Paper Prototyping & Usability Testing

For CATGames Developer Interface

IAT 334
D101 - Team 1

Evan Miller
Andrew Nip
Nathan Waddington
Shanna Walters

Fall 2008

Introduction

Design Challenge:
The team is building a system that works with the Gestural and Tangible Prototyping system (part of the CATGames project), which interfaces between game designers/programmers and a physical prototyping system. This is particularly challenging as the Gestural and Tangible Prototyping tool is a collection of physical objects all of which have their own specific properties that need to be interfaced with in intuitive manner. As the people we are targeting as an audience tend to be very time-constrained and may not be willing to spend the time to discover the functionality in the system it's particularly important to make sure the system can be understood and used rapidly.

Purpose and Rationale of Testing:
The prototype is intended to test the process of building a simple game with our proposed interface: setting a winning condition, testing input for that condition, and outputting the result onto a screen.

CATGames:

Creativity Assistive Tools for Games Network is a research network consisting of The University of Western Ontario, Simon Fraser University, Credo Interactive, with Seneca College as the host organization. Its vision is to create innovative, leading edge technology tools for game production that support Canada's burgeoning games industry by accelerating the creative process, and expanding and enriching content environments and platforms. (CATGames Proposal, 2007)

Prototype

Our prototype consists of a single sheet of paper with a series of sections drawn on it. On the upper left corner is an area designated "Rules," right below it is "Input Devices", followed by "Output Devices." To the right of these are two sections: a "Rules Editor" and an "Outcomes." The left side of the page could be described as a menu area with the user's toolset and the right side of the page can be generalized as the working area.

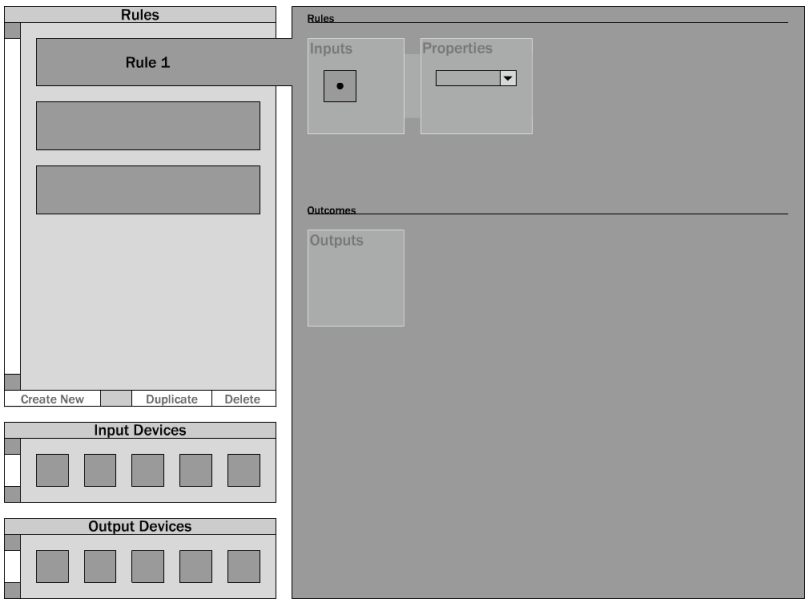
Interaction options with our interface follow common conventions. A combination of drag and drop items, drop-down menus, type boxes and buttons are used in the interface. A user can create a new rule to have a corresponding work space created. When a rule is selected it visually 'tabs' to the associated work space. From here users create 'rules' using tools for taking in input from objects, such as dice, and create output depending on input.

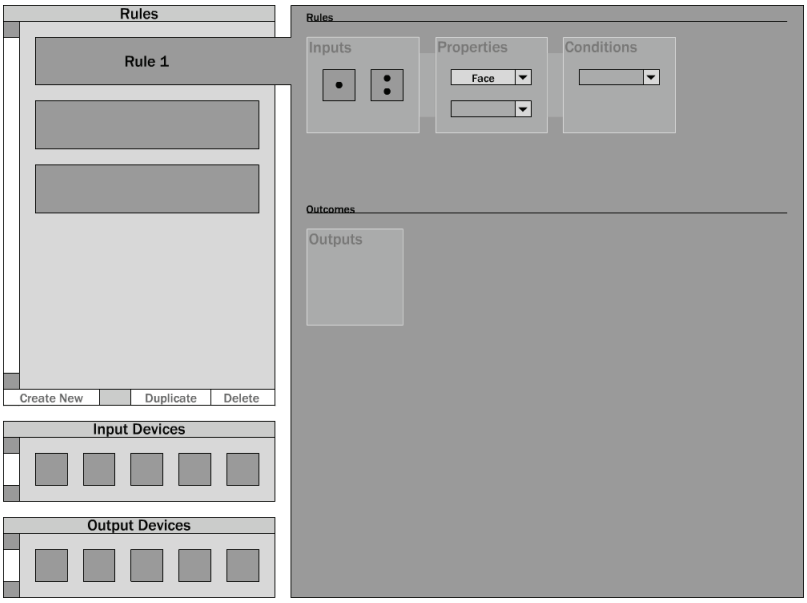
Objects able for use in the Input Devices and Output Devices section are drag-and-drop items. Input devices can be pulled into the Rules workspace and Output devices can be pulled into the Outcomes workspace. Once any of these icons is dragged into a workspace, more corresponding options come up on screen. Users then select a property related to each input/ output device they want to set parameters for or create an associated ouput, such as a message to appear on a screen.

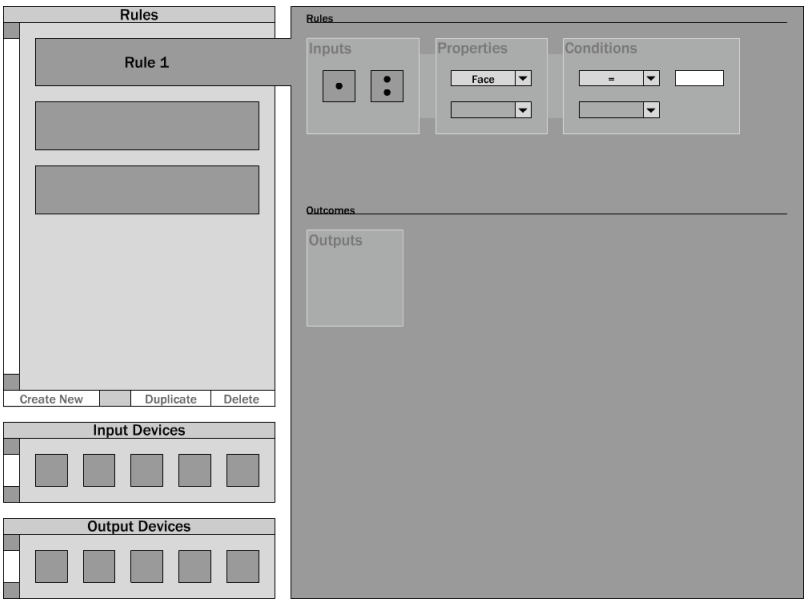
To replicate user actions, we used Post-It notes to allow participants to manipulate the user interface. When a participant 'interacted' with any part of the interface, post-its were placed or removed to reflect the actions and the sytem's responses. Participants used a pen in place of a mouse, pointing at objects they would be clicking on. For text boxes in the paper prototype, participants were asked to use a pencil to write in what they would otherwise type.

Wireframes:

The following four pages contains screen-shots of our wireframes that the team came up with and based the flow and organization of our paper prototype on. The final image is a shot of our actual paper prototype that was used in our usability testing.

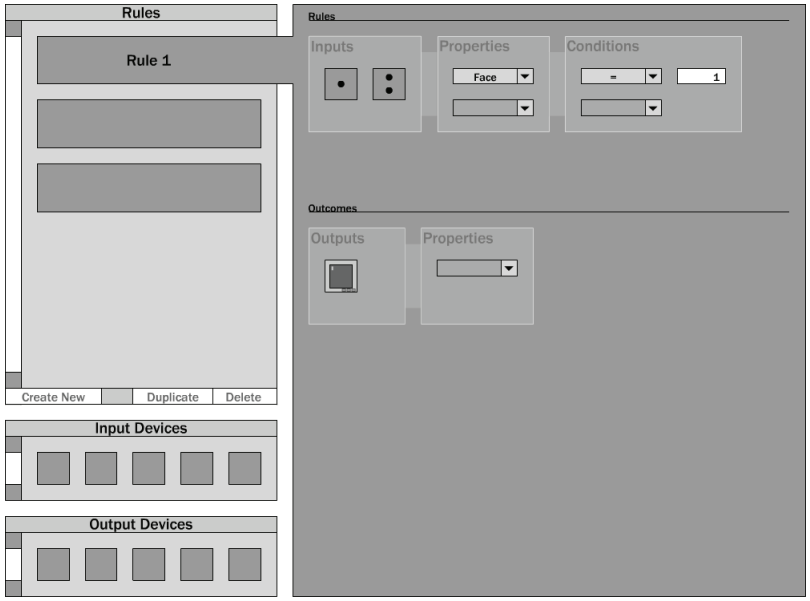


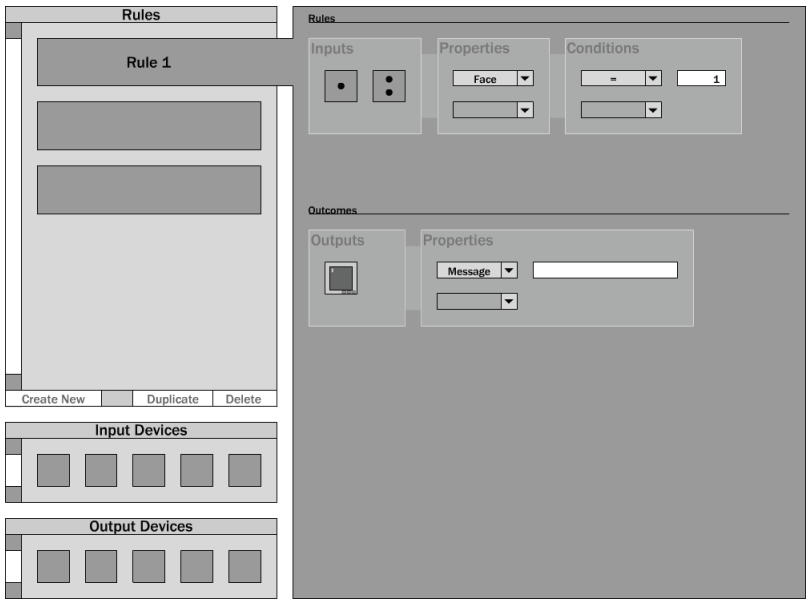


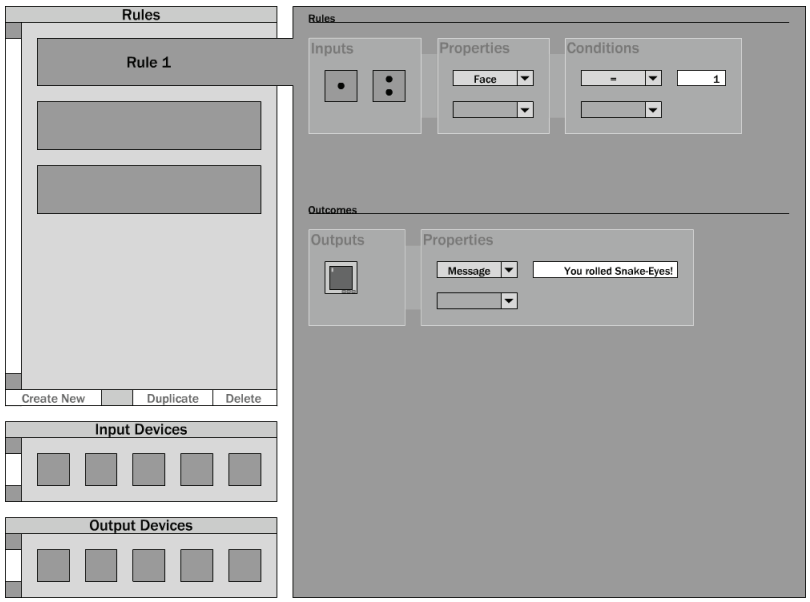


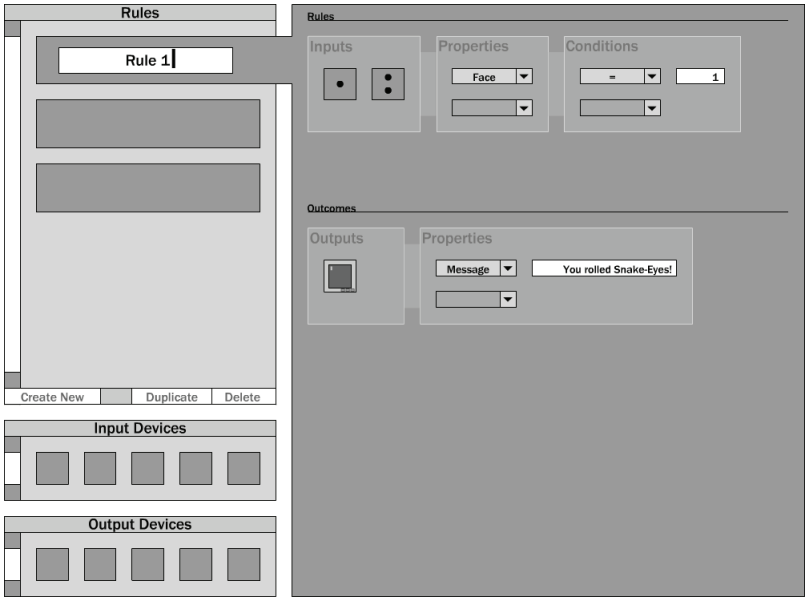
A Step-by-Step Process:

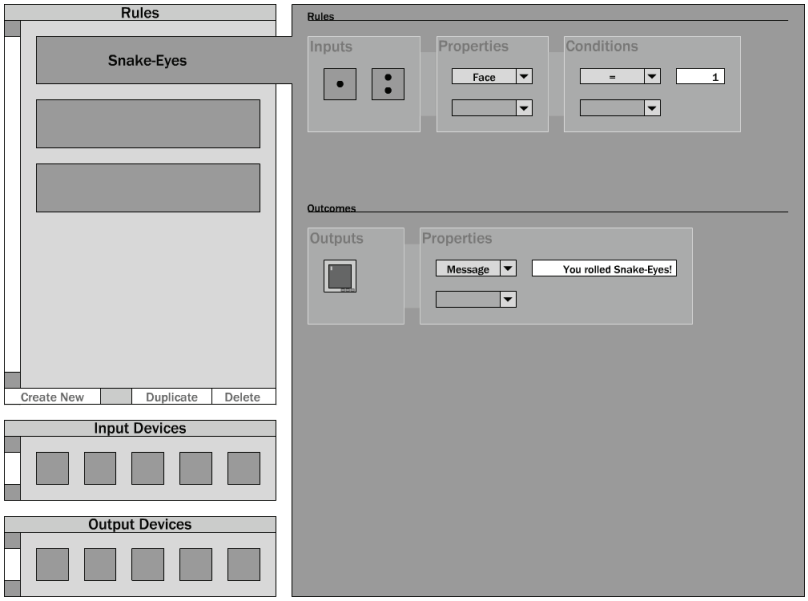
These wireframes show a step-by-step progression of the screen as we envision the users will go through to complete the tasks given to them.

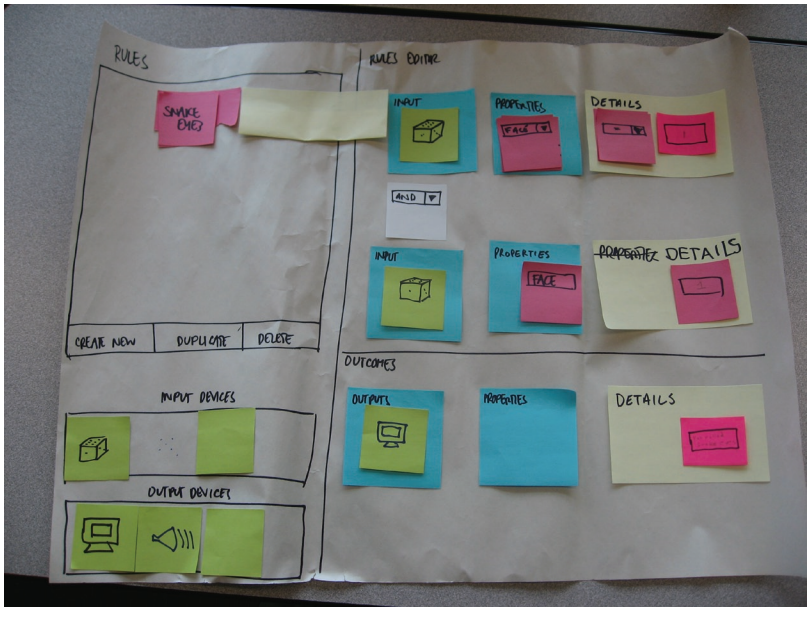

Rules
Rule 1
Create New
Duplicate
Delete
Input Devices
Output Devices
Rules
Inputs
Properties
Face
Conditions
=
1
Outcomes
Outputs
Properties


Rules
Rule 1
Create New
Duplicate
Delete
Input Devices
Output Devices
Rules
Inputs
Properties
Face
Conditions
=
1
Outcomes
Outputs
Properties
Message


Rules
Rule 1
Create New
Duplicate
Delete
Input Devices
Output Devices
Rules
Inputs
Properties
Face
Conditions
=
1
Outcomes
Outputs
Properties
Message
You rolled Snake-Eyes!


Rules
Rule 1
Create New
Duplicate
Delete
Input Devices
Output Devices
Rules
Inputs
Properties
Face
Conditions
=
1
Outcomes
Outputs
Properties
Message
You rolled Snake-Eyes!


Rules
Snake-Eyes
Create New
Duplicate
Delete
Input Devices
Output Devices
Rules
Inputs
Properties
Face
Conditions
=
1
Outcomes
Outputs
Properties
Message
You rolled Snake-Eyes!


RULES
SNAKE EYES
CREATE NEW
DUPLICATE
DELETE
INPUT DEVICES
OUTPUT DEVICES
RULES EDITOR
INPUT
PROPERTIES
FACE
DETAILS
AND
OUTCOMES
OUTPUTS
PROPERTIES
DETAILS

Methodology

Participants:
Our first participant was one of our team members, and was selected because we wanted to do a 'dry run' before testing anyone else. Our second participant was a SIAT student who is specializing in game design and is currently employed in a professional game design position as we wanted a designers prospective.

Our third participant is a programmer for Electronic Arts (EA). Our fourth participant was another SIAT student who has focused on programming. Our fifth and final participant is a SIAT student focusing on Media Arts. We were hoping to target both designers, who might move on to game design and production, as well as programmers who do implementation, as both groups are targeted by the CATGames Gestural and Tangible prototyping project.

Apparatus:
On all of the tests we had 3 computers. One of the computers was used as a camera focused on the participant, the other used as a camera focused on the paper prototype itself. The third computer was used by the note taker to record comments made by the participants and facilitators. Also on hand were a pen and a pencil. The Pen was a surrogate for a mouse for the participant, and the pencil was for the participants to use in lieu of a keyboard when we asked them to input text into our text boxes.

Tasks

There were three tasks that the participants were given:

1. Create and rename a rule; Creating a rule is the starting point of our system, creating a rule sets up a workspace which participants use to build up games from objects and conditions
2. Setting winning conditions with inputs;
3. Choose an output and output to that object when the winning conditions are met.

Procedure

The participants were asked to create a rule for a game. This was intended to contain the following steps:

1. Create a new rule,
2. Select an object (drag it object up to the workspace),
3. Set the properties on the object,
4. Set the details of those properties,
5. Select output object,
6. Set the details of the output

Test Measures

We looked for the following:

* Could the participants see the steps that they needed to take to complete the tasks?
* Do the participants know in which order to take those steps?
* Do the participants understand the drag and drop paradigm of the workspace provided?
* Do the participants understand the terminology used in the prototype?
* Do the participants struggle with any portion of the interface?
* Does the concept of a tangible and gestural prototype interfere with the understanding of the interface?
* Do the participants understand the rule oriented paradigm of the interface?

Results

Four out of the five participants were unable to complete the tasks without prompting, all of the participants were able to finish all the tasks with prompting.

There was hesitation before subjects would start working with the system as it was not immediately obvious that the rules panel was the starting point for tasks and the core of the workflow. The participants were also stalled by the need to drag input devices onto the work area as four of them tried to click objects first and received no response. However, once the process of setting the conditions for a rule in the work area begun it was straightforward, quick, and users committed few errors.

Once users had completed part of the condition setting task, checking the face value of a single die, they were unable to figure out the most efficient way to repeat this process for the second die. Instead of grouping inputs together all subjects opted to recreate the conditions over again for a second input. The tasks were still completed but not in the most ideal way.

None of the users had difficulty with setting the rule output as they utilized their knowledge of the work area from before and applied it to this process. What did come up was a number of comments about the extensibility of the system and support for features that users felt they would need but we hadn't presented (such as saving, loading, combining rules, and if/else control statements).

Discussion

What you learned from the low-fi experiment

1. We never put a losing condition rule in, we have been assuming that snake eyes always comes up.
2. There was no visible feedback for the user in our prototype to show system response to actions like clicking and dragging.
3. There was no contextual information bar to let the user know what they have selected does.
4. Revision history would be helpful.
5. It was not clear that the interface relies on drag-and-drop functions and what sort of functions were available.
6. The gradual introduction of property panels that appeared as needed was effective at focusing users on their tasks.
7. Having the constrained amount of choices in the properties drop-down menu that we created did not accurately show whether users knew what they were doing or not.
8. Our facilitators need to play more of a passive role and allow subjects to explore instead of guiding them.
9. It's important, even at this early stage, to consider colour choices and the salience of interface objects.

What you will change in your interface from these results

1. Add a contextual help panel or status bar.
2. Include a history revision option.
3. Provide visible feedback to all user actions.
4. Rethink the organization and handling of rules including the if and else components.
5. Consider whether we want to allow the user to build with objects first and then have the rules follow.
6. Redesign the labeling within the workspace to be clear, concise, descriptive and consistent.
7. Consider including tooltips, a tutorial, or example rules to teach new users how the system functions.

What the experiment could not tell you

1. Doesn't tell us how much it would help to have certain aspects such as visual cues would work out our paper prototype could not afford them.
2. We don't know what the results would be if there were multiple options in the drop-down menus and how the users would react to them.
3. It didn't show how users would explore the interface since we were answering questions and providing prompt throughout the user tests.
4. We did not have a very diverse range of participants. Most participants had some familiarity with either design or programming. Also, 5 participants can only provide a narrow range of test results and feedback. Further testing may bring more insights into strengths and weaknesses of the system.

Appendix A: Combined Usability Incident Logs

Task	Usability Issue	Severity	Reommendations
Create a rule for snake eyes.	User is unsure of whether to click or drag inputs and tries to click first, which has no effect.	minor	Provide some prompt or visual indicator to show that devices are to be dragged into the work area.
	User is confused by what the differing dots on the input dice refer to.	severe	Change the way inputs of similar type are handled to allow users to pick exactly which input they want.
	User creates a new check for the second die instead of grouping it with the previous input on the work area, resulting in more work than necessary.	minor	Provide some indication that inputs can be grouped inside panels so users do not unnecessarily double-up on identical rules.
	User expects a second input panel to be visible before choosing from the AND/OR/NOT drop-down box, instead of after.	minor	Tie the appearance, visibility, or location of the drop-down in with the second set of rule panels.
	User finds the AND/OR/NOT drop-down box to be very distracting; it draws too much attention to itself.	moderate	Make the drop-down box blend into the work area more than the other panels so users are not distracted mid-task.
	User feels rule editing buttons (Create New, Duplicate, Delete) are hard to see considering their importance.	moderate	Increase the emphasis on critical, frequently-used interface buttons; suggestion by one use to place them at the top of the Rules box.
	User expects to start working with objects first, which contradicts with the Rule-oriented interface structure.	moderate	Consider allowing users to work on a per-object basis instead of with a global pool of rules, or clearly indicating that the system requires users to start working with rules first.
	User is afraid the rule tabs may get confused with regular menus.	minor	Make sure that the rule tabs are distinct and do not blend in with surrounding menus.
	User thinks there should be more complex gestural recognition and recording features.	minor	Provide a robust range of properties and conditions for input devices.
	User is slowed down by a lack of support in the interface to guide through tasks.	moderate	Consider prompts, a tutorial walkthrough, tooltips, or other means to direct users to the next step in common tasks such as choosing an input device after creating a new rule.

Task	Usability Issue	Severity	Reommendations
Create a message output for when snake eyes is detected and rename the rule appropriately.	User does not understand why two separate panels in a row are named "properties".	minor	Eliminate naming conflicts between different panels and increase the consistency of naming conventions throughout the system.
	User is unsure of how to change the name of a rule, suggests three different control methods (click, double-click, right-click).	minor	Support multiple options for editing fields to match conventions established by other development applications.
	User cannot tell how to link together rules to form more complex scenarios..	severe	Investigate interface constructs for combining and organizing separate rules.
	User is concerned that the system would have trouble dealing with negative responses (ie. check if a rule did not happen)	moderate	Allow for the system to check if rules have not been triggered.
	User is confused by the lack of a save/load option.	severe	Take into account the menu structure needed to save/load and perform other similar functions on the interface.
	User feels the need for a help feature to explain in more detail what is currently selected.	severe	Implement feedback as to what the user is pointing at or has selected through a help panel and/or tooltips.

Appendix B: Observations and Notes on Usability Participants

Videos of user testing can be found at: www.sfu.ca/~nwadding/courses/iat334/

Subject 1:

Task: Create a rule to check for the snake eyes condition

- Immediately created new rule correctly.
- User is unsure whether to click or drag on an input device.
  - Tries clicking, no response
  - Tries dragging into the work area next
- Clicked property drop down to find out what options were.
  - Selected face.
- Clicked condition drop down.
  - Selected equals.
  - Types "1" into box.

Task 2: Create a message output for when Snake-Eyes is detected.

- Chooses "and" in between
- User confused by number of dots on dice icon and what it means.
- User dragged second die into the new input slot
- User mocks system for moving slowly
- User finishes this die check the same as before
- User notices inconsistency with drop down boxes
- User confused by second panel named "properties" just like the last one
- User has no difficulty completing task
- User double-clicks rule name to edit it

Subject 2:

Task 1: Create a rule to check for the Snake-Eyes condition

- User knows to drag die to input panel
- User wants faster system responsiveness
- User checks out what drop down box between panels means
- User thinks there should be a second input before he chooses from the drop down menu.
- User completes second input task the same as the first

Task 2: Create a message output for when Snake-Eyes is detected and change the rule name.

- Notes he might click, double-click, or right-click to change the rule name.

Post test comments:

- Thought it was odd that the "and or not" box would be there along with the property window after an input.
- Suggests that the box should not stand out so much.
- Felt that the system well matched previous knowledge with using computers.

Subject 3:

First task: Create rule that checks for Snake Eyes

Notes:

- Looking at interface, isn't any feedback in terms of which elements (except post its may be buttons) not sure what each window does. Team member must explain frames in window (working screen, input/output buttons). Team member points out permanent buttons in Rules section. Tester then presses 'create new' to bring up new options.
- "Is this a visual programming interface?" (Yes).
- User drags die over to Input box for rule one. Drop down menu appears.
- Tester asks if "details" brings up properties (works in reverse -> properties brings up details)
- Good point - > shouldn't bring up details until properties box is interacted with.
- Not sure about flow of interaction (not entirely sure how to use system without our assistance).
- (Tester wants a note pad to make notes for us) -> User suggests that when rule is created, a prompt (visual) should appear to suggest user needs to suggest use of Input device (ie Die)

- On properties box user selects face, only option, which he thinks makes sense given a die. We bring up details box. Details box brings up = > < symbols. User selects = and types one. Correct choices.

- Clicks on the drop down menu below input, AND, OR and NOT appear. Clicks on AND which brings up INPUT box. Drags another dice. Properties appears, puts face, equals, then types 1. Now user has completed rule.

- Prompt user to create output. Clarifies that the user rule generates a rule such that the outcome will be generated. Not sure how user would know to do output after finished input rule. Actually could be in any order. User suggests output should come up that input should come up at same point as output.

- Selects output screen, message, types 'you rolled snake eyes'.
- Next time we do this we should have written instructions for the user so we don't have to keep reminding them what the task is.
- User is prompted to rename rule. From user interface perspective would double click, display 'rule' as highlighted, and can rename as snake eyes.

FINAL WORDS:
Hector "from game designer's perspective, time is quite valuable so if things aren't obvious and they need to go around and start discovering they lose attention, move on to doing other things' 'as a programmer would go around and spend time understanding how to use tool' but from developer's perspective would want it to be more intuitive. Suggests referring to itunes smart tunes and the automator (for mac) and these perform similar tasks.

1 more suggestion: not really present in this interface, but a help box with text, a short description box as to what you've highlighted (selected)

- Asks after Rule is created, how would save? No save button, would automatically save, would record the automatic changes.
- Hector "at least from the programmer's perspective, one thing that might be useful would be having automated snapshots a 'history' -> versioning system

Subject 4:

- Looked around interface, see what options are - input devices, output devices
- What are the rules? User expects a list of properties, do they come up depending on which item I choose?
- Create, new duplicate etc needs to be more visible
- Drags a die onto the workspace
- User is confused; he would start with objects first and assign rules to them
- Objects get assigned properties/rules rather than create rules and assign objects to
- Create new – (computer does a bunch of stuff)
- Some confusion on what computer does
- Keeps clicking on new things as they pop up creating more and more menus
- Have other possibilities on drop-downs for incorrect choices
- Have only one die in input box
- Drags screen to output, task was to have output?
- Rules is almost like a tab, didn't want to confuse with regular file menus
- Click, click, type name
- Click on rule 1, write?

Comments/suggestions

- Different term for rules, make more dominant – create delete etc more bold
- Similar with editor/workspace (WORKSPACE)
- Object based instead of rules based
- Similar rule, but dif concept than rule, then within folders put objects
- how would you, with this system, expect to handle a negative response? How would you output a negative response to the check?'
- under outputs have an else – if this and this then THIS
- how do you link rules to other rules to create a more complex game
- more complex tasks for gestures, for recording them, eg on the wii (tracking)
- recording gestures instead of taking properties, or defining input

Subject 5:

- I'll try using 'create new rules'
- Click input, click input again close drop-down
- Drag die onto input
- Click new drop-down
- Face?
- Terminology to help user know what face means (about what side is facing up)
- I'll make a 2nd rule for the other die (hadn;t been considered)
- Closed up everything i was working on, click back to the rule brings up the old menu
- Drag die onto rule again, select face cuz it;s the only choice
- Dropdown, details, equals - write in 1
- And...another die on 2nd input
- Face, details, equals, 1
- Rule created, task one done
- Deleted rule 1
- Go to output, drag screen, click, click, message, fill in text
- Rename rule? click on rule, text box, fill in snake eyes

Comments/ feedback/thoughts

- Took me a few minutes to think about dragging the inputs and outputs
- Want to press buttons
- siat, media arts major