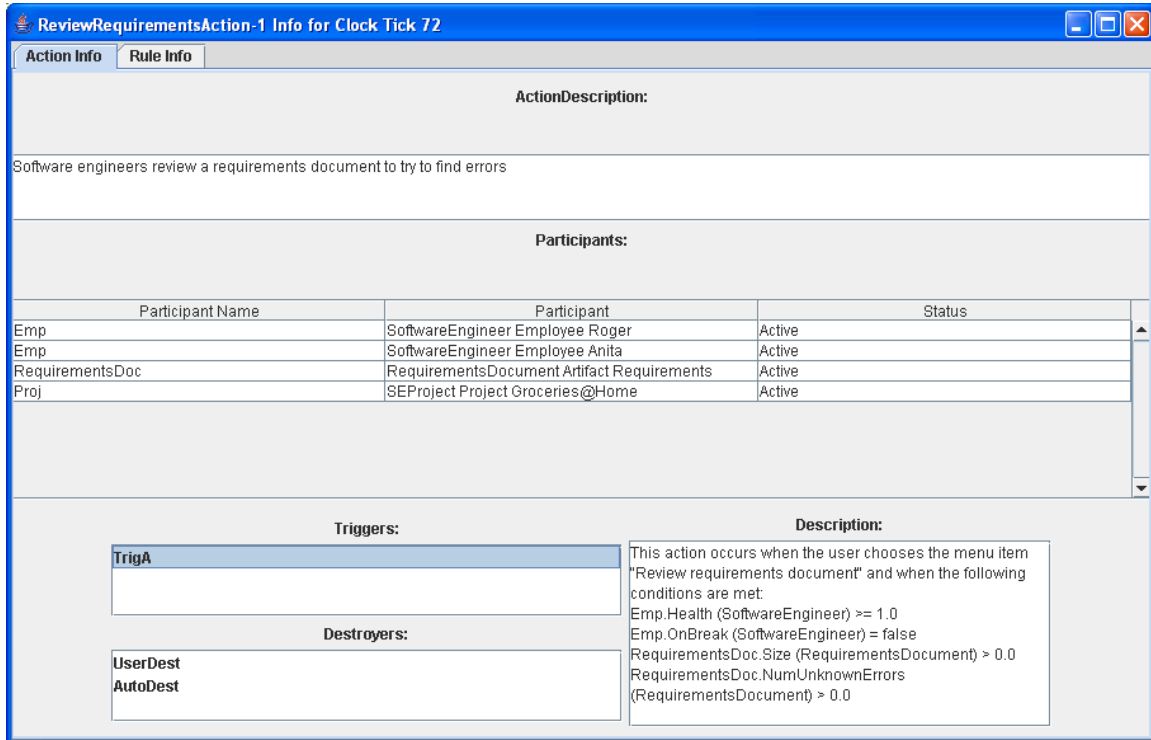
**Figure 2: An Action Graph Generated by the Explanatory Tool.**

Info tab contains information about the rules that are attached to that action. A rule defines what effect an action has on the rest of the simulation (e.g., one rule for a “CreateCode” action is that the size of the code increases every clock tick by an amount based on the employees’ productivity in coding).

The Action Info tab is divided into three portions, one for each type of information provided about the action. The top portion contains a description of the action. The middle portion displays the participants that were involved in the action during the clock tick of the selected data point (the point that was clicked on to bring up the action information. The bottom portion of the Action Info tab lists all triggers and destroyers for the action, so that the player can see exactly what could have caused the action to either stop or start. A user can click on any one of these triggers or destroyers to bring up a description in the field to the right.

The Rule Info tab is shown in Figure 4. On the left are listed all of the rules that are attached to the selected action. Trigger rules are those that occur when an action starts. Intermediate rules are those that occur once every clock tick during the duration of the action. Destroyer rules are those that occur when an action stops. Any one of the rules in the list can be clicked on to bring up a description of that rule in the right hand pane.

The third and final type of graph that can be generated is a composite graph (see Figure 5). A composite graph shows both an object graph and an action graph lined up on the same time axis. The composite graph shown in Figure 5 contains an object graph for the “RequirementsDocument” artifact, including the two attributes “NumKnownErrors” and “NumUnknownErrors”, and an action graph that includes the same three actions from the action graph in Figure 2 (“CreateRequirements”, “ReviewRequirements”, and

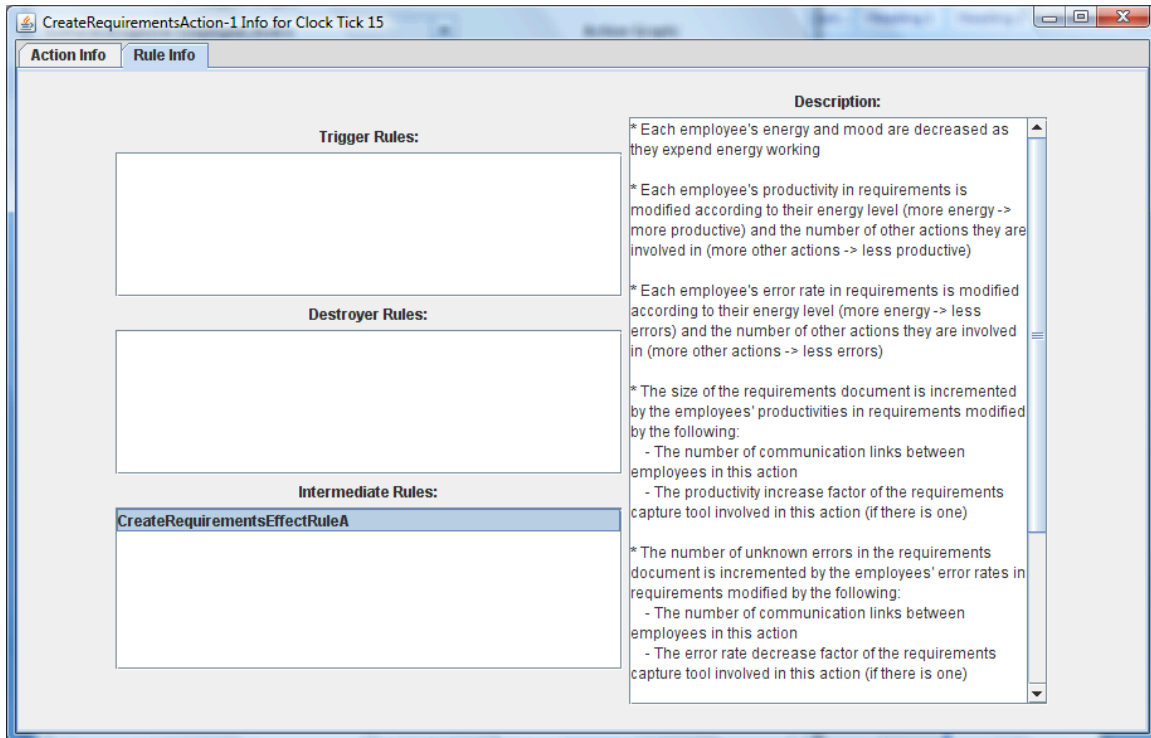


**Figure 3: Detailed Action Information brought up by Clicking on an Action in an Action Graph, with the Action Info Tab in Focus.**

“CorrectRequirements”). To generate a composite graph, simply choose the object graph and action graph settings you want, then click “Generate Composite Graph.”

Because it might not be obvious at first how a composite graph is useful, let us look at an example from the graph in Figure 5. Upon looking at the last upward slope in the unknown errors (approximately clock tick 330 to 580), it might be unclear why this slope is there. You would probably first notice this effect by seeing the hidden attributes revealed at the end of the game and observing that, even though you did a thorough review of the requirements document and corrected all of the known errors, there were still undiscovered errors in the document at the end. Looking at this composite graph would provide you with the reasoning behind this: The final upward slope in unknown errors corresponds exactly to the second “CorrectRequirements” action, indicating that this action caused some more unknown errors to be introduced into the requirements document. Clicking on this action in the graph will then reveal why—employees correcting requirements will introduce new errors into the document at a rate dependent on their requirements skill. Thus, you can infer from this that it is just as important to have skilled personnel involved in requirements document correction as it is for requirements document creation, and will likely be more careful in assigning people to this task in the next game.

All graphs can be further customized in terms of appearance by right-clicking on the graph—they can be zoomed in or out on, colors can be changed, and labels can be turned on or off. Using the right-click menu, you can also print a graph or save it as an image if you want to keep it for future reference.



**Figure 4: Rule Info Tab of the Action Information Screen.**

## 5.2 Viewing Rules

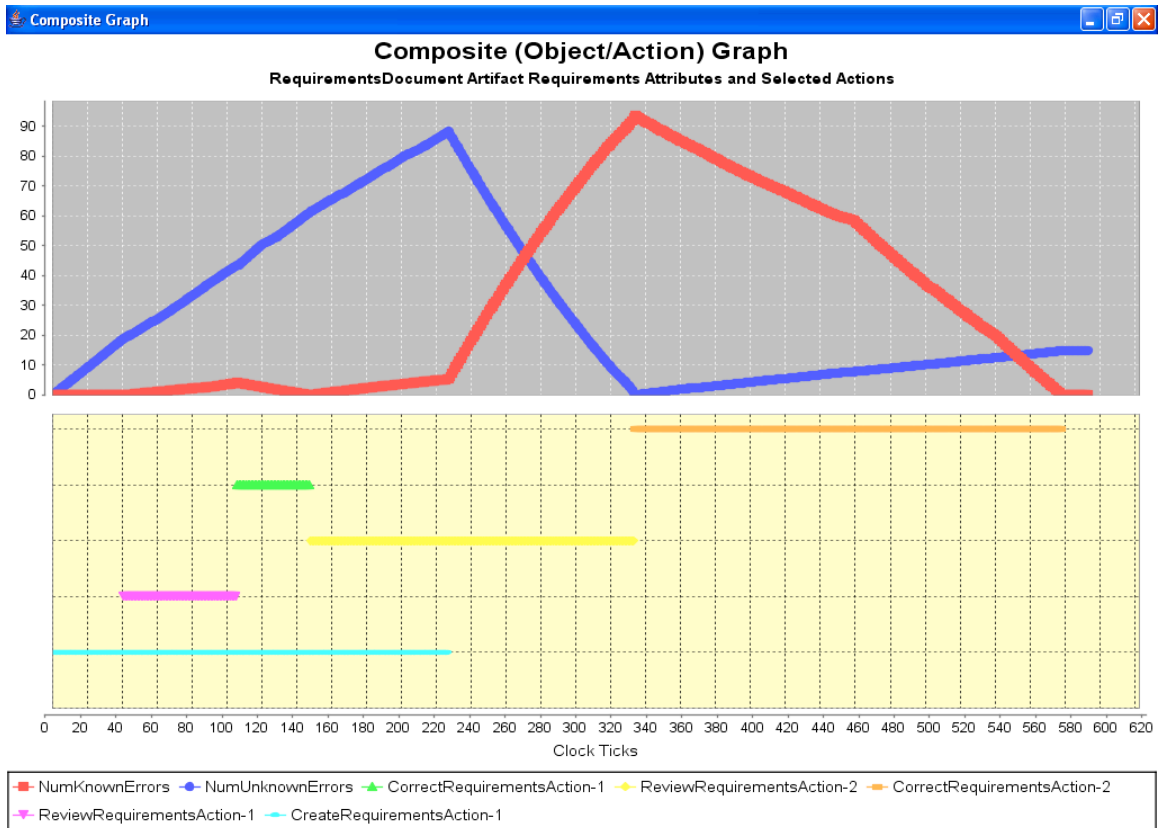
In addition to viewing rules through clicking on an action graph, rules can also be viewed through the lower half of the explanatory tool main user interface by choosing an action in the “Actions” drop-down list. This will make the rules appear in the lists below the drop-down list. Any rule in the lists can be clicked on to bring up its description in the window to the right.

Reading rules is one of the most effective ways to learn how to improve your score. As an example, let’s say that your requirements document ended up with numerous errors, and you don’t know why. Looking at the rules attached to the “CreateRequirements” action will tell you. One of the rules might say, “The number of errors in the requirements document increases at a rate dependent on the requirements experience of the developers working on it”, and another rule might say, “The rate at which errors are introduced into the requirements document is lessened by the use of a requirements tool.” From this, you could conclude that you should use developers with more requirements experience and also be sure to use a requirements tool.

As another example, you can often discover exactly how your score was calculated by reading the rule(s) attached to the game-ending action (e.g., “DeliverProduct”).

## 5.3 Branching a Game

You can return to any previous point in your game and start a new game from that point. This can be done in two different ways: using the explanatory tool graphs’ right-click menus, and using the Multiple Timelines Browser. To branch a game from an



**Figure 5: A Composite Graph Generated by the Explanatory Tool.**

explanatory tool graph (either an object graph, an action graph, or a composite graph), simply right-click on any point on a graph, and choose the option, “Start new game from here.” It will then prompt you to name that particular game, and the new branch will begin. Your current game will still be accessible as you also run the new branch. Try to keep the number of simultaneous games to about 5 or less, or else the performance will significantly suffer. The Multiple Timelines Browser will be covered in the following subsection.

### 5.4 Multiple Timelines Browser

The Multiple Timelines Browser, accessible through the corresponding button in the explanatory tool, allows you to see a graphical depiction of all of your parallel game branches that you have started (see Figure 6). A solid line represents a game that is still open, a dashed line represents a game that has been closed, and a number at the end of a line indicates the score given for that game. Left-clicking on a line (only for open games) brings that game to the front. Right-clicking on a line gives you the option of starting a new branch from that game. As in branching from an explanatory tool graph, you will be prompted to give the branch a name, and it will then start.



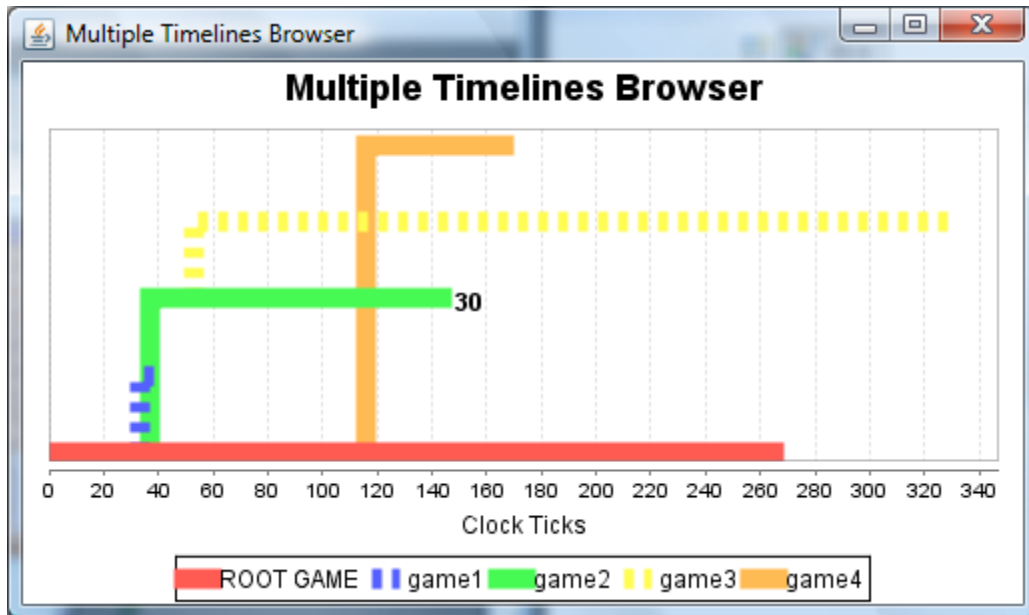


Figure 6: Multiple Timelines Browser

## 6. Ending a Game

Once the game is over, take a look again at your various artifacts, tools, etc. Attributes that were hidden throughout the game will appear (e.g., number of unknown errors), giving you further insight into how you did.

## 7. Important Tips for Success

- If you ever feel “stuck” and can’t figure out how to successfully get through the game, go back and look at the starting narrative (by clicking on the ‘i’ in the upper left-hand corner of the user interface. This narrative contains important information that might make more sense to you now that you have played a little bit.
- Be sure to read the employees’ speech bubble comments carefully! Sometimes they will give you valuable hints about what your next step(s) should be, or give you some other important information that could help your game immensely.
- When using the explanatory tool, keep an eye out for “secret” actions—these are actions that were not visible during the game but are visible in the explanatory tool graphs and rule descriptions. For example, in one model, there is a “DoubleProductivity” action that causes the whole team’s productivity to double when things are being done in an ideal way (what the “ideal way” is could be discovered by looking at this action’s trigger(s)).
- Be sure to read the rules in the explanatory tool! They are probably the most helpful sources of information, and are often overlooked by players.
- Always look at the rule(s) attached to the game-ending action (e.g., “DeliverProduct”). They will often tell you exactly how your score was calculated.