

• Introduction



This is The Sims, a game that I helped develop at Maxis, which uses pie menus to control the behavior of people living in a house.

• Debugging Tools



(Error dialog pops up.)

Woah! (Reading error message:) Debug assert: False calling portal tree from within a portal. Please ignore. Um, anyway.

This is the development version that has the Edith tool for editing the

characters. I'll show that in a minute. But first:

- Pie Menus



When you click on some object, it pops up a pie menus that shows the options you can do. I've directed this character: Get into the hot tub.



So if I click the right button on her, I'm controlling her. And now I can click on places, I can tell her to go over here. And she'll stop what she's doing, and go over here. Now I can click on the hot tub and tell her to go join, get in the hot tub.



Oh, a neighbor just came over, so I'll click on the neighbor, and I'm presented with "Greet". So, well I have to get out of the hot tub to do that, because... Oh, come on. Anyway.



- Dynamic Pie Menus



The point is: that pie menus pop up, and give you a list of actions that can change dynamically. And pie menus are especially good when the items don't change all the time.

- Pie Menu Change Dynamically for Social Interactions

But that's the design of this game, that when your social interaction with somebody changes, you might have different items on the menu.



They just greeted each other. Since they just met each other, when I click on this new person, we can ask her to leave, or we can complement her.

- Developing Relationships

And that's going to develop a relationship with her. And as I get friendlier with somebody, I will be able to do more things.



So Bob over here, she's quite friendly with. So when you click on him, you get a menu full of different things to do. And this will change according to the mood.



- Kissing and Falling in Love



A little privacy here. I'm going to kiss. OK, yeah. OK that's pretty good. Oh. Oh! Now they're lovers! So better not go kiss the other one in front of her, now.



- Walking Around and Pointing at People



Oh, here's a neat little thing. When somebody's walking, you can point at them, and it slows them down, so that you have a chance of catching them, because otherwise they run out of under your cursor too fast.

In order to do social interactions, you need to be able to pop the menu up when somebody's walking. It's kind of like the tilt of a pinball machine: it gives the unlucky person you're pointing at a little less speed, so that they stay under your cursor a little longer, and you can pop up a menu on them.

- Editing Tools

So anyway, the editor for the game: The users have editing tools for architecture, placing objects, and changing the terrain. But there are more advanced tools that the developers use to add new objects to the game.

- Object Placement, Architecture and Decoration



These easy to use tools built into the game for players let you pick objects up, and move them around, and throw them away, and buy more objects, and then place walls. You're not allowed to place a wall in certain places.



Then put a window in it. And then wallpaper it.





- Gridding



These have to be really easy to use. So it has a kind of chunkyness, a lego-like grid to it, where things click into place. That makes it much easier for kids to use than a full-freedom 3D graphics editor would be.

- Placing and Orienting Objects



So when you pick up an object, you can place it. And then while you're placing it, you can orient it by a pie menu like gesture. Say I have a chair, and when I place it down, I press the button down, now I'm holding a spin tool. As I turn around it will spin it. I can pick it up and move it, and also when I place it: turn it. This is really convenient.



- Placing and Orienting Consecutive Objects



For example, if I had a dining room table, and I wanted to put some cheap chairs around it: I have to press down and move towards the table to orient the chair, and now the next chair is going to be oriented the same way.



And now I'm going to press down and orient it again, and put the next one down. And then press down, orient it again, and put the next one down. Press down, orient, and then the last chair.



So it makes it easy to do this nice quick gestural style.

- Musical Chairs

Let's see: I could move them all.



We're going to play musical chairs, so we might want all the chairs rearranged like that. Just drag the direction you want it to face when you put it down. Here's the musical chairs scenario. Gotta get rid of that Bush. You know, you could just press down on something, and drag it like that to orient it.



- Multi Tile Objects



Now the multi tile objects are a little unwieldy, because they pivot around a tile. But it will get red and tell you that you can't place it there, when it's in a bad place.

- Clicking and Sliding at Valid and Invalid Locations



So here's a neat thing: See how it clicks into place when it's in a good place? But when it's in a bad place, it slides. You don't have that nice stickiness. But when you get into a good place, it clicks right in there.



So there's this feeling of slipperiness when you're between, and you can kind of tell just by moving that you're not in a valid location. But then you get this nice chunking. That tends to make it easier to use for people with low dexterity.



- Downloading Plug-In Objects



The game is able to download new objects and install them at run-time, and plug them right in and they interact with all kinds of other objects that are already in the game.

- User Created Objects

And also users are able to create their own objects by cloning game objects that already exist, and changing the graphics and the descriptions with a program called Transmogriifier.

- Character Skins



And additionally you can paint the skin of your character. The characters are 3D skeletal character meshes, that have a two dimensional color image mapped onto them, that's unwrapped, that anyone with a 2D graphics editor, like Photoshop or Paint Shop Pro, can easily paint.

It's harder to make a 3D character out of polygons, but you don't have to do that.

Just simply paint your face on the head or your clothes on the body.

- Transmogrifier

The Transmogrifier program allows you to export all the images of the object at the different angles and zooms as bitmap files that you can edit, instead of editing it as a 3D object.

- 3D Studio Max Object Exporter

There are development tools for making 3D object sprites from within 3D Studio Max, and then rendering them out to Z-buffered sprites.

- 3D Studio Max Character Animation Exporter

But the characters are authored with Character Studio on skeletons with deformable meshes. The system is based on Ken Perlin's layered animations that you can blend together.

- Walking Engine





The characters are autonomous, and just walk around, and there's a higher level walking engine, that follows a path. Umm, show paths. Well anyway, it's not very interesting.

But anyway, they walk down the path, and go to the object. And the people know how to walk around, but they don't know how to use the objects.

- Object Animations



The objects know how to make the people use themselves. So when you download a new object, it has with it some character animation, of how to manipulate it and interact with it.

- Object Programs

It also has some programs that control the person's behavior, how it effects the environment, and everything else.

- Object Advertisements



The menus are generated from the advertisements that are in these objects, that say "I can do this" and "I can do that", or "You can do this with me" or "You can do that with me".

- Preconditions

But they depend on the mood and the personality type, and they can be inhibited under certain conditions. So there are little pieces of code that get run, that figure out what behaviors are appropriate for each person and object.

- Subject Oriented Programming

This is described by some people as object -- not Object Oriented Programming, but Subject Oriented Programming.

- Objectifying People



People are objects, so when you click on another person, the items that you get in the menu depend on your mood, and your relationship with that person.

- Different Strokes for Different Folks



I don't know this person, so I can "Ask to Leave", because they're a guest. I can "Joke", "Talk", or "Call Over" because they're in a different room.

"Call Over" wouldn't be there if they were in the same room, and "Ask to Leave" wouldn't be there if they lived in the same house.

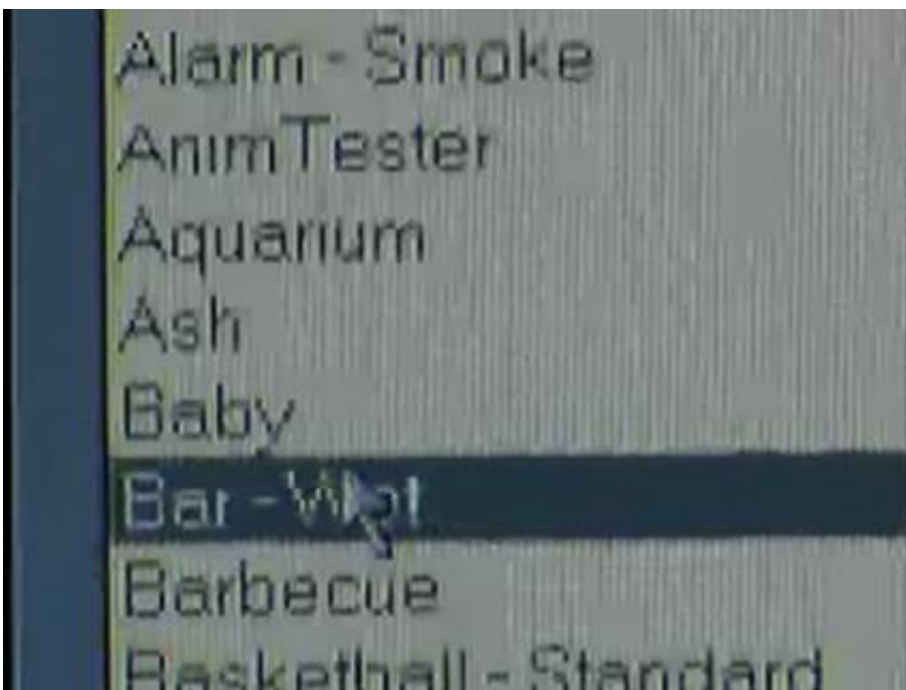
- Edith Sims Programming Tool



There's a development tool called "Edith", which is for "Edith Bunker", the first character in the game. And the "EDIT House" acronym.

It runs in a window, and uses all these control panels here to edit the properties of everything in the game and browse things.

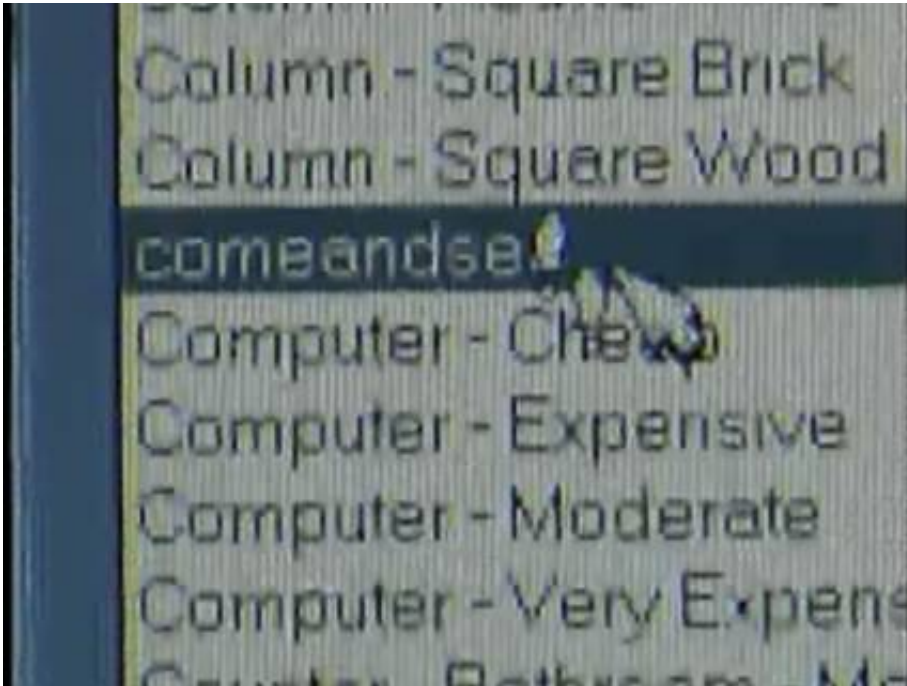
- Listing Loaded Object Classes



So for example, I could look at all the classes of all the objects that the game has loaded. We just have all kinds of things.

- Special Objects

There are special objects for character interactions, and sort of invisible things. For example, you can download a guinea pig pet, but ...



Here's the "Come and See" object for when something exciting happens, it brings people over like when there's a baby.

- Viruses

You can download this guinea pig object, and if you don't care for it it will make this secret hidden virus object that gives you a cold, and has with it the animations and sounds of coughing and going (cough cough).

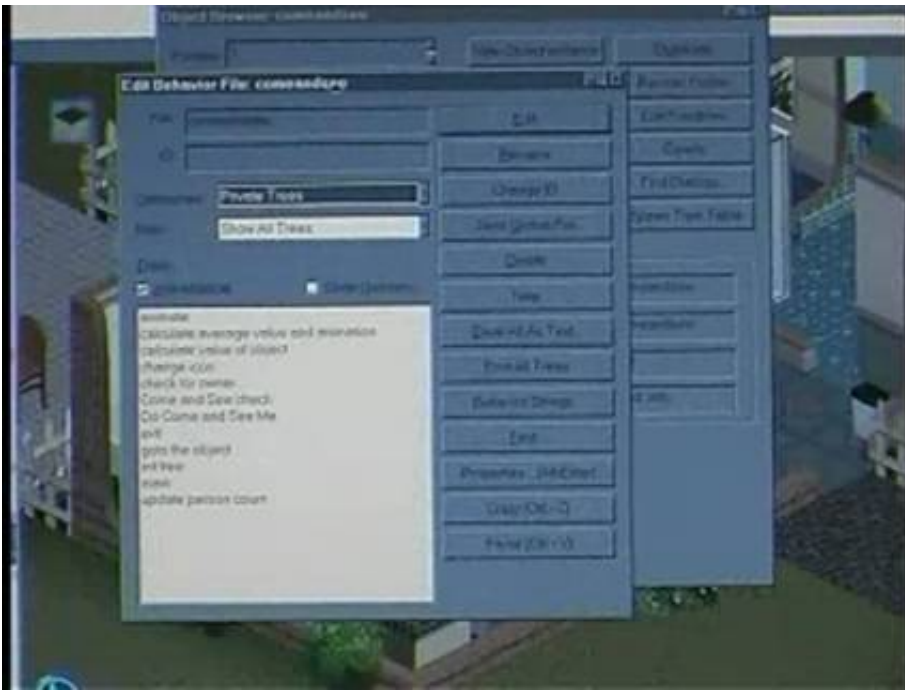
Every one in a while it will just interrupt what you do and cough, and it will be bad for your health. And if you don't get enough sleep, you have to get sleep to get rid of it.

This little program is literally a virus that runs and lives in your household and in your characters, and your characters can spread them to the neighbors, and they'll bring them home to their families.

- Programmable Plug-In Objects

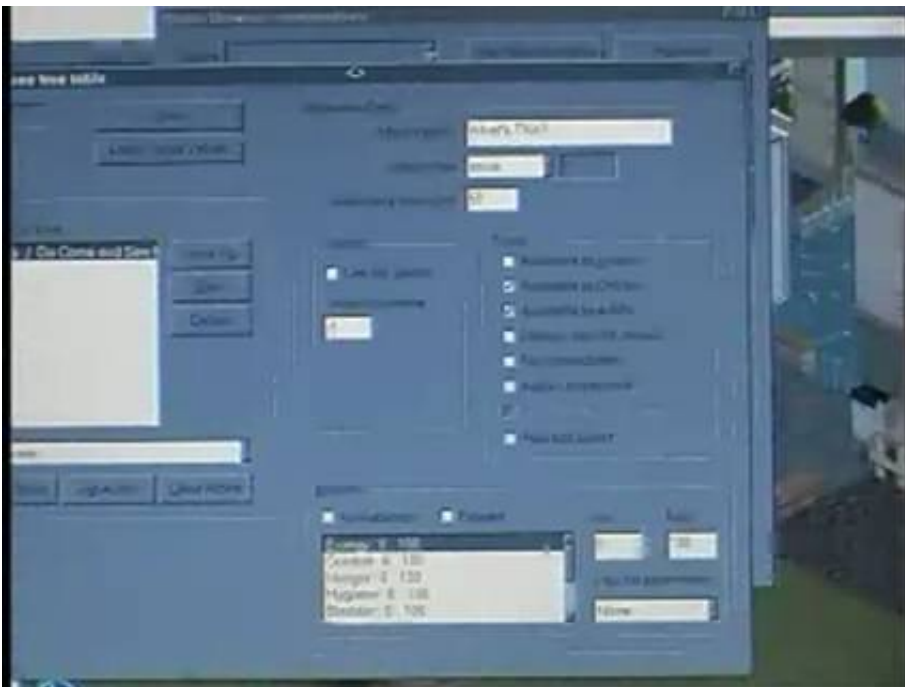
Anyway, there's quite a lot of interesting potential for what new plug-ins additions can do because of this programming languages.

- SimAntics Visual Programming Language



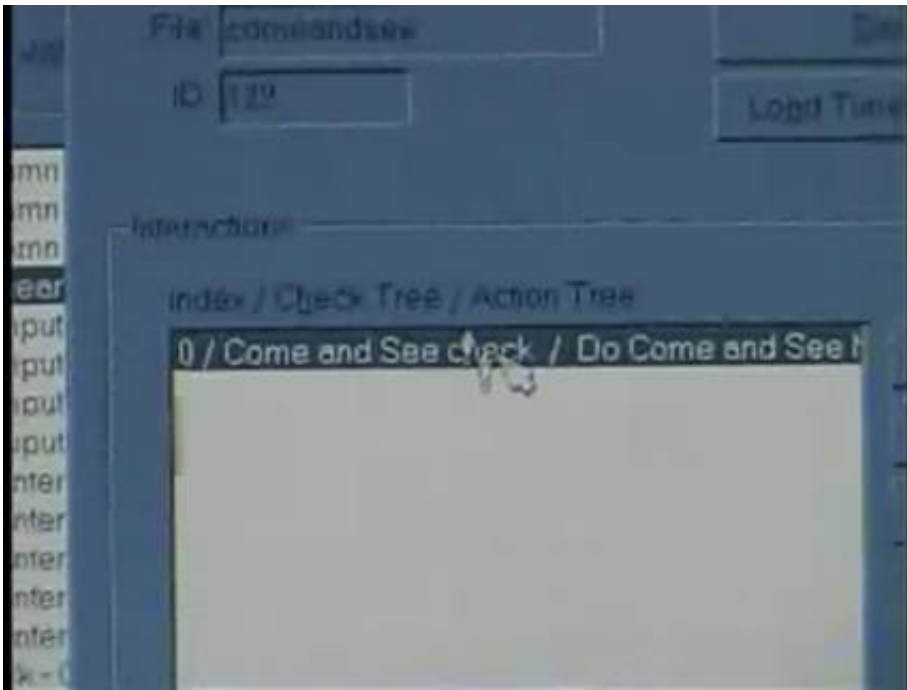
There's a built-in visual programming language called SimAntics, that is a control flow decision tree type of language. So the "Come and See" object, we'll look at that.

- Tree Tables



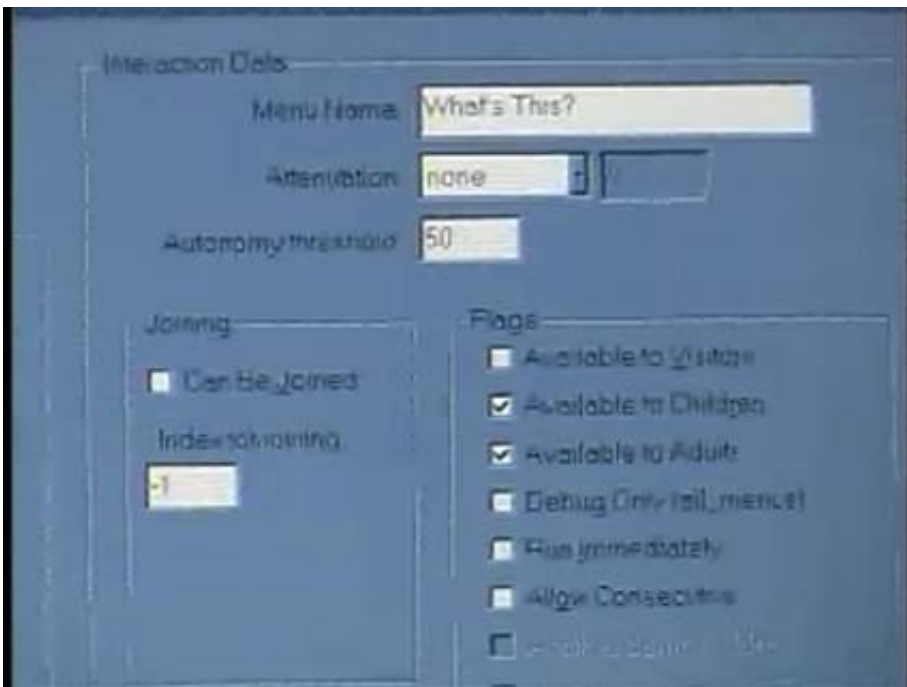
First of all, we'll look at its tree table, which the menu is generated from.

- Advertisements



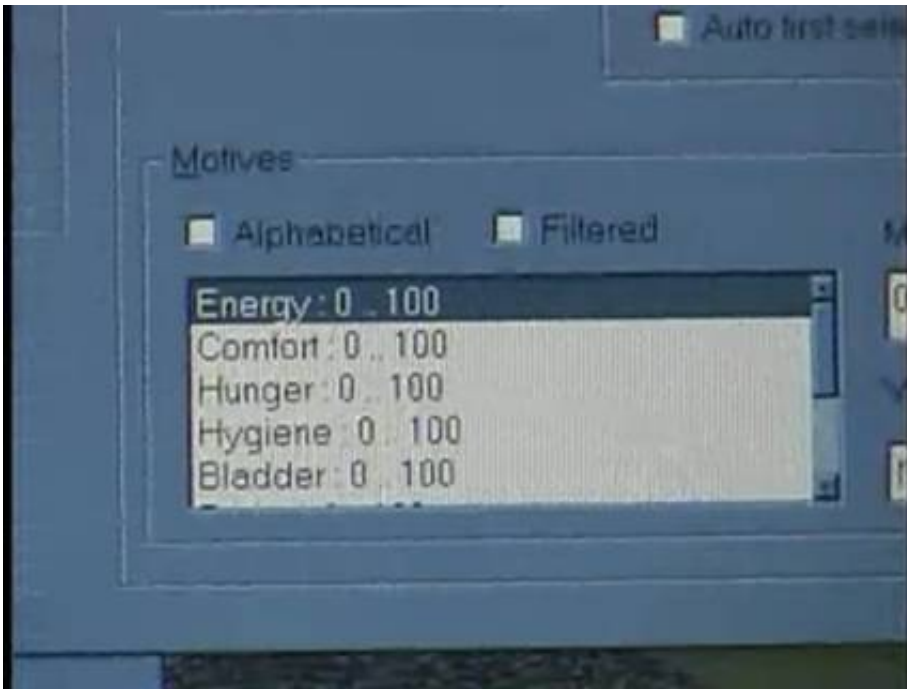
It has an advertisement, and the advertisement is "Come and See". And there are all kinds of parameters for each one of these ads, including the menu name that you get when you click on it, the attenuation, which is how close you have to be, to be attracted to it.

- Advertisement Parameters



Autonomy threshold, whether people will decide to do it themselves or you have to force them to do it. Joining: There are multi person interactions like campfire or hot tub that you can join, and then get personal interaction and score points that way, so to speak. Whether it's available to visitors, available to children, adults, debug only, run immediately, allow consecutive, auto first select.

- Motive Effects



And then there are the motives that effect how people react to this. So basically, things advertise that it helps you with a certain motive, or any number of motives.

- False Advertising

And there could be false advertising, like dangling a carrot in front of a donkey. For example, the food chain is implemented as a pretty amazing Rube Goldberg device.

- Food Chain





See, they're eating right now, but what it took to get them to eat, was a long series of interactions one by one.

So I'm going to tell her to go to the refrigerator and serve... Or I'll tell him, yep, ok, he's going to serve dinner.

But he is currently eating, and talking. That's a social activity that they do together. Even though nobody's sitting, because, I don't know, bad chair?

- Starting a Short Food Chain



First he has to put down what he has, and he's kind of sloppy, so he puts it down on the floor. Now he goes and executes in the refrigerator the "Serve Food". Now he has his food.



- The Next Link in the Food Chain



The food's telling him, "I need a place to sit down". This tabletop advertised that.

- A Short, Unsatisfying Food Chain: Canned Food

The thing is, he wasn't very good: He went straight to making a low quality canned food right out of the refrigerator.

- A Longer, Yummy Food Chain: Appliances

But if he had the cooking skills and the appliances, they would be advertising:

"Hey, I'm a food processor, I can make your food better by using me", or "I'm a Microwave".



But they do it in a sequence called the "Food Chain". So I'm going to go and do a little shopping in appliances, and get a food processor.

- Bad Feng Shui as Comic Relief

Now this is going bad Feng Shui. He's going to have to run around a whole lot to do this, so it will be comical.

We will just put another table outside to frustrate him. Here's the refrigerator, and this will demonstrate the importance of laying your kitchen out carefully.

- More Links in the Food Chain



Then after you do food processor, you're going to need a sink and a countertop to chop it, and then a microwave. Cheat to get some money. And then eventually a trash can to throw the refuse out in, or a dishwasher or sink to wash it in.

- Eating off the Floor



Ah! Somebody put a serving plate on the floor. So he just went and picked up some food, and he's going to be penalized for eating off the floor. But basically, let me try to get him to do the food chain, and we're going to serve dinner. And I hope he doesn't just put it on the floor.

- Starting a Long Food Chain



So he puts down his food, and he goes over to the fridge. He's taking something out of the fridge. He may not be very adept at this.

- Preparing Food with the Food Processor



OK, so he's using the food processor there. That's good news. This may not be so bad.

- Cooking Food with the Microwave



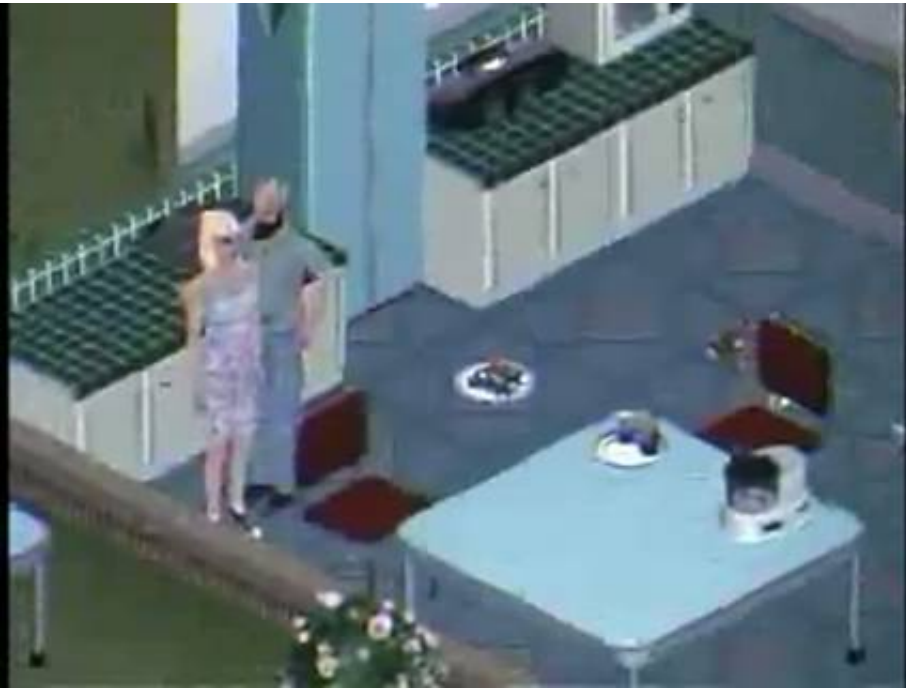
OK, he puts it in the microwave. Now once it's all done, he's going to need to place it somewhere to serve it.

- Serving Food on a Tabletop



So these tables are advertising the service of being able to have things put on them, unless they have something on them already.

- Routing Failures Effect Feng Shui



(A hot babe walks right up to him and prims herself.) OK, she's in the way. He's stunned.

So that was bad Feng Shui there. That kind of amplifies the penalty of having bad Feng Shui, the way they get stunned.

- Swarming Agents in a Feeding Frenzy



Now they're all going to swarm over, and now go to look for a place to sit down. Which also involves interacting with the chairs. So if they don't find a place to eat, or are just stupid, then they might just stand eating. The language has all these decision trees. She found a spot, and she's going to put it down there and adjust the chair.

- Calling Chairs Subroutines to Sit



So the chair was kind of called as a subroutine, to route her over to the chair and then sit her down in it. Chairs, you can route behind them, you can route to their left, to their right, or in front of them. And then it has a different animation to get you into the chair from any of those.

So you can have them crowded around the dinner table and still able to get in. They're very complicated.

- Running Out of Batteries



(A window pops up warning the laptop's batteries are about to expire. Cut to



next scene.)

- Back to Edith



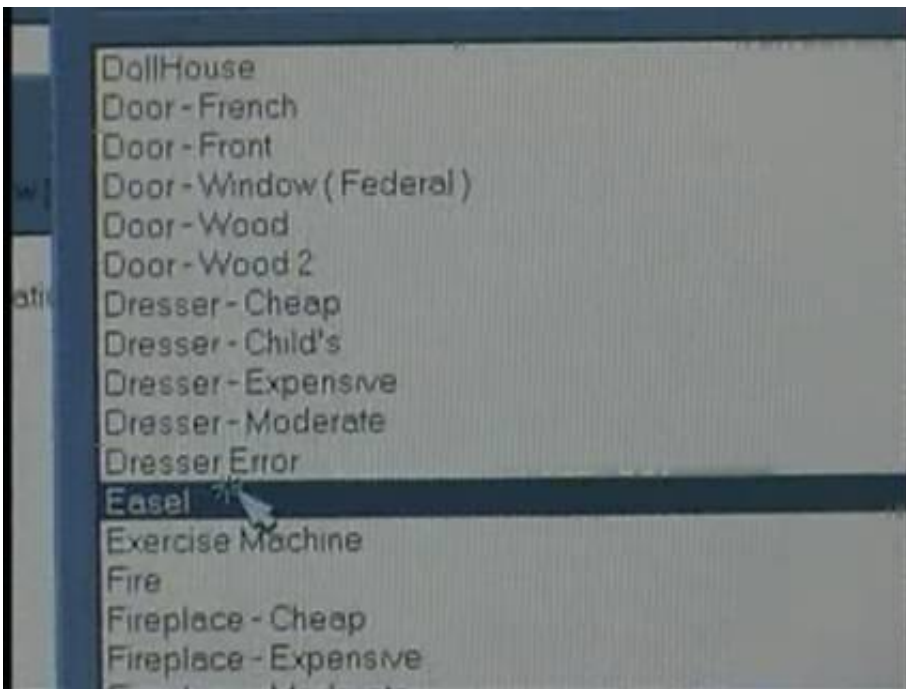
OK, now back in Edith, which is running at the same time as the game, as part of the game, with its tendrils deep within the game.

- Class Browser



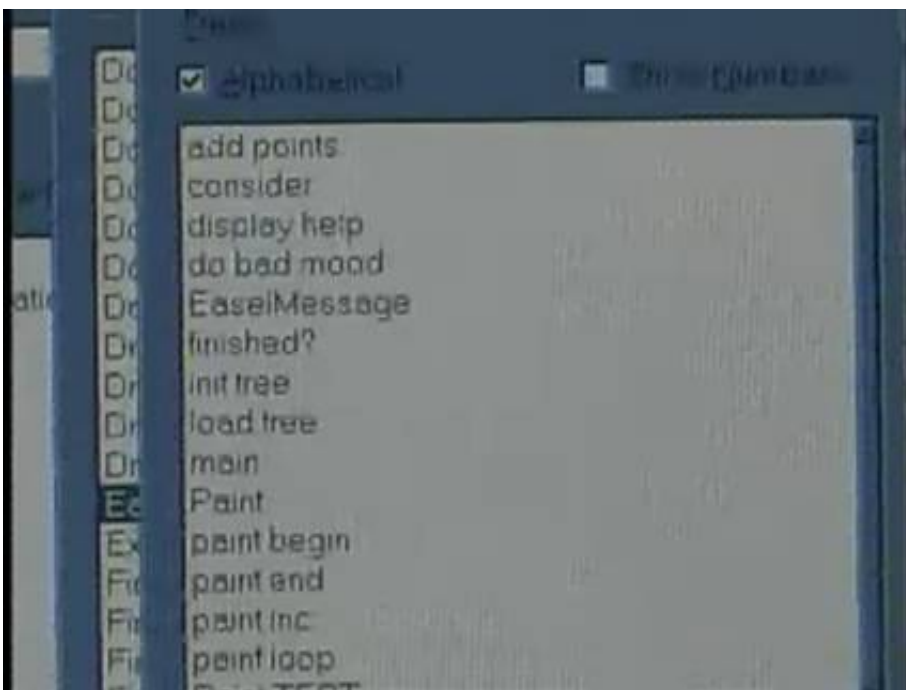
We can look at the classes of objects. If you double click on a class... Let's find an interesting one, let's see...

- Examining the Easel



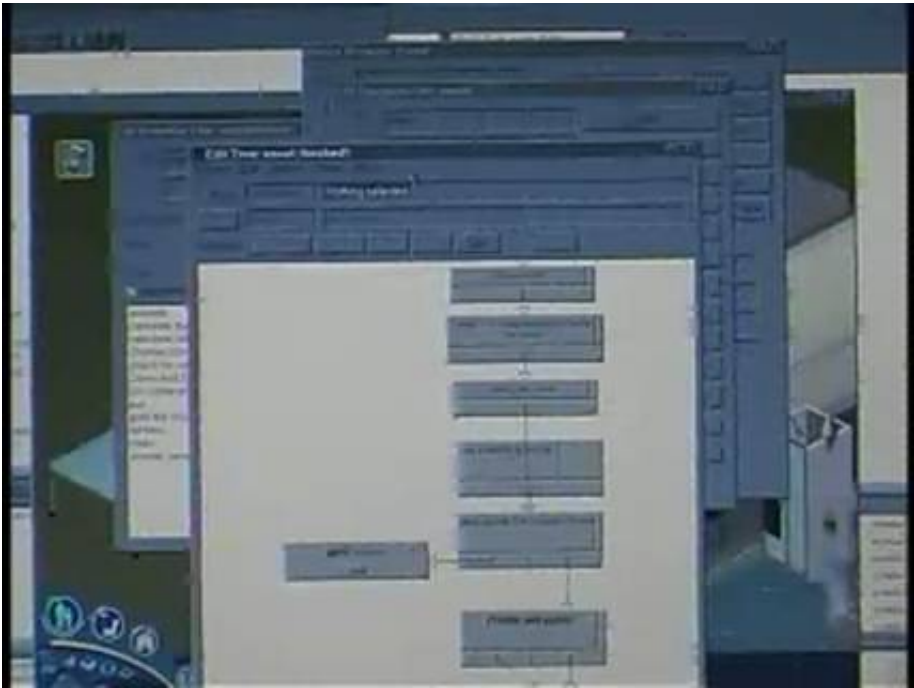
Right, the easel. So we double click on it, and this is the browser for that class. It's got categories of functions that are available, globally or private. We can filter how we see them.

- Objects have Interactive Functions



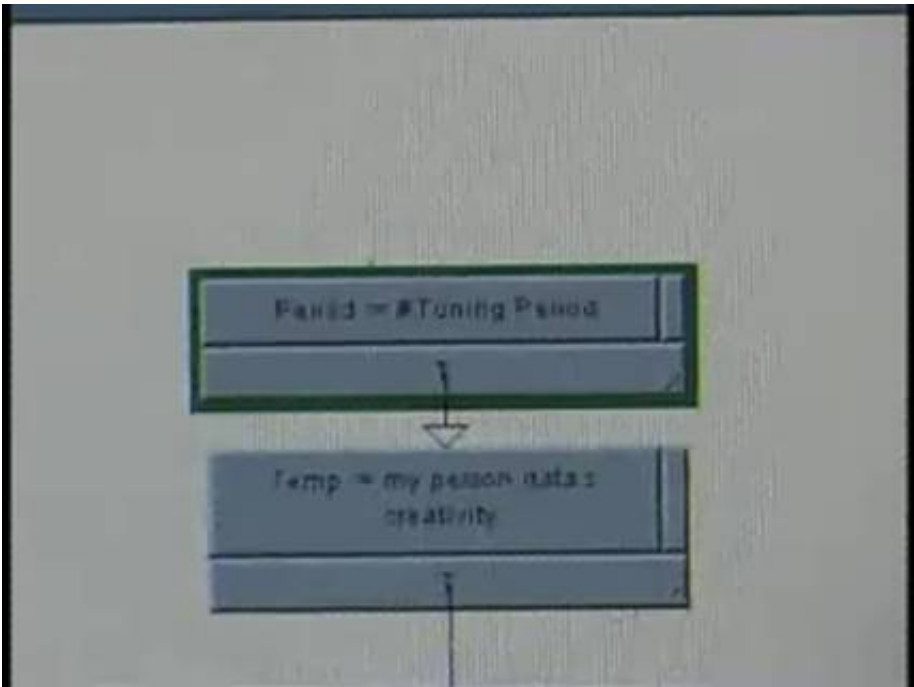
But basically, this easel come with a bunch of functions, to make it interactive and make people able to use it. Here's a function called "Finished" that figures out if you're done painting.

- SimAntics Visual Programming Language



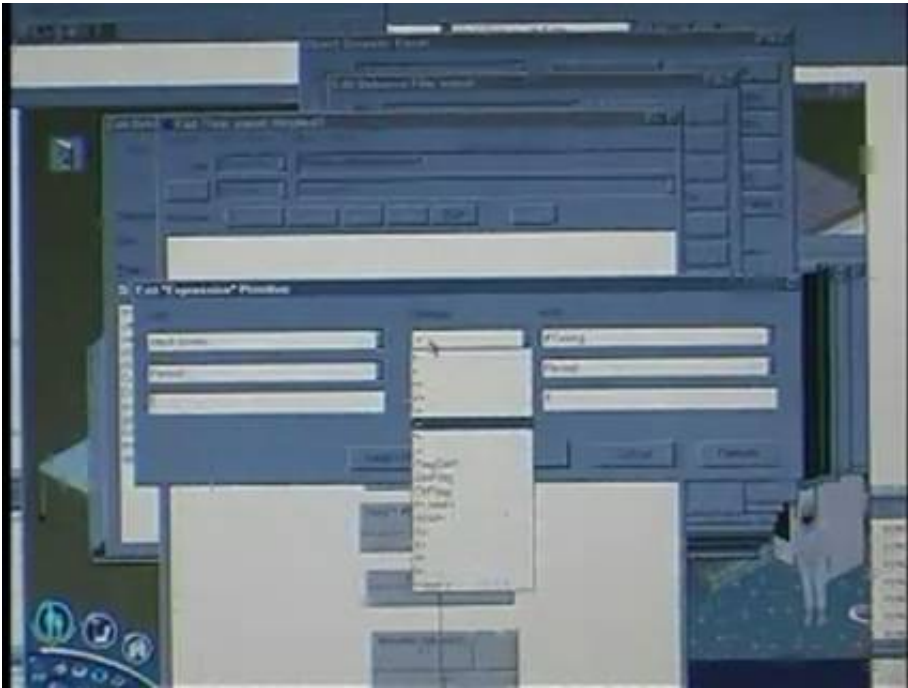
What this is, is a SimAntics diagram editor. It's the boxes and arrows language that is like a visual control flow. It's very good for spaghetti code, which is just perfect for what it's needed for.

- Making Decisions and Keeping Score



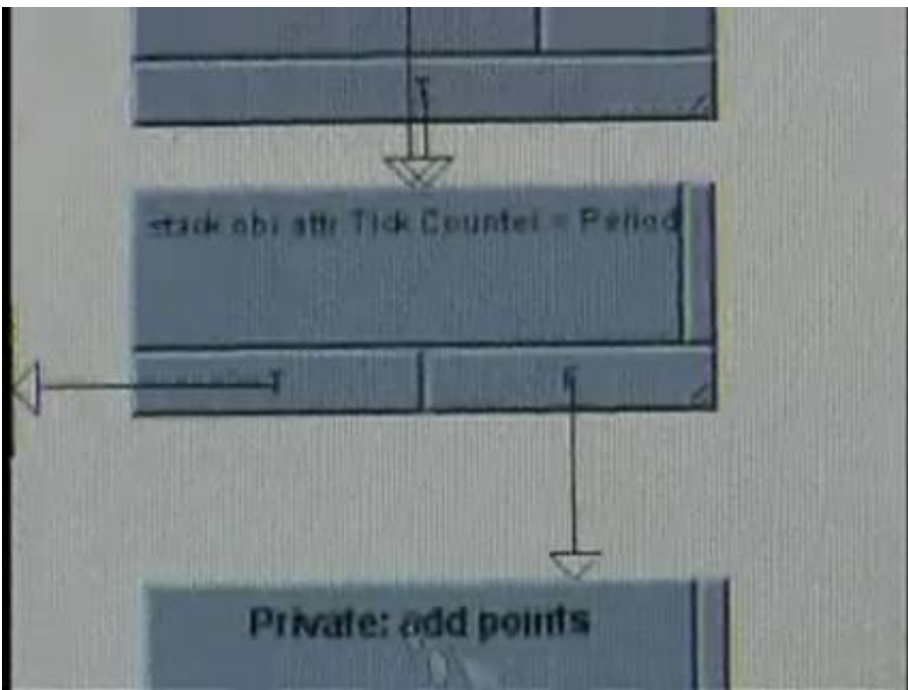
So you can do comparisons and set variables with these little boxes, and double click on them to configure them.

- Special Purpose Primitive Functions



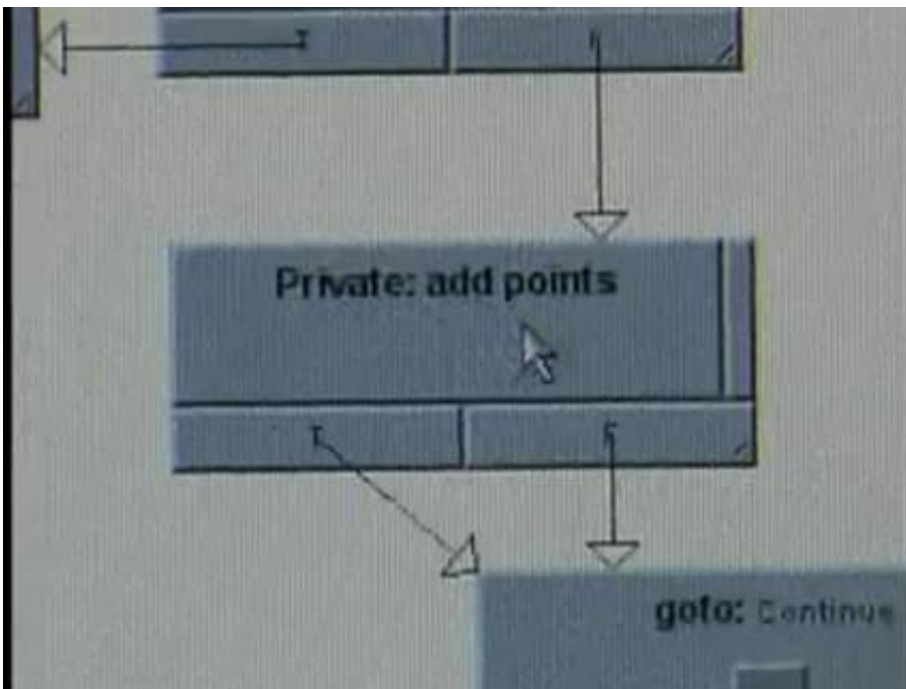
And there's a whole bunch of different special primitives for taking one of those things, and doing this thing with it, to that thing.

- Virtual Reality Machine Language

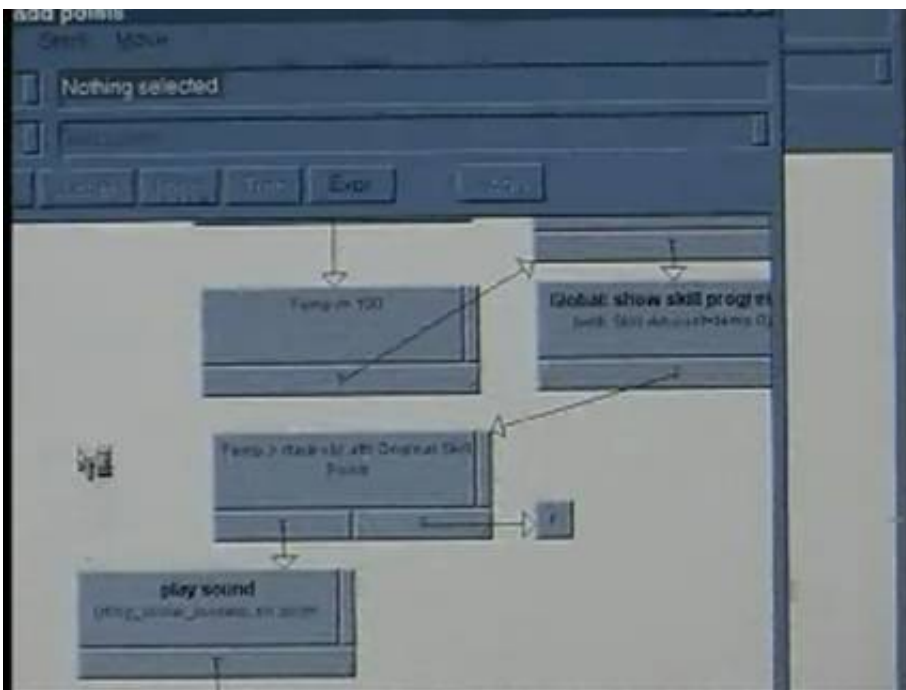


The game has this whole virtual machine, this model of what objects are, that is very unlike anything else I have ever seen, but is designed to make it work the way it does.

- Browsing SimAntics Subroutines



So you have subroutines, this is pretty standard. You can double click on this private subroutine "Add Points", and bring up another editor.



- Calling SimAntics Primitives

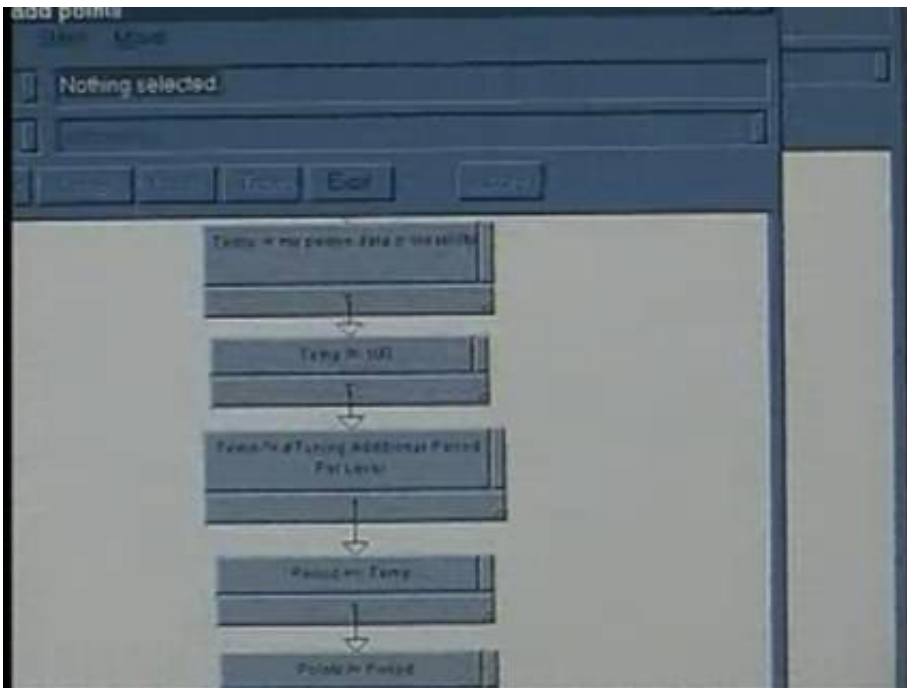


are tools that are geared toward solving problems, like making the painting, or making a certain type of game within a range. Making a parameterized object, and then we can make simple tools, that tweak and put new content into an object like a Jukebox.

- SimAntics is Fragile and Tricky

But programming Edith is really hard. This would unfortunately break the game very easily if anybody made any mistake, and there could be a lot of subtle mistakes.

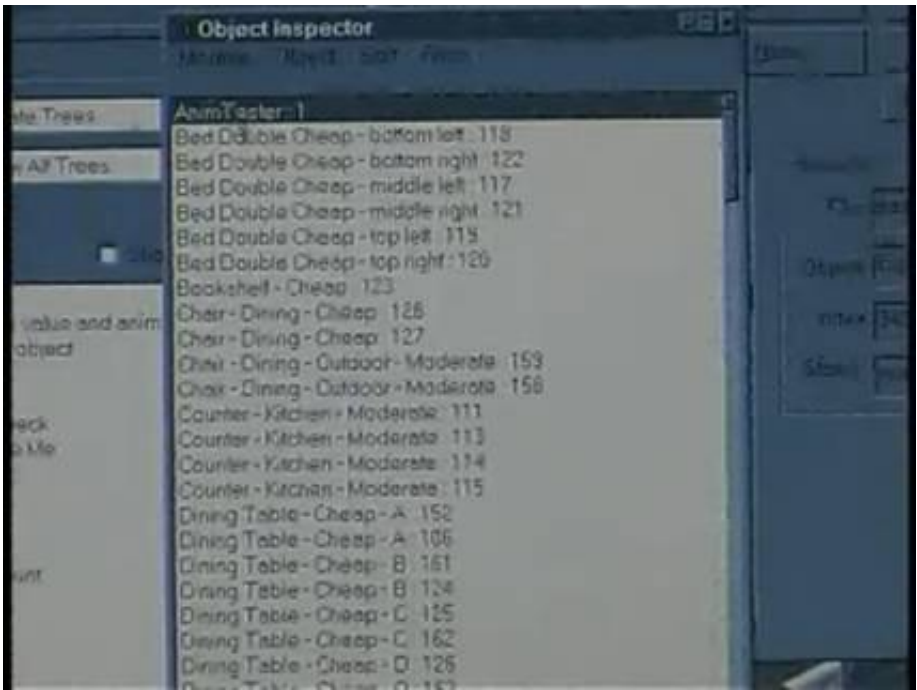
- Hunting for Easter Eggs



Nevertheless, SimAntics code is very interesting to look at, because it defines how every object behaves in the game.

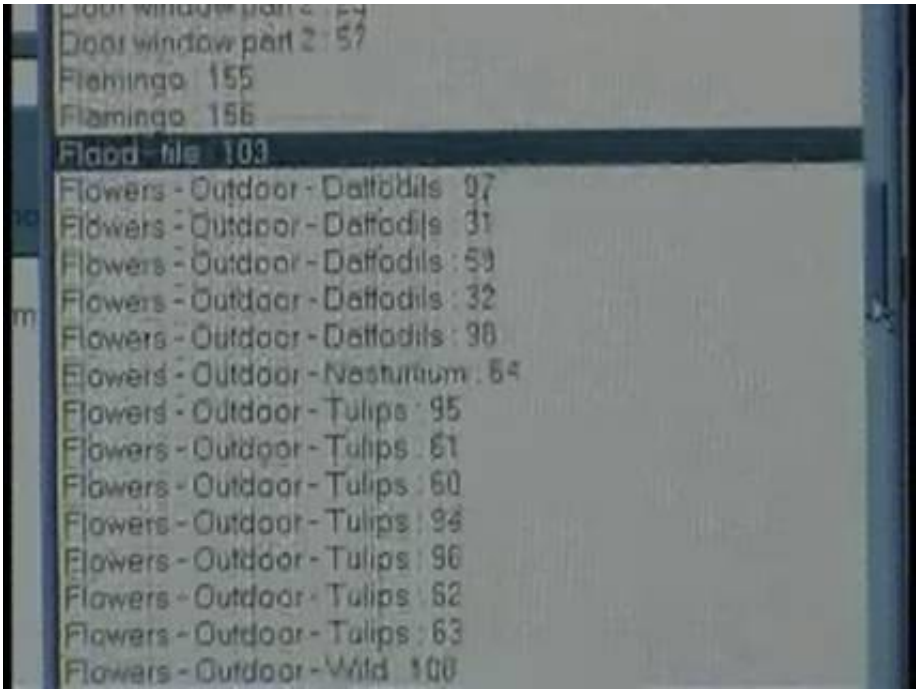
Who knows, there may be all kinds of easter eggs in here. You have to just scroll over it and look at it, or print it out on a giant piece of paper.

- Module Inspector



There are all kinds of special editors for things that are in the game. Of course there's one, the module inspector, that shows all the running objects.

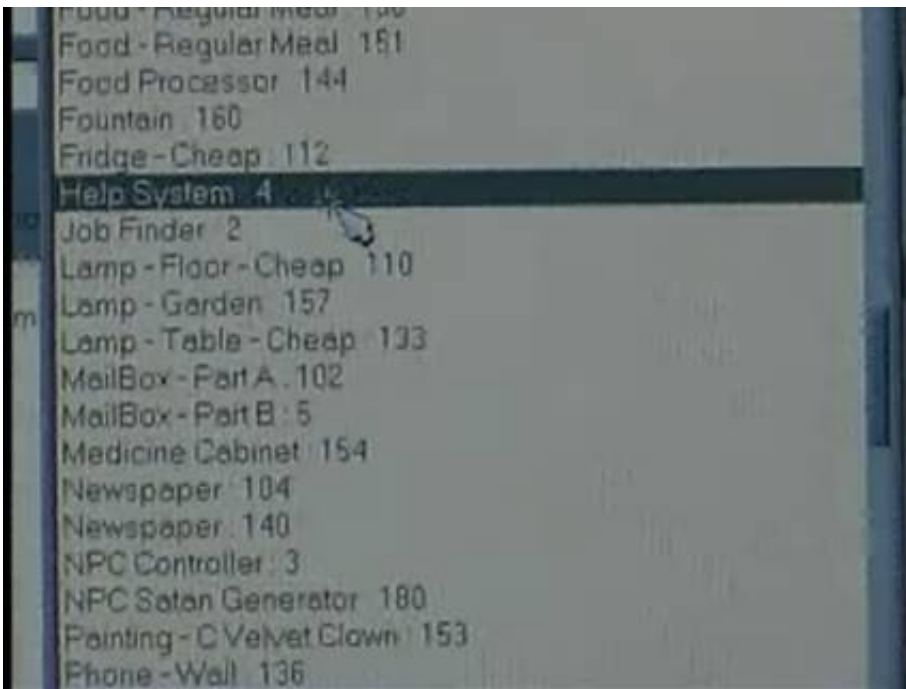
- Live Objects in the House



The "Bed" is broken up into all these separate tiles, that do their different things. There are some "Chairs", and the "Doors". Evidently there's a "Flood" somewhere in the house. Probably next to the "Shower" or some water thing. There are a bunch of "Flowers".

- Invisible Magic Objects



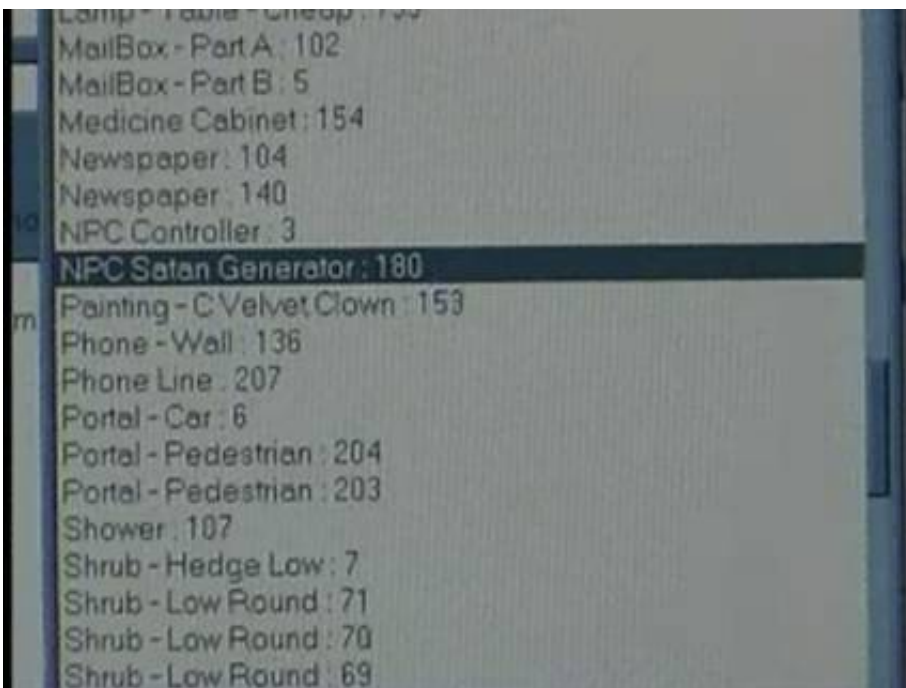


There's a "Help System". There's all these invisible objects that do their little things. The "Job Finder".

The "Mailbox" is a very important object. The whole game breaks if you don't have a mailbox, because it does a lot of scheduling and stuff like that.

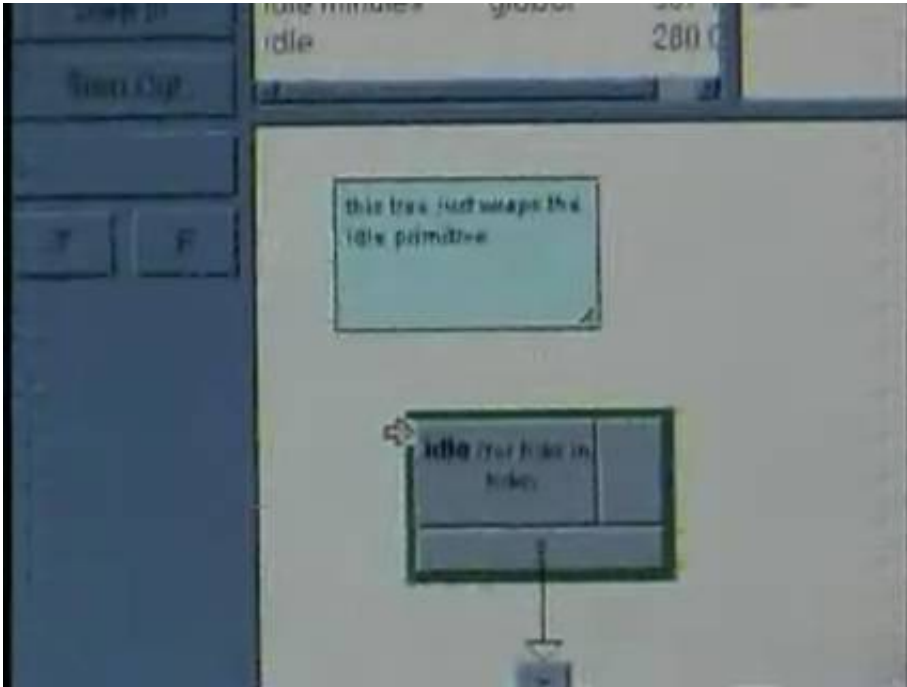
And the "Non Player Character Controller".

- Satan Generator



Oh, this is a very strange thing. The "Satan Generator", let's see what that is. Oh, my. Umm... (reading comment) "This tree just wraps the idle primitive." So this is kind of the main loop of the "Satan Generator". Let's see, where did that go?

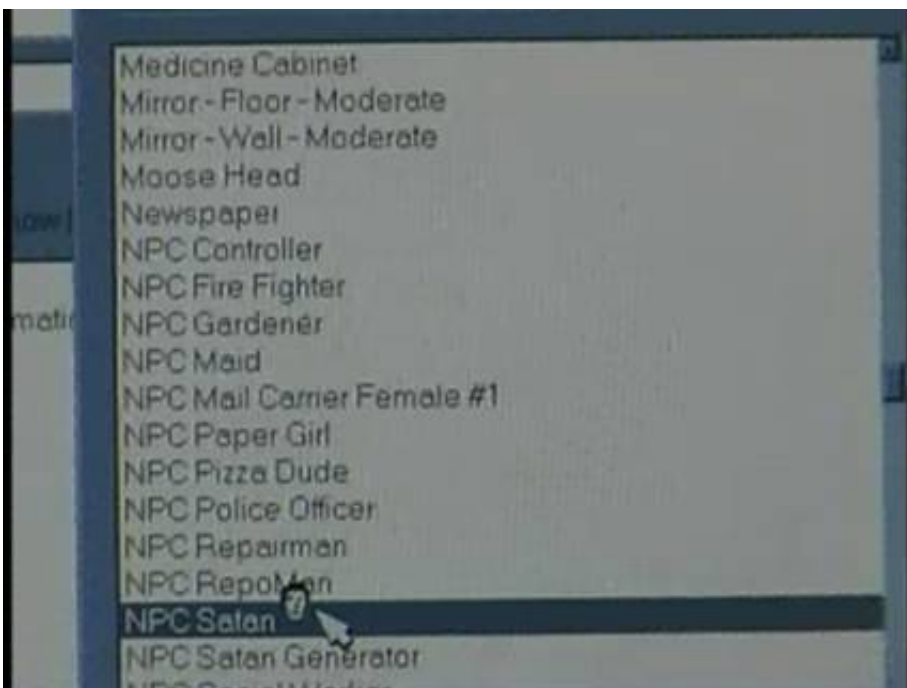
Maybe we weren't supposed to look at that.



But what that does, is that it waits until everybody in the family is depressed. And then it generates Satan.

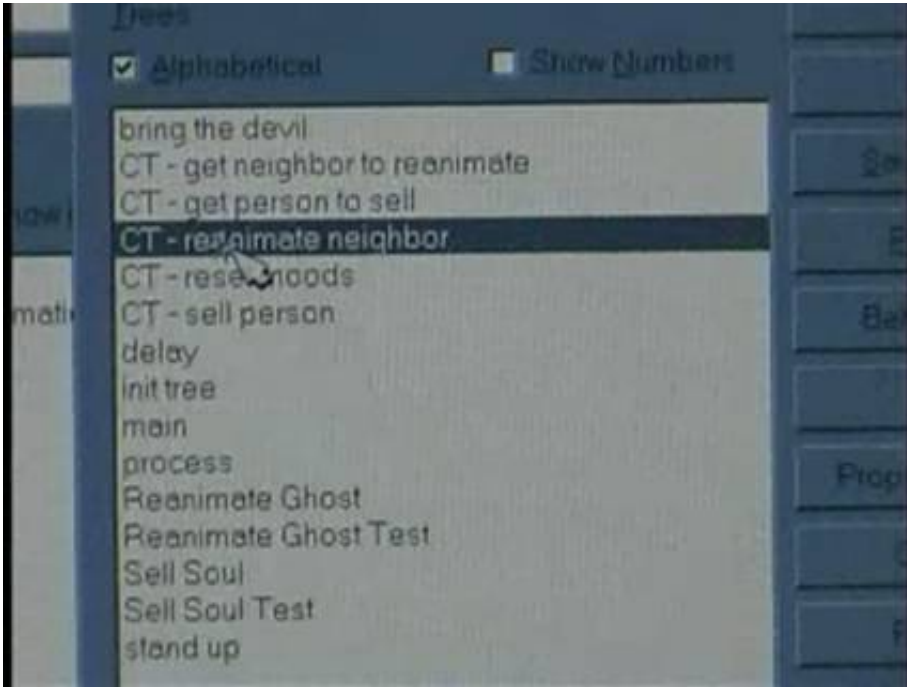
- Looking for Satan

And if we look... We don't have a "Satan" alive right now. But let's see, see where he is here.

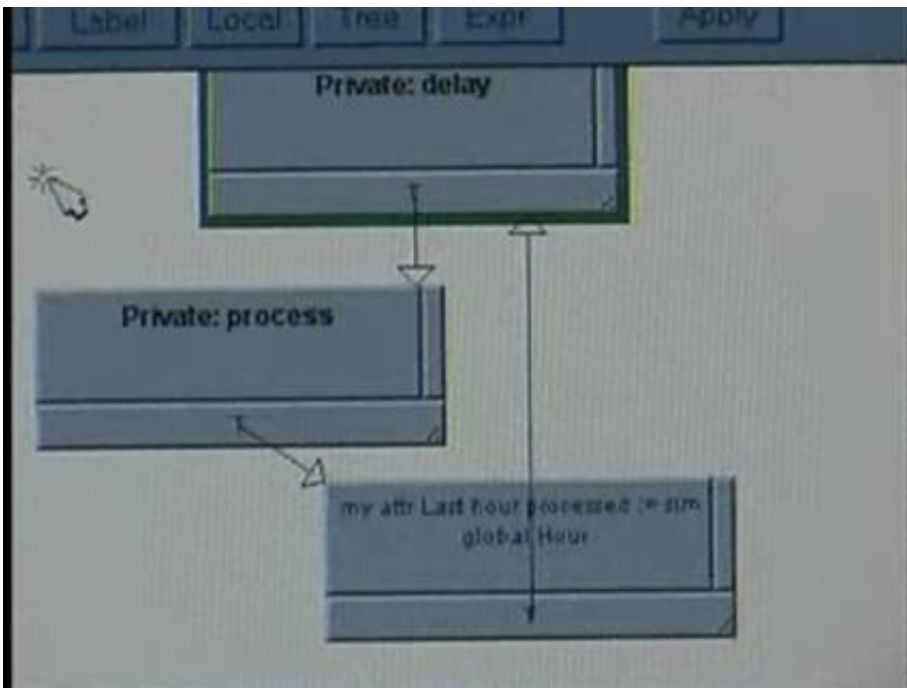


We can take the class browser, go through all the classes, and look at what they are... Satan. NPC Satan? So here are the users, those are the people. And there is "NPC Satan Generator", and "NPC Satan".

- Inside the Satan Generator

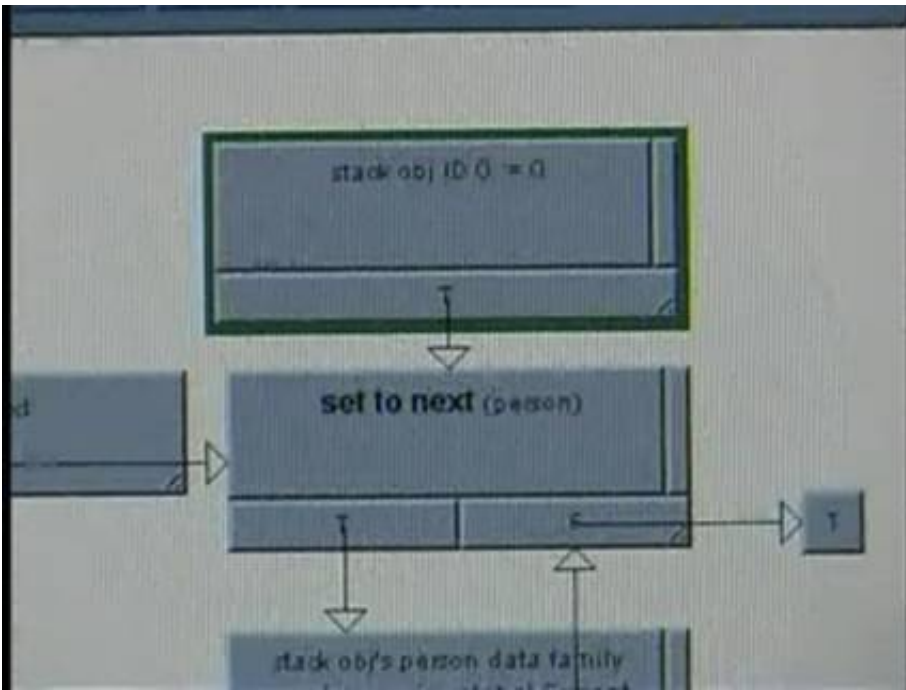


Basically, if we look at the generator, its main function... Umm... Oh, my. Umm... Maybe Satan does a lot more than we knew. Umm...



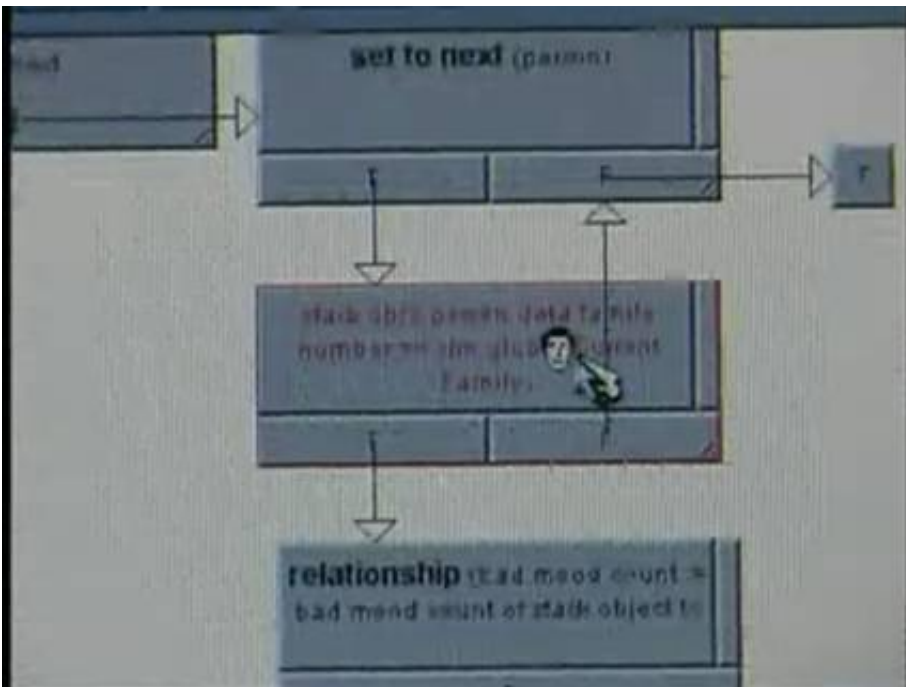
Its main function: sits around, delay, process. And my attribute "Last Hour Processed" gets "Sims Global Hour". So the process is the interesting one here. We'll just climb up here.

- Single Stepping through the Satan Generator



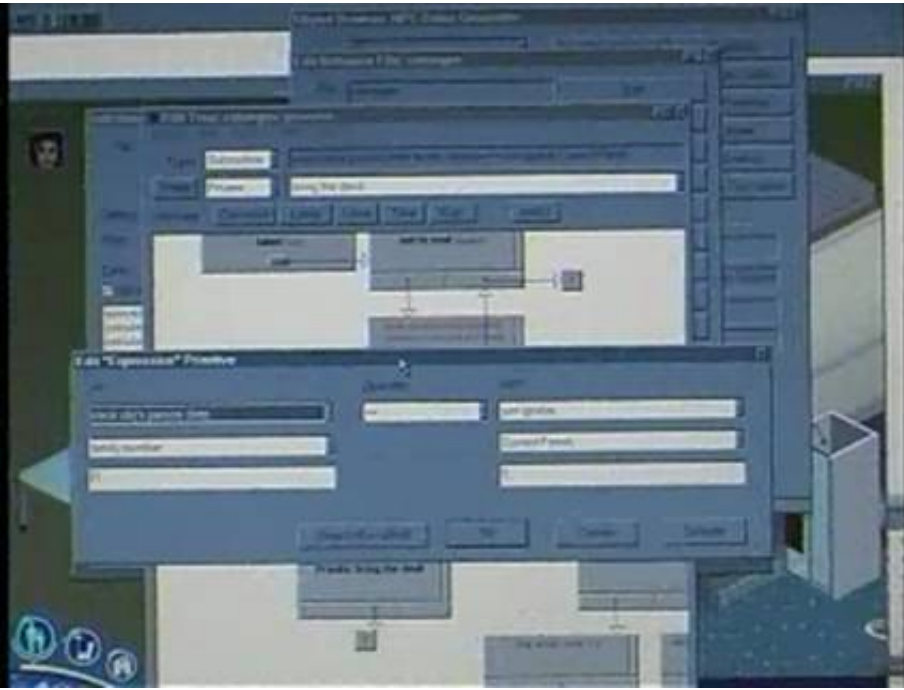
Basically, it's setting... This is actually more like a machine language than a high level language. So it's setting these registers, and kind of the pointer in "Stack Object".

- Satan Generator Searches for its First Victim

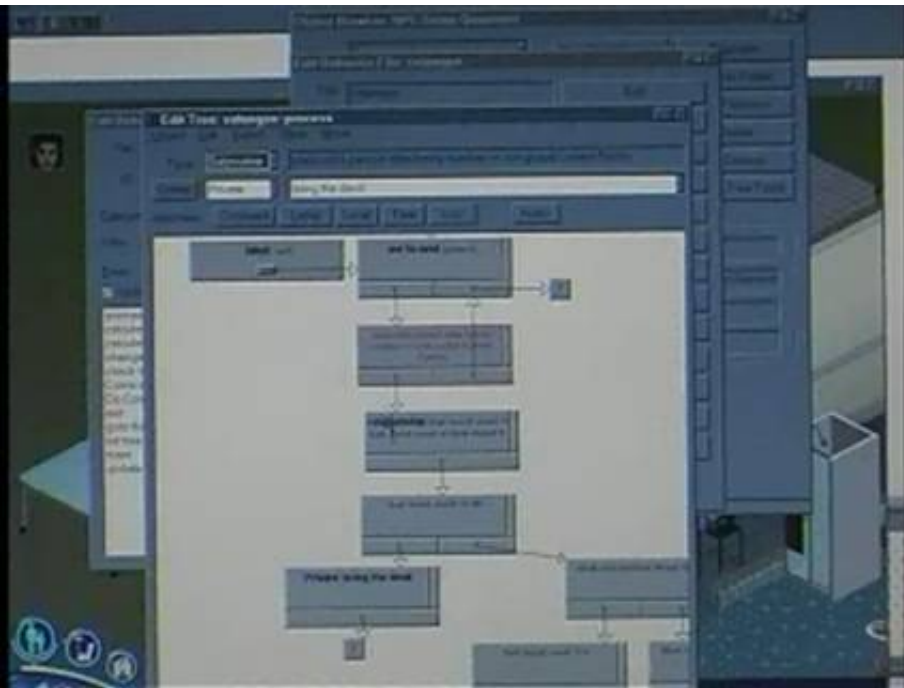


"Set to Next Person" to do a loop. I'm going to loop over all people. Now the "Stack Object"...

- All in the Family



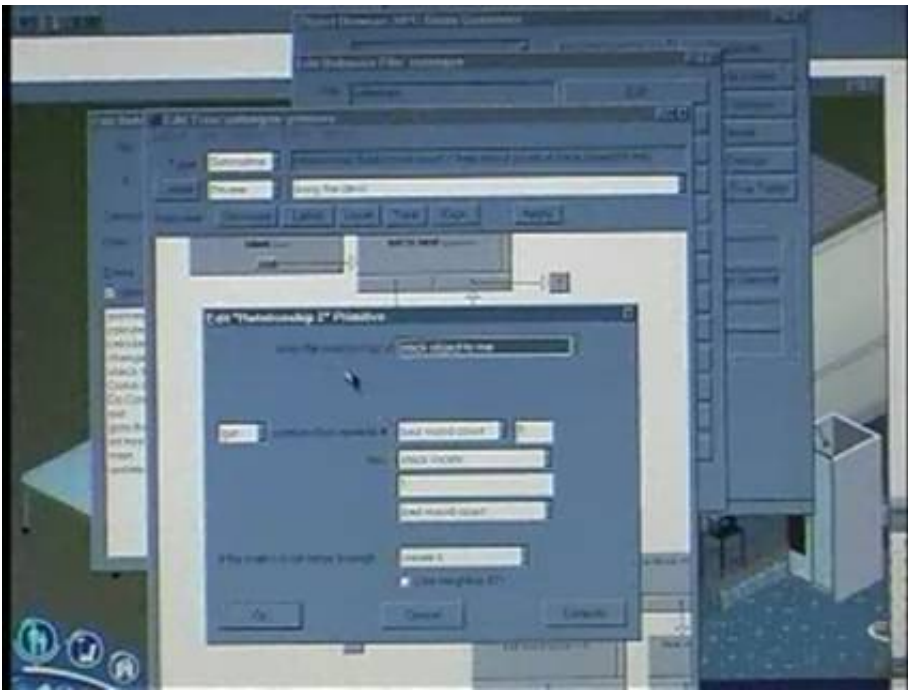
We don't type these in as sentences, we click things on a dialog. So basically, the meaning of this dialog here, is that Stack Object's Person Data called "Family Number" is compared to the Sim Global Current Family. So we're just searching for a family that's current.



- How Long Have You Been in a Bad Mood?

Then it looks for the relationship "Bad Mood Count", gets "Bad Mood Count" of stack object, to...

- Your Personal Relationship with the Satan Generator

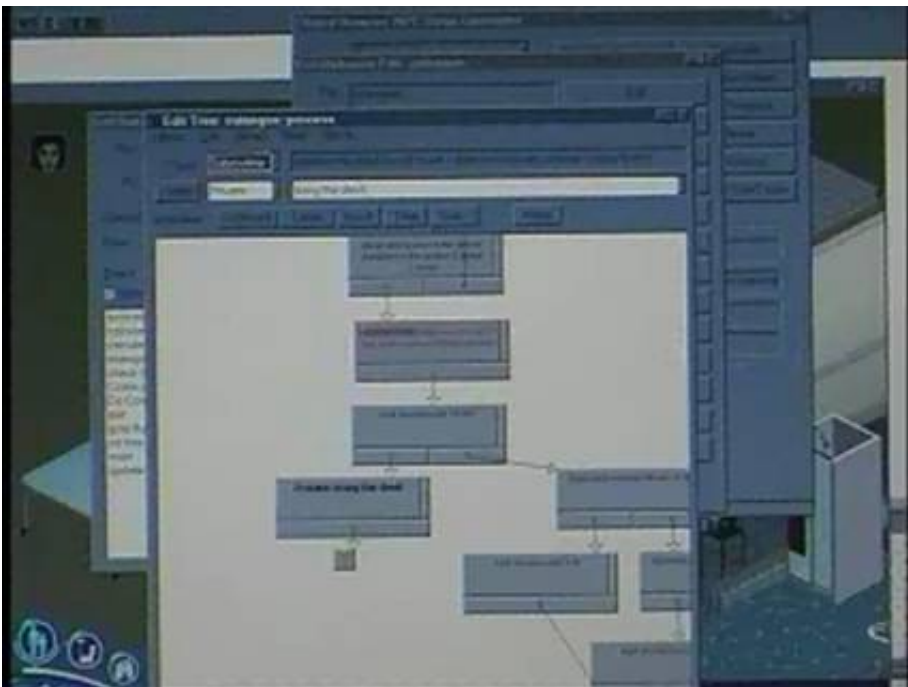


OK, this is editing the relationship of the "Satan Generator" to a person in the family. Here's the relationship of the stack object to me.

Get the relationship variable number "Bad Mood Count" into the stack local 1, called "Bad Mood Count". If the matrix is not large enough, then create it.

Basically, we fill out these little forms, and then make higher level functions out of them.

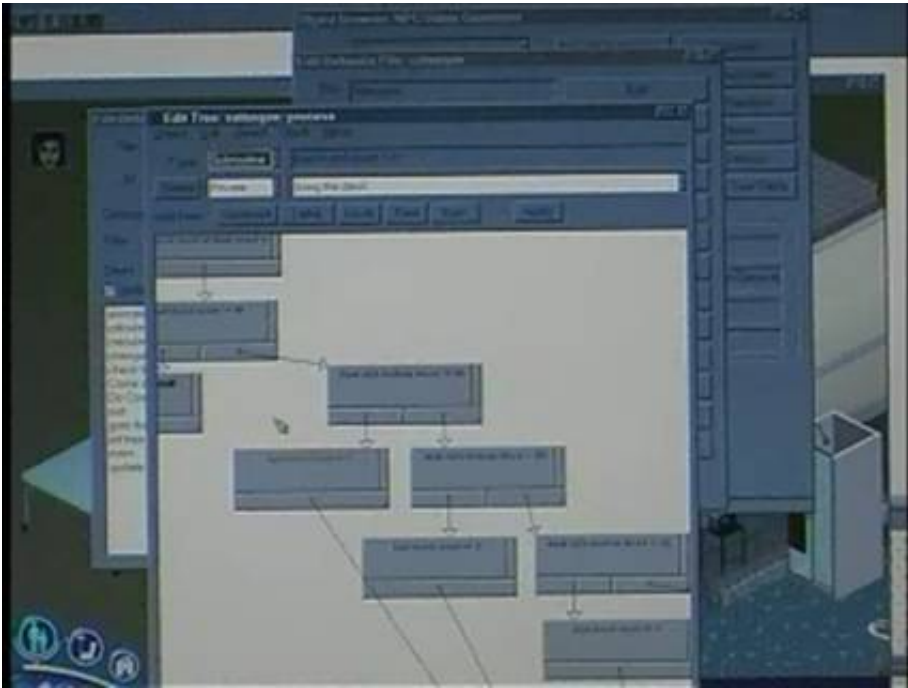
- The Bad Mood Bears



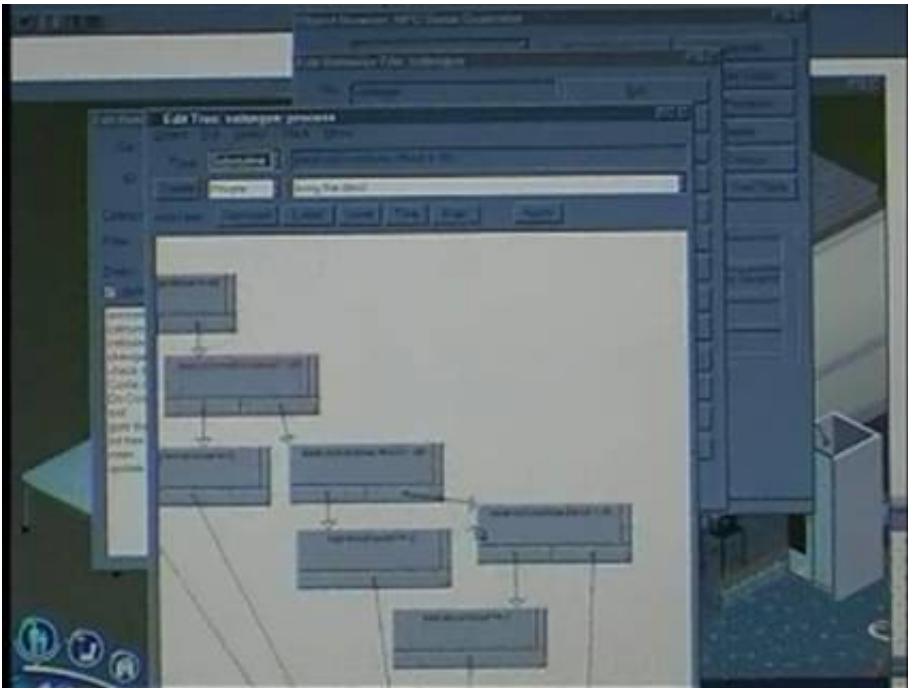
Now the "Bad Mood Count", is that greater than 48? If it is, then "Bring the Devil", that's a function.

Otherwise, the stack object's motive "Mood", is that bigger than 60? Then the "Bad Mood Count" gets 0, if it is.

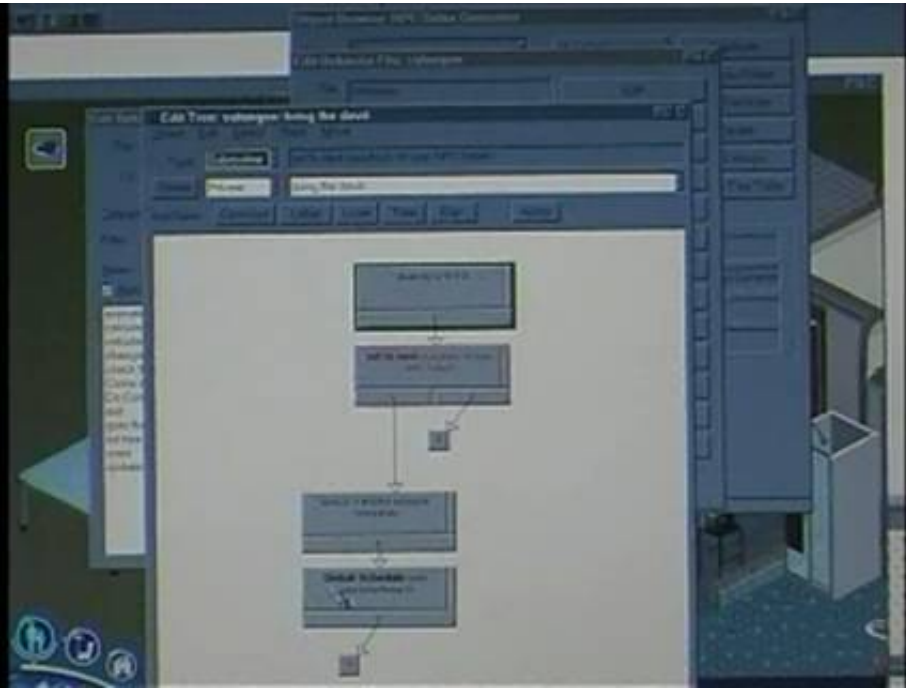
Otherwise, the stack object's motive "Mood", is that less than -80? Then we increment the "Bad Mood Count".



So we're just modifying our tally of how long they've been in a bad mood, until we bring Satan.

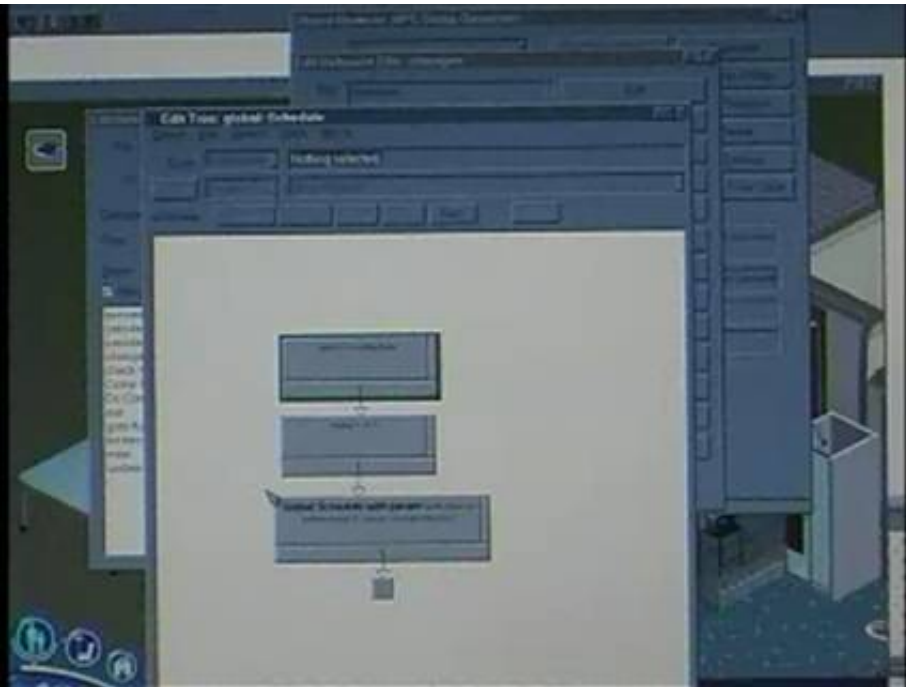


- How to Bring the Devil



And then:

"Bring the Devil" says "Set to Next Neighbor" of type "Non Player Character Satan". Now if there was one, see if that succeeded, then we go down here, otherwise we just return false, or true.



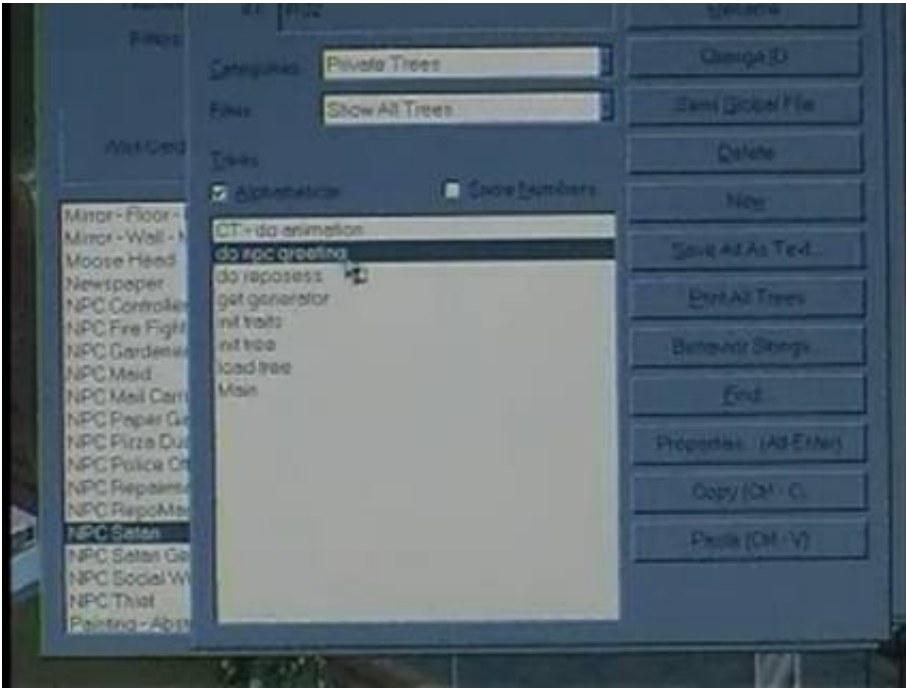
If it succeeded, we get to "Visitor Schedule", and we schedule a visit from Satan, which is this global subroutine that does all this stuff, basically.

- Satan is Just Another Object

That like kicks Satan off, and Satan's another object. And character objects are pretty, uh... They've got a lot of things that they can do.

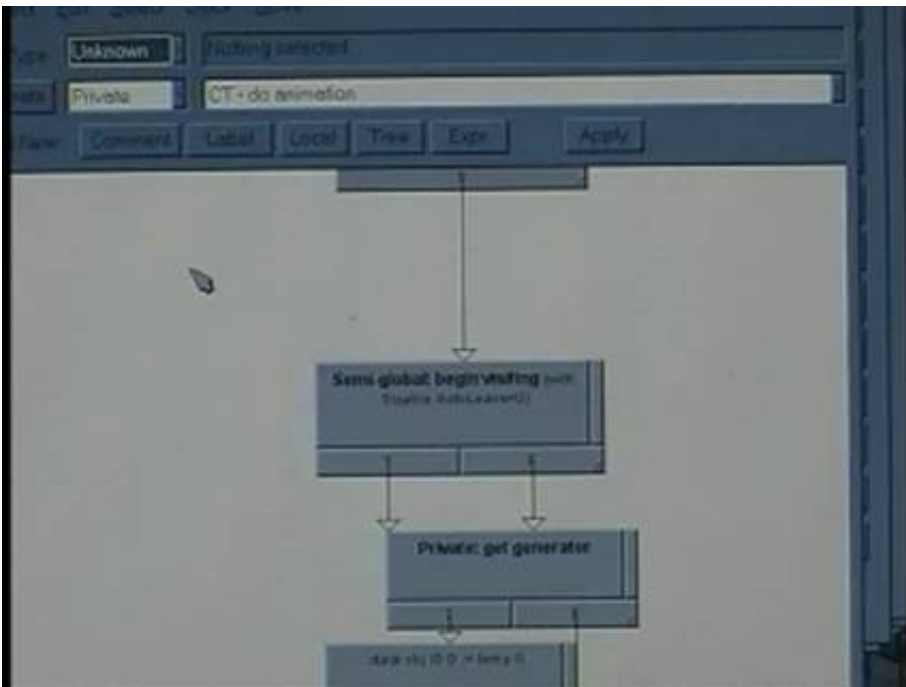


• The Satanic Verses



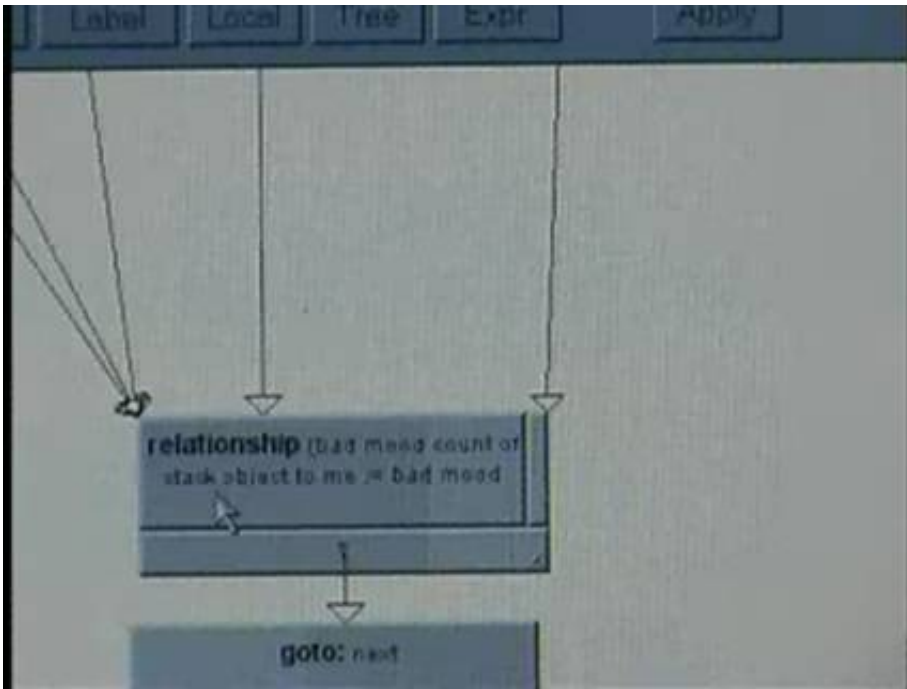
But you can imagine, if we look at Satan's functions, he's got... "Do Non Player Character Greeting", "Do Reposesess", "Get Generator", "Initialize Traits", "Init Tree", "Load Tree", "Main". Not too much, but most of it's probably in "Main".

• Satan's Little Homunculus

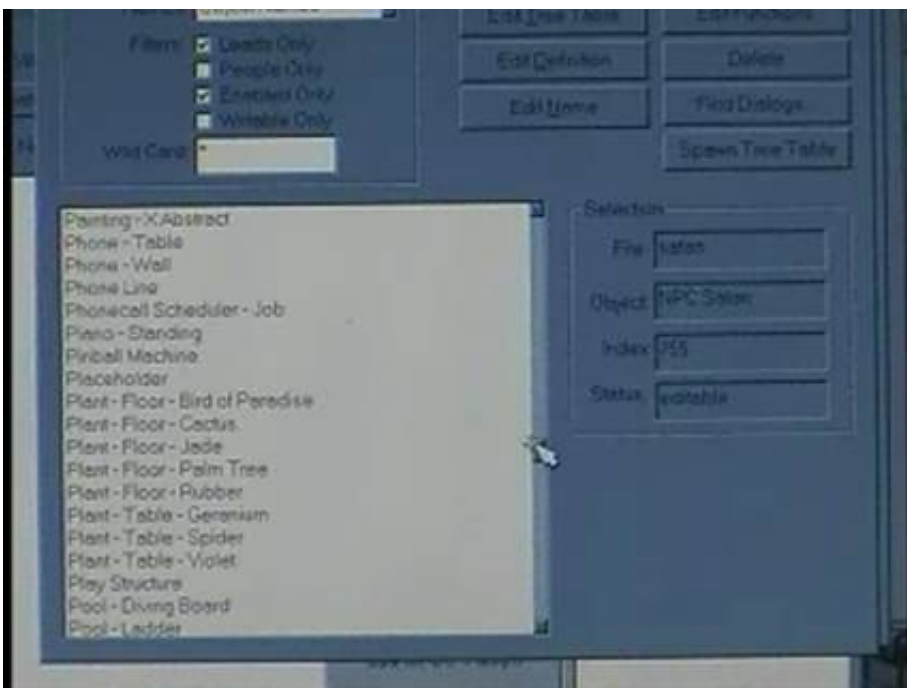


So all these characters have a little program counter in these trees, which is walking down over time. So they're all kind of asynchronous, and waiting for things to happen, like "Walk Me Over" to this place, and it return true or false whether you got there. And then you can figure out something else to do.

- Pasta Oriented Programming



This is the spaghetti code aspect of it. That's what's needed to make it work. There are certainly higher level approaches. But this one was developed over a long period of time, and has a lot of inertia and baggage associated with it.



And it took a lot of work to make all these tools, but it paid off in that we got the interactive editing ability. Prototyping, design sessions, quick turn-around, and get it actually running really fast the next day, see what it's like, get it to the testers.

- Debugging The Sims

You need a lot of tight feedback and to be able to just jump in and fix a bug. Because boy, there were some really obscure bugs, and still are.

It's got a debugger that will pop up if something goes wrong, and show you the stack and let you browse where it was doing what.

- It's Better to Regret Something You Did Do, Than to Regret Something You Didn't Do

It's a big investment to make all these tools, and if we did it again we might approach it totally differently, because there are different tools available today. But it's a lot of fun programming in these languages.

- The Summer Intern Did the Maid

Certainly we have had summer interns that came in and sat down with the programmers and they learned how to approach this with some guidance. But once they got it, they struck out on their own, and really did some amazing things.

The maid was written by a summer intern, and that was the first non player character that really did anything, and it did quite a bit.

- The Testers Dunked the Maid

But then the testers found that the maid could easily... Well umm, you could make the maid do kind of funny things. Like you could build a swimming pool, and put an island in the middle, and then put something dirty on it, and she would jump into the pool and swim across, and climb out and clean it, without cracking a grind.

But that got reported as a bug. But stuff like that, you can't expect, but you can't sweat. Because at least the game doesn't crash, and maybe it will cause some amusement.

- Just Ship It

There were lots bugs, or things that were flagged as bugs, but that's just the physics of the thing. So you just have to say when you're going to stop fixing bugs and ship it.

- Releasing Tools Before the Game

The character animation system was shipped before the game as an application

called "SimShow", which draws the people in a window, and let you paint the skins, and see how they looked animated.

So people were authoring animated skins before the game was released. And I have a more advanced version of that now that I'll show.



(End of this demo)