Don's review of
The Sims Design Document

Draft 3 – 8/7/98

Note to reader: the information here is for the use of The Sims development team. Sections that are decided and can be considered complete are in **bold**; other sections that are under discussion, or have competing opinions, or will be discussed at a later time in the milestone timeline, are normal font.

People

People Mode – Control Panel UI

People Mode - In the World UI

Motives

Simulation rules

SOCIAL MOTIVE

Romance note on 8/6: The July milestone implemented much of current Romance design. The success of that design is again under discussion. We will review Romance again when we design the story using character skills set in scenarios. This section has been un-bolded because it is returned to open, undecided status.

Romance in Social Motive

Romance occupies the high end of the social motive scale, from +70 to +100. In the social motive, regular and romantic interactions accrue social points ONLY to +69. To go to +70 and beyond, it takes romantic interactions. The social motive bar should reflect the "romantic territory" by appearing a different color. It is not possible for a character to move above +70 with another character unless they are "in romance".

Note that the social motive will often be well below +70 and all interactions (social and romantic) will boost it up. However it is only romantic ones that can boost it up over +70.

Romance in the World UI

Immediately after any type of social interaction between two characters finishes, there will be either a friendship or romance icon that will pop up and flash over the heads of the characters. (all social interactions are categorized as Romance or Friendship.) The friendship icon is TBD but the romance icon is a heart. These icons will reflect the change in relationship that just occurred as a result of the interaction. If romance moved up, the heart would flash on screen and would be followed by another larger heart etc. (# of sizes tbd).

The flashing size technique is visually ambiguous and not very obvious because you have to stare at the screen and wait to see the change, which cycles. As a rule of thumb, if you can't read the meaning of the icon from a STILL screen snapshot, then it's not obvious enough. The current design fails this test. Forcing the user to wait through the animation and interpret the changes over time requires a lot more cognitive attention than just using a unique icon for each state that the user can recognize in a glance. If the icons are unique the user can look at the whole screen and take in the information at once in parallel, but if they have to concentrate on individual animations and wait for them to play out, it takes a lot more attention to see what's going on. Flashing the same size, flashing larger, and flashing smaller require that there are more than two frames to disambiguate, three at the bare minimum, and at the small size of the icon there is not a lot of room to vary the size. Why not use completely different animations? A Broken Heart, animated or not, is instantly recognizable as a negative social factor, and visually distinct from a whole positive heart, for example. The size of the icon is good for showing the intensity of the change, but certainly not the sign of the change (+ or -). Bouquet of Flowers, Dove of Peace, Vulture of Antipathy, Parrot of Piracy, etc…

Friendship icon gets larger and larger = positive relationship points (no romance)
Romance icon gets larger and larger = positive relationship points with romance
Either icon just flashes same size = no net gain or loss
Icons flash increasingly smaller = relationship point loss.

Romance in Relationships

Romance between characters can occur if the relationship between them is > +50. It is at this point that the menu items begin to include romantic interactions. Relationships can be romantic one-way and two-way (of course). One character may reach a >+50 relationship and may initiate a romantic interaction without the other character up to the same level. In fact, it is this difference in relationship values that will make it interesting to watch because a romantic interaction may be reciprocated OR may be flatly refused. If a refusal or rejection drives the initiating character's relationship with the refuser to plummet, there will be no more opportunity for romance until the relationship value is boosted up again to >+50.

Relationship scores can go to the max (+100) with or without a romance between characters, thereby reflecting the difference between best friends and lovers. The relationship panel reflects the state of the relationship.

The romantic benefit

There is a strong benefit to having romantic interactions because every point above +70 is added to the mood score. Since the mood score is an average of all the motive scores, the addition of the straight romance points is a BIG WIN.

The same color that indicates "romantic territory" in the social motive bar should indicate the mood points attributed from romance. (We'll try this and see if it works. It may be too hard to tell the effect of an individual interaction on the motives since they are simultaneously being controlled by other effects too)

When a romantic interaction causes an increase in the social motive, the amount of that increase is added to the mood score, but only when the social motive exceeds 70.0, and only to the extent that the increase increased the score over 70.0. For example, an increase of 10 points from 62 to 72 would only add 2 points to mood, but an increase from 73 to 83 would add 10 points to mood).

Relationship panel

The relationship panel should be changed in the following ways because player testers do not understand the colored backgrounds behind the heads. So:
The heads will have a small "gas gauge" type rectangular bar under each of them. This will fill up as the relationship gets better and will go down when the relationship score decreases.
There will be a heart outline that will encircle the head when there is a romance going on with that character. The outline of the heart will indicate the strength of the romance. The higher, the thicker the outline.
(Let's try the heart outline. We'll see if thicker outlines "read". If not we'll evaluate whether we need to show the romantic movement or whether just showing it's existence is enough.)

TBD Design – There is an anticipated problem of how to display the flip from declining romance into friendship when the relationship points are high. Does the friendship icon display after interaction, in the middle of interaction, or what?

The whole relationship design and implementation (I've looked at the tree code) is Heterosexist and Monosexist. We are going to be expected to do better than that after the SimCopter fiasco and the lip service that Maxis publically gave in response about not being anti-gay. The code tests to see if the sex of the people trying to romantically interact is the same, and if so, the result is a somewhat violent negative interaction, clearly homophobic. We are definitly going to get flack for that. It would be much more realistic to model it by two numbers from 0 to100 for each person, which was the likelyhood of that person being interested in a romantic interaction with each sex. So you can simply model monosexual heterosexual (which is all we have now), monosexual homosexual (like the guys in SimCopter), bisexual, nonsexual (mother theresa, presumably), and all shades in between (most of the rest of the world's population). It would make for a much more interesting and realistic game, partially influenced by random factors, and anyone offended by that needs to grow up and get a life, and hopefully our game will help them in that quest. Anyone who is afraid that it might offend the sensibilities of other people (but of course not themselves) is clearly homophobic by proxy but doesn't realize it since they're projecting their homophobia onto other people.

Art Notes

3D Animation

- There will be one skeleton model for all adult characters
- Child-sized skeletons are under discussion as a result of meetings with Pet Fly Productions and Ron Cobb
- There will be ?? number of skins
- Naked skins are required – each character will have one set of clothing, one naked set
- Why only one set of clothing? If each person had several modular sets of interchangeable clothing, there would be many more varieties of dress that one person could wear, like real life.
- Bathing suits/underwear under discussion
- Hats? Hair? Held items (accessories like mugs, plungers, etc).

Upper and lower body scheme?

Sprites

- Some animated (duck) others not (baby)

In the Design a Family Feature

Design

- **Will be included in game**, particulars of interface and if applicable to story are TBD.

Simulation

- Designer family members can act as neighbors, but will not be used in level play.

People at Work

- **Newspaper/job will be included in game.**
- Carpool with vehicle outside to pick up character is TBD

People Voiceover

Several experiments (using Freshness tool for speech) and discussions with EAC sound department (to provide tools to support real language) have proven that we will have to use our fallback option of unintelligible speech for our character language.

The possibilities here are to use:

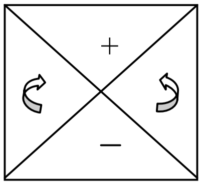
- An obscure language (like Navajo or Esperanto) for entire "script"
- Use a more scripted and directed version of Charlie Brown speak than we were able to provide in June milestone.
- It's easy to play back sounds at slightly different speeds to effect the pitch. Each person could have a certain speed they played their base voice effects, and that could be multiplied to several base voices, to result in many more distinct effective voices.

Control Panel –

The Control Panel consists of 4 main sections: the View Controls, The Game Controls and the general and specific mode tools. The game controls and view controls are common to all modes. The mode tools change by mode.

View Controls

- **zoom in**
- **zoom out**
- **rotate world 90 degrees right**
- **rotate 90 degrees left.**



The View controls need to indicate only two states: inactive and active.

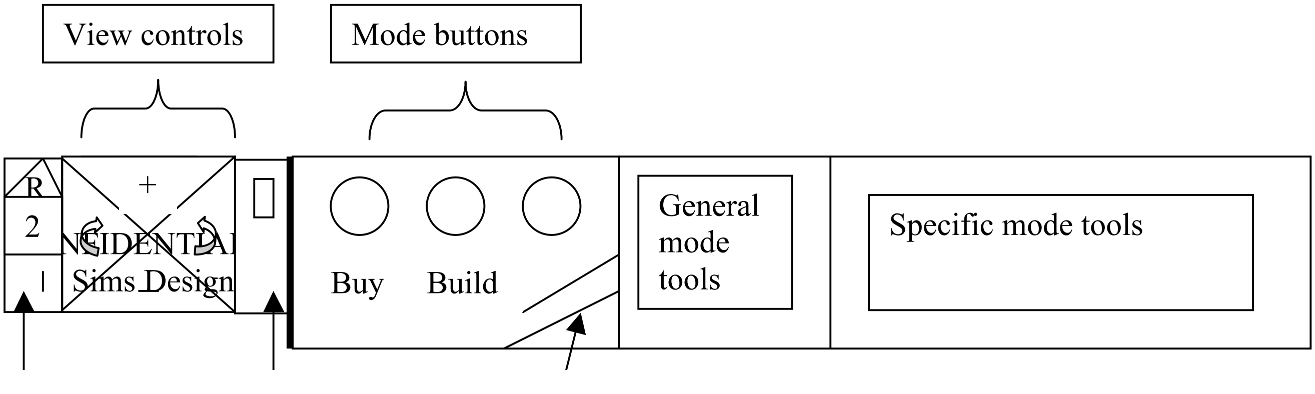
The zoom & rotate view controls on the control panel currently operate differently than the ones on the pop-up menu. The pop up menu pops up over a location in the world, and when you zoom or rotate from the menu, first the world is centered on the spot where you popped up the menu. This is not possible with the view controls on the control panel, since you're not pointing to a spot in the world to center on. The center command only makes sense with the implicit argument of where to center. The pop up view menu does not disable the zoom and unzoom items at max and min zoom, since they are still useful for their centering behavior. So when you zoom in when you're already completely zoomed, it just centers. Maybe the graphic could change to show that. Maybe we should redesign the pop-up to have an optional center command, or center instantly when it pops up. The pop-up would be much more convenient if it were capable of zooming and rotating more than one notch at once. It's extremely often that I'm zoomed all the way out, and want to zoom all the way in, or turn around 180 degrees, and it's very annoying to have to select from the menu twice in a row. How about using the cursor distance to select zero, one, or two notches? (All that's needed is 2 notches for rotating since you can rotate the other way one notch instead of rotating one way three notches, and there are only 3 zoom levels so 2 notches is always enough). I suggest a zone in the center that just centers, and outside of that are two rings at different distance, for one notch (inner) and two notches (outer). The feedback could be immediate, as soon as you pop up the menu the center happens, and as soon as you move into a slice the zoom or rotate happens. That way you could temporarily "peek around 90 degrees" or "step back and get the big picture" by moving out and back in then canceling the menu.

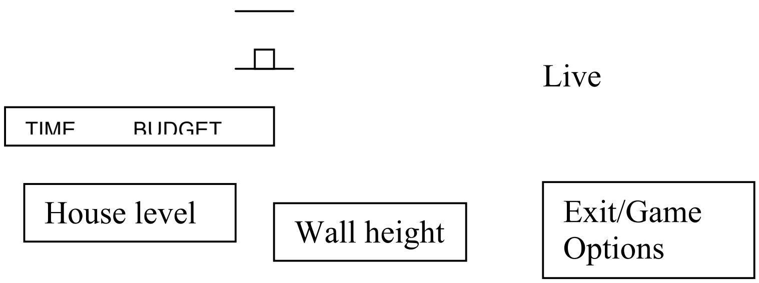
Game Controls

- **Time**
- **I suggest the time control does not have a “pause” setting, just speed (slow to fast, continuous, with a plateau or notch at normal). In order to pause, you can go into buy mode or architecture mode. People mode is ALWAYS live. No pop-up modal dialogs in people mode that freeze time. It’s horrible for a pop-up modal dialog to freeze time in people mode.**
- **Budget**
- **Wall Level**
- **House level**
- **Mode buttons: Buy, Build, Live, Game Options (this mode is smaller and less important visually than the others but it does include EXIT)**
- **We should get rid of the “Game Options” mode. What use is there for it, that we simple buttons or menus don’t satisfy. Is there some reason the game needs to be frozen in order to save? In order to abort and go to the neighborhood screen? Etc, Each mode you add has a high cognitive overhead because it distracts from the other more important modes live Live and Buy. It’s a waste to throw in unimportant modes like Game Options, when they could be buttons accessable from any mode instead.**

- **We should have just two modes, Play and Edit. Two is much simpler than four, and simpler than three. The more modes, the less important each one is. Obviously “Live” is the most important, so it must stay. “Game Options” is the least important, and it should go. This leaves us with three, one of them most important and two less important. We are already struggling with the problem that Architecture mode is the “forgotten bastard stepchild” mode that nobody gets around to trying out, and having architecture be one of the lesser modes only amplifies this problem. We should merge architecture and buy mode, so that there is “Play” and “Edit” mode, and building and buying are just subpanels of Edit mode. It’s really good to have two “opposite” modes, since they are very distinct. Time flows in play mode, and clicking on objects tells people to do stuff with them. Time freezes in edit mode and clicking on objects moves them around. With two modes, they can be balanced so each is equally important. Merging build and buy mode also solves some of the ambiguities about the extent of undo: if you buy something then build something then buy something then build something, can you undo and redo that whole sequence because you never went back to “Live” mode, or was undo thrown away when you went between build and buy? Throwing away undo seems unnecessary and obnoxious. The Build/Buy distinction is artificially inflated by giving each a different mode. The default behavior of moving objects around in build mode should be exactly the same as moving them around in buy mode, so they should be the same mode, since they are so similar. Of course we can still keep the commands in separate sub-panels that aren’t displayed at the same time, but making three or four modes simply takes up a lot of screen space for the big column of mode buttons (when a Toggle Switch would be so much more succinct), as well as cognitive real-estate in the user’s mental map of the system.**

The name “Live” for play mode is not a good word, since it is a homonym, and there are two different ways to pronounce it, with different meanings, which are ambiguous in this situation. Does Live rhyme with Give and mean the opposite of Die, or does Live rhyme with Hive and mean the opposite of Dead? How about “Play”? Or some other word that means time is running and people are in control.





The following controls need to display three states: Wall level, Mode buttons and house level. Their states can be unselected, selected, or unselected but being rolled-over-highlighted. The unselected state should be darkest, the selected state should be the brightest and the rollover highlight should be somewhere in between.

Time is a readout of the game time. It is also a button which accesses the Speed control in the game. When the Time button is selected, the text hilights, and takes over the Budget space next to it. In the Budget space the speed controls are displayed with a slider. Slider has x number of positions to set game speed from slow to fast.

Budget is readout only and displays the amount of money in the household account.

Walls display three stages available for selection: no wall, cutaway walls, high walls. Default position is cutaway walls.
Only the walls of the current level are effected by the wall selection. Walls on the lower level(s) are drawn at full height.

House Level also has three stages available: first floor, second floor, roof.
Default is first floor level.
OK so we've decided on 2 levels.

The Look

The overall style will be in accordance with Charles's identified modern minimalist look.

The common non-mode specific items in the control panel will all be placed in two blocks, of approximately equal size, starting at the left end of the control panel. Mode specific items will be placed in two blocks in the center and right sides of the control panel. Refer to the illustration.

General - In the World UI

The view controls can be displayed in the World by clicking the right mouse button when cursor is in the World.
See comments on the popup view menu above.

Memory in Modes

Each mode (Buy, Build and Live) will remember the tool/room/function submenu the player was using prior to switching modes. For example, if a player has been using the wall tool and then switches to Buy Mode, when he returns to Build mode, the wall tool and its submenu is selected. The memory will not go so far as to remember the slide-up info panel that could have been up for any particular tool.
Should not remember global view controls like zoom, rotation, and level.

Mode-Specific Tool Info
Each mode has tool info. For a description of this feature, see the section Object Info Panel under Buy.

Objects

Vocabulary:
<u>Control Panel</u> (CP) is all buttons, text, etc below the Control Panel top line.
<u>The World</u> is the entire simulation area above the Control panel.

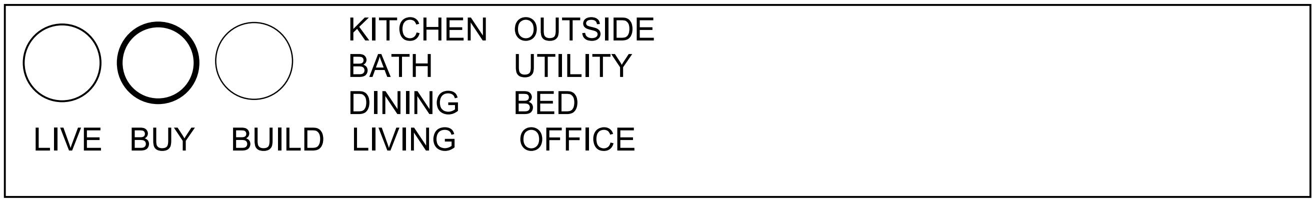
Buy Mode – Control panel UI

Buy Mode is the section of the game where the player purchases and places objects into the house/environment. The player is limited by the money in his household budget. During Buy Mode, the simulator pauses and time does not pass. There is no penalty for staying in Buy Mode for any period of time.

SELECTING BUY MODE

Click on the BUY button in the left set of buttons on the interface. Room choices display to the right of the buttons Why isn't there a default room selected? How can the user select no room? Why should the system come up in a default state that is impossible for the user to put it back into? How does the system behave when no room is selected? The first room should be selected. I can't think of any reason not to.When player leaves Buy Mode to visit Architecture or

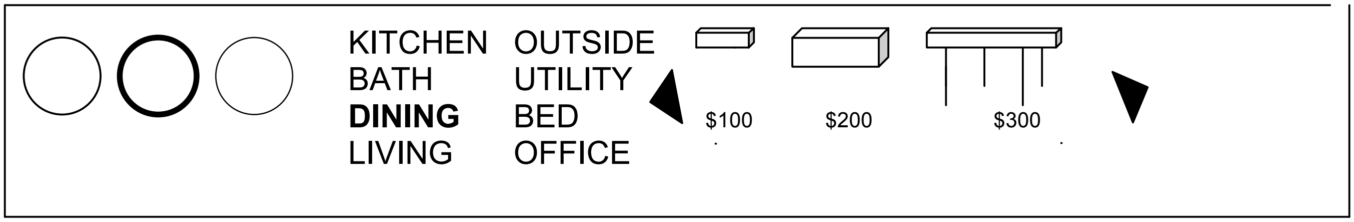
People Mode, then returns to Buy Mode the last object selection is still displayed. (since the Info Panel closes with the selection of another mode, that Panel will never be displayed on return to Buy Mode.) The text names of the rooms act as buttons to call up that room's specific objects. The text names of the rooms do not sit on graphic rectangular buttons, the text name itself serves as a button in order to allow for easier localization. It would be a good idea to provide a scrolling list, so plug-ins can come with their own rooms. I don’t agree that we should get rid of the “Function” categories, but separating the categories into super-categories of “Room” and “Function” seems cumbersome. Why not just merge the function and room categories together at the same level? The list will have the names of rooms, as well as the names of functions, and it could even have the names of styles of furniture, that came in plug-in packs. Plug-ins should be able to define their own categories. The user should have some freedom to rearrange the list of categories so the ones they use frequently are easy to get to. SC3K has a scrolling list that the user can edit by dragging and dropping items around, that would suit our needs. They use it for the list of map views, or something like that. The user’s category arrangement could be saved in a separate file.



SELECTING A ROOM

Rolling over the room name will highlight the text . Mouse down on the room name will select the room and the text will assume the "selected" color. As many rows of objects as will fit appears to the right of the room names. At the rightmost end of the row, there will be an arrow displayed only if there are more objects to be viewed. Clicking on the arrow will bring up another new set of objects (i.e. objects do not scroll on one at a time.) A new set of chairs will appear if scrolled. It is assumed that there are as many rows of objects as will fit on the screen. (Charles’ layout will test this assumption). There will be an arrow on the left side to allow the player to get back to the first row. The price of the object will appear as a hot label when the cursor rolls over the object; no object name (i.e. “Antique Mahogany

Table") is displayed, only price. The price hot label may not obscure it's object, but can obscure any other neighboring objects.



(These illustrations are not in scale. They are for discussion purposes only)

Sorting Object Functions in the Control Panel

The default sort will display the Room name. If player Clicking and holding is a horrible user interface technique. Don't do it! It's invisible, there are no visual cues to tell anybody that they can do that. Don't force users to go around clicking and holding on everything to find out if it does something special. Requiring the user to hold down the mouse button is bad. Small children will have problems with it as well as adults. Requiring the user to sit there and wait, even a fraction of a second, makes the user interface sluggish and un-responsive. If the 400 mhz computer is going to do something, it should do it IMMEDIATELY when you click the button so as not to give the appearance of slugish non-responsiveness. the Buy Mode button the Rooms are replaced with a Function sort. Functional categories are Chairs, Surfaces, Appliances, Electronics, Plumbing, Decorative, Miscellaneous. Get rid of Room –versus- Function sort, and put all the categories together at the same level. More like categories in the yellow pages. Who would be so anal to separate all their catalogs by such a distinction of room –vs- function?

Here is an example of what happens when sorting and moving between modes: If player is in Buy mode and objects are sorted by room, and chair object is selected. Player selects Architecture mode, then goes back to Buy mode and click and holds. The sort is now by function, the chair is in its position on the CP UI, but is not selected.

ART NOTES

The size of the Control panel objects are assumed to fit in a 45 x 45 pixel square. This assumes a fit within the trimmed down 100 pixel Control Panel that runs on the bottom of screen. These numbers are estimates and will be altered during Charles' layout of CP.

The Objects should appear with no shadows in CP.

The reworked version of the interface (from 6-24-98) is monochromatic. The objects will look realistic and colorful. They may not match the style of the interface and may look weird against a stark background. Perhaps the background will need to have some color added to fix this visual problem.

The objects displayed in the UI will be a separate set of graphics tailored to fit the size of the UI CP space. (45 x 45 is estimate.)

Selecting an Object

Is this about selecting an object in the buy panel, or an object in the world?
How can you select an object in the world while you're in buy mode, if clicking in buy mode buys an object?

To select an object, click it's graphic on the Control Panel. Object's footprint highlights. Grab hand cursor displays. What happens if you press down over an object in the buy panel, hold the button down, and drag it out into the world, and release it where you want it to go, like a normal drag-and-drop operation. People will definitly try this, as well as clicking and moving without holding the button down. Both modes of operation should be supported. Dragging an item out should be a shortcut that uses as few mouse clicks as possible. Right now dragging an object out does not work. In these descriptions of mouse interaction, please be VERY specific about when the mouse button is up and when it is down. When writing about pointing and clicking, consider both cases of dragging with the button down, and clicking and releasing the button, because there's no way to prevent people from trying both, and they will. Always tell when the mouse button should be held down, and what happens when it goes down, and what happens when it releases. Don't gloss over the mechanics of mouse tracking assuming it will all work out, because this is where many bad snags happen that could make it hard to use. As the cursor moves up over the top line of the CP and into the World, the small size object becomes it's zoomed size, matching the level of zoom of the World it will be placed in. Does the small icon track the hand cursor? When? When the cursor is over the control panel? Does the small icon disappear when the cursor moves into the world? Is it replaced by the real object (at current zoom scale)? When the cursor moves back into the control panel, does the object you're placing stay in its last position in the world, or does it disappear? This may have to be different for moving an object that already exists and placing an object you just bought. If you pick up an object in the world to move it, and move the cursor into the control panel, the object currently stays along the bottom edge of the world in the last valid place you pointed to, but snaps back to the cursor when you return to the world. A horrible bug is that if you pick up an object, move into the control panel, and select another tool that deselects the pick-up tool (like clicking on a catalog item), the item you were moving IS DESTROYED without asking you! What should have happened instead in this case? Does the object you were moving stay at its last position? (Then it should not have dissappeared when the cursor entered the control panel, but if you were buying a new object instead of moving one that already existed, then maybe it should have disappeared – this is an inconsistancy caused by the rule that you don't buy an object till you place it). Or should the already existing object that you're moving snap back to its original place where you first picked it up when you move the cursor into the control panel? Or should it do that only when you change to some other tool in the control panel?

I don't believe you should ever see the real object in the world unless you have already payed for it. You should buy it when you press the catalog button, and then it's automatically placed somewhere in the world. You should be able to click the table once, and then click four times on the chair, and have a table and four chairs in the driveway to place. Then place them according to the normal placement rules, and there won't have to be any of these weird anomolies between moving pre-existing objects and placing new objects. Seeing the real object in the world that you can't afford to buy could be considered cheating. We want to save the expensive eye candy for when you can afford it, we don't want people to lose interest in the expensive objects after they've looked at them before they can afford them. Say there's an expensive merry-go-round that you can't afford, why the hell should you be able to see it and move it around all over your yard without being able to afford it? The feedback showing you can't place an object is ambiguous, does not tell you the reason you can't place it, and the inside/outside only rule only makes it worse. So if you click on the merry-go-round and try to place it in a big room of the house, you can't tell if you simply can't afford it, or if you're just not allowed to place it outside. So you clear a bunch of trees outside and try to place it again, and you still can't place it, so you sell a bunch of stuff, and you still can't place it, and you finally discover it has to be up against a wall!!! The "red X" feedback is totally ambiguous and gives no indication of why you can't place things, and there are TOO MANY reasons why you might not be able to place something (collisions, missing floor tiles, the indoor/outdoor only flag, the must-be-against-wall flag, and worst of all not being able to afford something). We should design the user interface to eliminate the problem of not being able to place something because you can't afford it, by making it impossible to see it in the world if you can't afford it. You simply can't buy it if you can't afford it. Click on the icon for a merry-go-round, and IMMEDIATELY hear the buzzing sound that means you can't afford it. The icon does not select, and you're left in "move object" mode, or whatever you were in before.

Buying

To buy an object, click on the object in the Control Panel. Click to place the object in the World. This is the only way to buy an object -- selecting the object from Control Panel does NOT purchase the object. THE OBJECT IS BOUGHT WHEN IT IS SET DOWN. The budget is decreased by the price of the object when the object is placed. The footprint highlight disappears when object is set down. What is a footprint??? Is it something on the 2d icon or the 3d sprite? Is this the "mouse cursor" holding the object underneath? What does it look like on multi tile objects? Is it based on the shadow mask? What if the object's big and you can't see the footprint underneath it? Is the object lifted off of the ground by some amount?

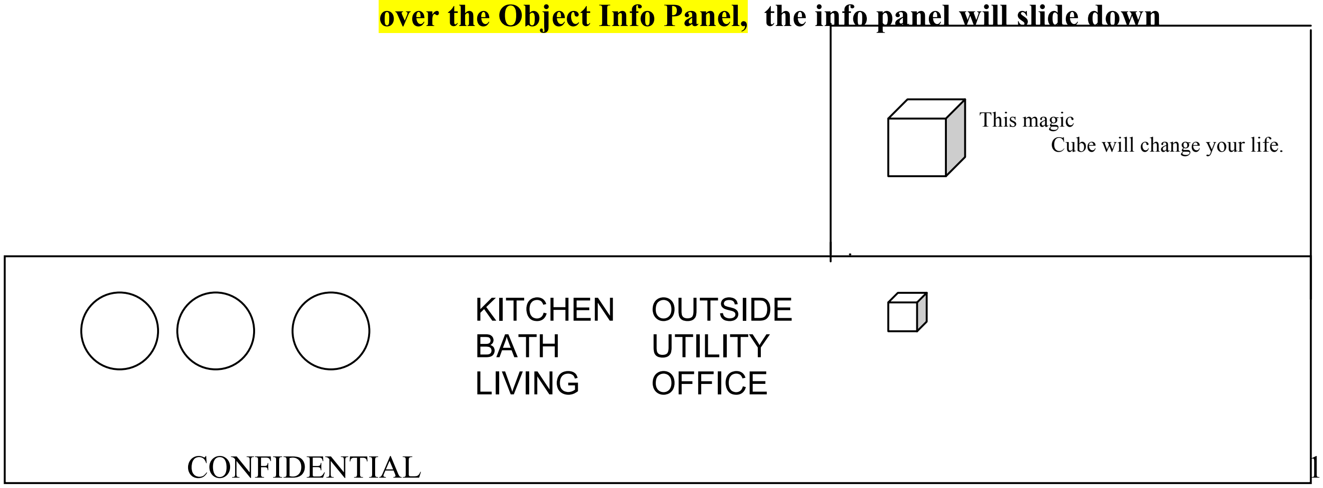
Object Info Panel

Holding the mouse button down is a bad UI! See the discussion above. Why is there any need for a delay? Why can't it slide up instantly? Why this fascination with clicking and holding? on an object in the Control Panel for the info panel to slide up. This panel will display the largest zoom level size of object with a short (less than 50 words, two sentences or less) text string with hints on the motive or other benefits of this object. This text string will reveal only some of the benefits the object provides – it is not meant to give total object information, nor should it be too specific (will not say "this aquarium will give +5 environment points.")

The width of the Object Info Panel slides up above the area of the CP objects. See graphic below.

Sliding over to adjacent object and click once on another object in Control Panel will replace the first batch of info on the slide up panel with that second object's zoom size graphic and text.

As soon as cursor crosses over from the Control Panel into World, or over the Object Info Panel, the info panel will slide down



DINING BED

Returning Before Buying

The player can return an object before buying it. Here is the sequence of actions:

Select object from UI CP by clicking once, footprint highlights, grab hand cursor displays on object. Move up into the World or anywhere on the CP, and without placing object in World, user can Drag seems ambiguous. Is the mouse button down or up? Can you deal with both styles of dragging? (Click, move, click; press, drag, release. A click is defined as a press followed by a release, with no motion inbetween. A drag is defined as pressing, moving while holding the button down, then releasing.) the selected object back to any position near the CP objects, click once and object will be replaced (no dollar amount in bank account needs to change, since object is not bought until it is placed in World). It not necessary to put back exactly in it's "home' spot on the CP UI. This sounds really weird. There MUST be a visual affordance (a button or a drop target) that tells the user that this command is available, and invokes the command. Even if you can think of 10 nice shortcuts for deleting an object, there should be at least one visible way. There's nothing wrong with a button that says "Sell", as long as it works rationally. It's much more important to design a user interface that visual reveals what the user can do (with labeled buttons and icons), than inventing a bunch of convenient shortcuts you can only figure out by reading the manual.

While the object is being moved around in World, its CP UI equivalent retains its highlight until the object is placed, or the object is returned to the UI. (object footprint unhighlights if placed in the World, and CP object unhighlights at same time).

We need some way to highlight the currently "selected" object. What is this footprint? Won't it be covered up by the object? We could flash the highlighting at some rate between dark and bright, which would draw attention to the object.

Buy Mode - Objects in the World UI

Selecting

To select an object in the World, click on it. The selected object displays a footprint under it. This is the same footprint as seen during placing an object in the World. The cursor becomes a grab hand.

How can you click on an object to select it in buy mode? As I understand it, when you're in buy mode, an object to buy is selected from the control panel, and any time you click, you buy one of those objects. So you can't click on objects to select them, because clicking means buying, not selecting. How does it know you don't mean to buy an object and place it on a table, when you click on a table? It's much too "clever" (i.e. confusing) to temporarily switch out of buy mode when you point at an object you already own. (What do you do with the object you're dragging around to buy? Does it disappear? This violates conservation of matter. Does it stay in the last valid location? How can you tell it from other objects that you actually own? This is another reason I argue that you should never see an object in the world that you don't already own, and you should buy it at the time you click on the catalog button. Of course you can still get a refund, it's just that the time you buy it moves back in time to when you click the catalog button, so that when you point to the world, the object you're dragging around with the mouse is something you already own. Then it's possible to temporarily set it down in order to move another object out of the way, without needing to draw some visual distinction between the un-owned "ghost" object you haven't bought yet, and the already owned object you're moving out of the way. When people are placing objects they've bought, they are going to want to interleave the moving and rotating of that object, with moving and rotating of other objects they already own, so they can make room for the new item. This is why you should always own something before you place it, so the placement tool can behave the same in all cases. It would be horrible to have two different behaviors for simply moving and rotating objects around. We need ONE good consistant design, with no weird exceptions.)

Only objects that are bought in BUY mode can be selected. Other objects bought and placed in Architecture mode (fireplace, staircase, plants) cannot be selected or moved in Buy mode.

Placing Object in World

Select an object from CP and move to location in the World where the object can fit. As the object passes over an unsuitable location (a place where the object cannot fit) an X will appear over the object to indicate that it cannot be placed there. Over all suitable locations, the object appears normal (the object looks the same as it will after it is placed.) After the object is placed, the cursor changes to an open hand. The current "Red X" placement error feedback is horrible. As described above, it gives NO indication why you can't place it (and there are SO many reasons why something might not be placed: space occupied, wall in the way, no floor tile support, can't afford, must be

outside, must be inside, must be on a certain level, must be against wall, etc.) Not only that but you often can't see the red X or can only see an ugly part of it, because it intersects the object you are placing, or an object that's in the way. A bunch of red X's on multi-tile objects looks really horrible, and emphesizes that the multi tile object is broken up into a bunch of tiles. After all the effort we went to to get rid of the seams, you'd think we'd steer away from feedback that emphesized that a big solid object was broken up into a bunch of tiles. The red X feedback is ugly, not always visible, gives no indication of the problem, and no hint to the user what they can do to solve the problem. Please don't just write down descriptions of what just happens to be implemented at this time and pass it off as a purposeful design. Things like this really need to be thought out from a fresh perspective.

Moving an Already Placed Object

To move an object already placed in the World screen It will be hard for people to figure this bizarre maneuver out by themselves, and if they do they'd probably think it was a BUG due to flakey mouse tracking.The grab hand cursor now changes back to an open hand cursor.

Do you have to select the "move" tool in buy mode? Or is there some way to set down the object you're buying (or already bought), and move other things around? How do you distinguish an object not already bought from objects you already own? There should be a distinction if there are objects in the house you don't own, that would disappear if you leave buy mode without paying. But I don't think there should need to be a distinction, since you should have to buy it before placing it.

Rotating an Object

This whole rotation scheme is TERRIBLE and should be redesigned. Please don't just accept the current implementation as the definition of the design. The code is really complex and has lots of wrinkles that are hard to describe, that you probalby don't know unless you've read or wrote the code. Try placing two chairs of different types down, then try moving one chair onto the other chair of the same type, and clicking. Then try moving one chair onto a chair of a different type, and clicking. Go figure. There are other features that exist but aren't in the spec, like holding down shift to "paint" with an object, that interact with the other behaviors in subtle ways. (i.e. Notice the red x's when you hold down shift and drag out flamingos.)

These features were accumulated over time, designed at different times by different people, and the way it works currently should not simply be reverse engineered and called a spec. We really need a fresh design for rotating objects, at the same time as a fresh moving objects design, so that rotation does not interrupt the flow of moving things around, and it not require so many clicks.

To rotate an object, click on the object in the World to select it, and grab hand with rotate cursor appears. Rotate object by clicking into desired position. Move cursor off object, get grab hand cursor and click once to set down. Moving cursor off of the selected object also brings back the open hand cursor.

< > keys will also rotate selected object.

A proposal for rotation: When picking up an object, you can change its direction by pressing and holding the mouse button down and moving in the desired direction. Once you release the mouse button, the object follows the cursor until the next time you click to set it down. When setting down an object, you can also change its direction by pressing and holding the mouse button down and moving in the desired direction. When the button goes down, the item is anchored in its current spot, and if the direction is desireable you just release to begin or end the movement. But if you want to change its direction, just move in the direction you want the front to face. As with pie menus, you don't have to look at the screen to be 100% sure of the direction you're going to turn it, since your control over the direction is ABSOLUTE. This gives you absolute unambiguous control over the direction at both the beginning and the end of the movement gesture. You might want to rotate it before picking it up, as well as after setting it down, and this interface supports optional rotation in both cases, with a minimal number of mouse clicks. When you try to set an object down, but you're not allowed, a temporary dialog could pop up telling you WHY you can't set it down. You could still rotate it around until it was facing the right direction.

Another way to rotate an object while you're dragging it, would be to move the mouse around in circles in the direction you want to turn it, and it would start clicking and turning once you completed a whole rotation gesture. You could keep circling the mouse around and the object would keep turning. Circle in the other direction to rotate in the other direction.

The current "Setting the Time on an LED Watch" interface to rotation (clicking again and again until it rotates around to where you want) is horrible. (At least there aren't 60 rotations!) It's not absolute,

and it's an arbitrary design decision which direction it does rotate (so if you have to click 3 times because you wanted to rotate the other direction, it's very annoying, since you have to look to be sure which way it turned, since some objects turn funny.) The hoops you have to jump through to switch back and forth between rotating mode and moving more are very inconvenient. The technique of switching from rotating to moving by pointing at an object, then pointing away, then pointing back, is awful, and it has the feel of a BUG that somebody discovered was the only way to do something because of a lack of good design, and then documented.

The current design puts rotation at a higher importance than moving. I argue that moving is much more important than rotating, since there are only 4 rotations to choose from, so any rotation has a 25% chance of being right, but there are thousands of positions you can move to, so you need to move an object much more often than you need to rotate it. Rotation is very low res, Position is high (or medium) res. The first command you get by clicking on an object should not be the low res, less useful rotation command (one that requires you to click 3 times to rotate ccw), it should be the much more commonly used movement command.

Selling an Object

Selling an object can only take place in Buy Mode. To get rid of any object that has been bought, the player must sell it. The longer the time between when the object is bought and sold, the higher the depreciation on it and the less the player can recoup of the original value. If an object is sold in the same Buy Mode period as it was bought (that is, the player has not switched to Build or Live Modes), the object loses no value upon resale. Therefore, depreciation starts upon exit from Buy mode into another mode.

Why is this defined in terms of Mode Changing instead of Time Passing? This is another problem with having a distinct Buy mode different than Build mode. Why shouldn't you be able to switch to Build mode, then back to Buy mode, then get a refund for something? For example, I go into Buy mode and buy a big bed that does not fit in the room, so I go to Build mode to move a wall, then change my mind, hit undo (at this point I should be able to hit undo twice to undo the bed purchase, if there weren't an artificial distinction between Buy and Build), then I go back to Buy mode, and now I can't get a full refund for the bed! Why not define it in terms of time that has passed? If I go back to Live mode, then time starts moving again, and the value of the bed goes down then.

There are two ways to sell an object:

1) select the object that has been placed in the World . Hit the Delete key on keyboard. Avoid modal dialogs! with "you paid $xxx.xx for this object. You will make $xx.xx from selling this item" and "Are you sure Y/N?" will appear and the player will have to confirm the sale. Click Y or N with the arrow cursor (which appears only in World mode when used with a modal dialog.) The user just issued a delete command with the keyboard, using the Delete key. So by definition, the user's hands are on the keyboard. It's inefficient to require the user to then home their hand to the mouse (takes at least 1 second), and then move the mouse to the small "ok" button. The dialog should support keyboard input, so you can simply type "<DEL>y" to delete an object.

2) select [assuming there is a way to select an object in buy mode, without buying an new object] the object that has been placed in the World. Pick it up and move back What about the view controls? You should be able to zoom, rotate, and change levels while you're placing an object, without deleting the object when you click the view controls in the panel! It's inconsistent if you exclude the view controls (what do you draw to make it visually obvious that the view controls work but everywhere else in the control panel deletes the object?) but a bad UI if you don't (if clicking on the useful view controls while moving an object deletes the object instead of changing the view)) and click once. (it is not necessary to find the exact spot location on the CP UI to replace an object. I don't like this invisible user interface. There should be affordances that show you what it's possible to do, or nobody will ever figure it out.) A Dialog box with "you paid $xxx.xx for this object. You will make $xx.xx from selling this item" "Are you sure Y/N?" will appear and the player will have to confirm the sale. Click Y or N with the arrow cursor ((Where else is this mentioned in the design document? Why? I think the cursor is useful for orientation, are you suggesting making it invisible some of the time?)

3)

4) This design will require unnecessary mouse movement. The user has just moved the mouse to the bottom of the screen, now you are popping up a modal dialog that is in the middle of the screen. Why not pop up a dialog near the cursor? Or design it so it does not need a modal dialog at all. Perhaps a "Yes/No" pie menu.

5)

6) Better yet, why not rethink the whole design of selling objects? The original proposal was to drag the object into a small target (i.e. the catalog item that it came from, or a "Sell" target), but that was thought to be too difficult, so the spec was changed so

the target was the whole control panel. But now that is TOO EASY to do it accidentally, so the pop up confirmation dialog was added, whose "OK" button is just about as small as the original catalog button or buy button we were trying to keep from having to home in on, in the first place.

7)

8) Here are other ways to get rid of objects: drag them out to the curb on heavy trash pick up day. Have a garage sale on the weekend. Put the object in storage (for a monthly fee you could maintain an off-screen storage locker that you could drag possessions in and out of – this would be really useful!).

9)

10) I think the simplest thing would be simply to have a button on the control panel that ways "Sell", maybe a small drop target recession that has some icon like a goodwill store or newspaper want ads, that highlights when you drag an object over it, and you're told the price you'll get for it somehow, like a temporary popup tooltip. When you drop it on the Sell button, it's sold, without any "Are you sure" dialog. You can always use "Undo" to get it back, so the confirmation dialog is not necessary, but the price feedback is (so it could be displayed as feedback in or by the Sell button when you drag an object over it). Imagine how much you'd hate that far away "are you sure" dialog, if you wanted to sell 25 items, one after the other (after the "flock of flamingos lands in your front yard" disaster).

Since these two options for selling are not permanent buttons on the interface, user will discover them through gameplay. I don't believe so. This is way too much of a risk, hiding such a useful command as selling objects. What's so great about not having a button?

The cursor will turn to an arrow in the modal dialog box. A selected object held in a grab hand will inhibit any modal dialog from appearing.

This is a band-aid rule that will only cause other problems, probably more than it solves. Think VERY VERY carefully about every modal dialog the game is going to have, when it can pop up, what else can be happening at the time, and what order conflicts are handled in. You can't pop up a modal dialog while a pie menu is up. You can't pop up a modal dialog while another modal dialog is up. You can't pop up a modal dialog in the middle of a drag operation (while the button is down), or the operation will never be completed, and leave the system in an inconsistant state. There are severe flow control constraints on modal dialogs, and problems with objects calling out modal dialogs

and blocking, and re-entering the event loop, which eventually finds its way back into invoking the interpreter. The system can easily deadlock and crash. One example of a typical bug we have now is that when you go into "Options" mode, time (and the simulator) is stopped, and when you press the "describe scenario" button (or whatever its called), nothing happens, until you switch back into people mode, at which time the dialog finally pops up. This is because modal dialogs had to be deferred until the simulator loop ran (which only runs in people more), in order to prevent all kinds of other problems, but of course that band-aid patch simply caused other problems that are going to be hard to solve without redesigning other parts of the (already extremely complex) system. It's horrible to pop up a modal dialog in people mode, because it interrupts the flow of the game, and disables all the other controls. Somebody might want to zoom in or out or pan in order to decide what answer to give the modal dialog, but they can't. When the phone rings, and somebody answers it, time should not freeze until the user clicks Yes or No they want their fingers gnawed off by wild weasels. Time should keep flowing and the character on the phone should animate waiting and thinking until the user clicks the button, or the question times out and the caller hangs up.

I repeat: MODAL DIALOGS CAUSE MANY MORE PROBLEMS THAN THEY SOLVE. DON'T USE THEM UNLESS THERE IS ABSOLUTELY NO OTHER RESORT!!!

MISC NOTES

This design eliminates the following interface buttons: Function, Move, Sell and Info. Move, Sell and Info interface replacements are discussed in above section. The objects will not be categorized by function therefore the Function button is not used.

Sell and move do not work in Buy mode, according to my reading of this design, because in buy mode, clicking buys an object (or beeps that you can't buy an object for some mysterious reason), it does not select objects, so it's impossible to "select" an object in buy mode to pick it up, move it, or drag it to the control panel. Buy mode would need a state in which you were not buying anything, and currently the way to do that is to press the "move" button (the "sell" button doesn't currently work, except in certain cases, and only then due to a move tool design bug that also causes the object you're moving to be abruptly sold if you select an item from the catalog).

I don't see where the "Info" button has been replaced. Even given the ability to select an object in the world in buy mode, what do I do to get info on it? Page around the catalog until I find something that looks like it, and then press and hold that icon for a while? No way!

Getting rid of pesky buttons is not so important that we should make the interface hard to use and self concealing. For every command there should be a visual affordance on the screen that lets people find it. You can design all the shortcuts you want, but don't throw away the map. But there has to be something visual to point to, before a tool tip can pop up and suggest a shortcut.

The player in 1024x768 mode will be able to see the next extended set of objects without having to use arrow buttons. If there are still more objects to be displayed, the right arrow will appear on the extended CP. (There's not another panel of objects, rather the object panel is just wider with more columns).

How do you place objects on tables?
Should you be restricted to only placing certain objects on certain other objects where they belong?
Right now many objects like chairs have slots at their origin for people to sit in, but you never want to place another object in that slot or it occupies the same floor tile and intersects. If you put food on the chair, it probably goes underneath, and wouldn't even be visible "inside" of couch. Maybe you should not even be able to put food on couch. What about putting food on ground? Should that be restricted? Food processors should only go on counters. If you place them on the ground they should not work, so maybe you cant place them on the ground. How does the object placement tool tell you the placement restrictions of an object? The user should have some way of telling where they can place something, be it only on a counter, or only on the floor, or only outside or inside. What if an object has several slots to place something? How do you show that to the user? How do you decide which slot to drop into or to pick up from? How do you change things from slot to slot? There are some undocumented and undesigned keyboard shortcuts for dealing with slots, but they are bizarre and invisible and complex, and not something that I think should be exposed to the user. I think placeable slots should be very rare and/or very smart, to avoid exposing the user to all of these problems. Also, I say we get rid of the inside only and outside only placement flags, and maybe the "against wall" requirement, because they are only annoying and can be easily defeated by using the wall tool to change the definition to inside and outside. No way we want to program the wall tool to remove objects that don't belong, when it closes or opens a room to the outside. So why bother preventing people from planting trees indoors and putting refregerators outside? I actually KNOW somebody with a fridge outside, and my mom has trees indoors (small ones, though, but sometimes a large one around xmas).

SFX - Buy Mode Control Panel

Sfx of "ka-ching" is played every time an object is bought. When selling object back by Delete key or by dragging it back to CP UI, the addition of dollars back into account will play a different SFX.

CP and in the World object SFX will be specified when all game interface issues are set.

Observations from shopping for furniture in the real world:

- Used furniture stores and Goodwill are important sources of low cost, funky, one of a kind furniture, already worn. Miscelaneous selection, usually single items. No packaging, or even advertisements (for individual items). Categorized by type.
- Garage sales, flea markets, junk and salvage stores.
- New furniture stores.
  - Very often sell COMPLETE SETS of items: Table with 4 chairs, sofa/loveseat/stool set, bar and stools, matching shelving and tables.
  - Also these groups and individual items are grouped together into higher level "style" groups, like Queen Anne style furniture, Ikea style, building or material.
  - At an even higher level, furniture stores are laid out in different sections for different rooms.

Another kind of shopping, not to replace what we currently have, but to add another rich interesting realistic way to shop: Shopping could transport you to another place, a shop full of new furniture, goodwill store, garage sale, etc, where you pick objects to buy, collect them into a "virtual shopping basket" or something, buy them, and have them delivered home to your driveway, for you to then place in the house.

Scenario: Watching TV at home, and an ad comes on for a sale at "Fast Eddie's Used Furniture Store". Click on the TV to pop up a menu, one item is "Go Shopping at Fast Eddy's Used Furnature Sale". The currently selected person walks outside. (After they pass through the front door, the scene cuts to Fast Eddie's, to keep it ambiguous just how they get there, to keep the game moving along, since nobody wants to sit around watching the character wait for the bus or drive a car and look for a parking space.) Cut to entering the front door of Fast Eddie's. You can move around the store with the normal "go here" menu on the ground. When you're done shopping you can exit the shop by going outside, and go to other shops or back home. There are aisles of items for sale on display. Clicking on an item on sale pops up a menu that has a "Buy for $XXX" selection. Some items also have an "Inspect" item, that plays an animation of looking over the object and trying it out, maybe sitting down and reclining in a lazy-boy chair, or kicking the wheels so to speak. Your reaction to it might be based on your personal preferences, so different characters will like or dislike the objects. There should be some way for a group of characters to go shopping. You can derive some entertainment value from the objects, but not too much. For example, you can't cook a meal on a stove and eat, but you could have an entertaining evening by shopping for TV's, looking at them all, listening to sterios, but never buying anything. This way the act of shopping itself can be fun and useful, without actually spending any money! When you inspect an item, a window with an ad or catalog description might pop up. The salesman, Fast Eddie, might notice you looking at something, and walk over and tell you about it, demonstrate its features to you himself, or whatever. If you decide to buy something, and select "Buy", the item does not disappear, but a big "Sold" sign can appear on the item, and the "Buy" menu item changes to "Return" in case you change your mind not to buy it, so you can see all the items you're going to buy, and change your selection. A virtual "shopping basket" is another approach, but the items in the basket won't be visible (too complicated to draw, and not appropriate for big items anyway). When you're done shopping, you can check out to finalize your purchases, and leave the store, or just leave without buying anything. You might keep on shopping at a different store, or go home. When you finally go home, it cuts to you in the driveway, waving bye bye to the delivery truck, which is pulling off after unloading all your purchases in the driveway or front yard. Then you go into object placement mode (arch or buy, whatever), and place the object in your house.

Architecture

Control Panel UI

The Architectural control panel will include the common View Controls and Game Controls shared by all modes. The architectural tools listed in the table below will be graphically displayed (not textually as in Buy Mode). They will appear in a block containing two rows of six tools. When the player clicks on any of them (except Hand), the specific tool choices in that category will appear in a block to the right of the tool block. For each tool selected, the cursor will change.

Architectural Tools on the Control Panel

Tool	Tool Choices	Cursor in world	Function	Shift + Cursor (Extended Function)
Terrain*	1. Hill with up arrow 2. Hill with down arrow 3. Hill with horizontal arrow	1. Hill with up arrow 2. Hill with down arrow 3. Hill with horizontal arrow	1. terrain up 2. dig 3. level terrain	1. None 2. None 3. None
Water*	1. Water drop with up arrow 2. Water drop with down arrow 3. Pool 4. Diving board 5. Pool ladder 6. Pool filter	1. Drop with up arrow 2. Drop with down arrow 3. Pool in hand 4-6 Floating object in closed hand Bla on the hand.	1. raises water level (only works on water) 2. lowers water level 3. Drag out a pool 4-6 Place each	1. None 2. None 3. None 4. None 5. None 6. None

			object	
Plants	4-6 trees 6 shrubs 6 flowers	Floating selected plant in closed hand Bla on the hand.	Place plant	None
Floor*	X flooring styles	Arrow	Place floor	Fill entire room with selected floor. Shift is room.
Wall*	1. Wall 2. Fence 3. Banisters 4. columns	1. Pole (hammer removed) 2. Small pole for fence 3. Small pole for banister 4. Floating column in closed hand Bla on the hand.	1. Drag out a room. Shift is room??! Inconsistent. 2. Drag out fence 3. Drag out banister 4. Place column	1. Drag out a wall only. Shift isn't room??! Inconsistent. 2. None 3. None 4. None
Door	X door styles	Floating selected door	Place door in a wall.	None
Window	8 window styles	Floating selected window	Place window in a wall.	None
Wallpaper*	16 wallpaper styles (10 inside, 6 outside)	Looks like door frame until hits wall – then displays pattern	Place wallpaper on a wall.	Place wallpaper on all walls in the room. Shift is room.
Stairs	8 stair styles	Floating selected stair in a closed hand Bla on the hand.	Place stair	None
Fireplace	X fireplace styles	Floating selected fireplace in a closed hand Bla on the hand.	Place fireplace	None
Roof	4 roof styles	None	Roof style applied automatically then pops house	None

			into outside view (roof).	
Hand (Move Objects)	None	Cursor changes to open hand. Bla on the hand.	To move objects in the world.	None

*** Have a running total. See Rollover Cost section below.**

Architectural Defaults

Upon first entering Build Mode, the following are the default architectural controls:

- **independent of mode**
- **independent of mode**
- **IS the mode!**

Upon returning to Build mode after having played and left it for another mode, the game will remember any submenus that were in use when Build mode was exited.

Rollover Cost and Running Totals

Each tool will display a unit cost when rolled over in the control panel. Additionally, some tools will display a running total as part of the cursor when they are being stretched out in the world. These are marked with an * in the Architectural Tools table.

Tool Tips

Each tool will display its name when the mouse rolls over and No wait, or very short!. It will work in the same way that Word We are not designing a word processor! tool tips does. Tool tips are not only for novice users in the tutorial, experts expect them in this day and age. will be turned off if the tutorial mode is turned off in Game options. Tooltips should be de-coupled from tutorial mode. The game options dialog (advanced options, not basic options) could have a slider to adjust tool tip time from 0 to infinity (off).

Undo and Redo

These two buttons need to be added to the left of the general tools. The two current suggestions for look are:

1. **Write them in vertical text Eech.**
2. **Make them icons like the undo/redo buttons in Word. We're not designing a word processor. Icons + text like in Explorer. (We're not designing a web browser either, but we want to use icons + text because it's a good design (recognizable + readable), not because it's similar to some other program.)**

Charles needs to design when working on the entire layout.

Miscellaneous Architecture Issues

- **Will advocates supporting the intellimouse input device in this way for Architecture Floor selection mode: each "click" of the intellimouse wheel will put up an Arch level. Starting on the first floor, click one will put up walls on the first floor, next click will put on floor of second floor, then second floor walls, then roof. Reversing the wheel will take off each of those layers. What about dynamic cut-away mode? Is that skipped over by the wheel? What if you're already in dynamic cut-away, do you lose it? Remember you can press the wheel down too. Perhaps pressing the wheel button could pop up a pie menu that lets you select what turning the wheel button does, including level, zoom, speed, people selection, etc. The wheel is so useful it's a shame to use it for only one thing.**

- **No shearing of terrain.**

- **The hydrological model will be included, is not yet designed.**

- **There will be 200 levels of terrain and water.**

The Storyline

A unifying story for The Sims is being sought on several fronts;

- Will and Jim story
- Pet Fly productions as game design and story collaborators
- Ron Cobb as occasional, drop-in consultant

Neighborhood

Art

- **Houses will be actual representation of player-built house; the artist will create thumbnail-sized set of architectural objects for programmatically built neighborhood scene.**

Design

- **Success/failure or game is reflected graphically in nbhd screen**, but not yet designed. Also not yet designed, but will be included in game are: saved games display differently than no yet played houses, how to evict families, or to place a homeless family, explore/preview modes.

Other game elements

Disasters

- Fire – do objects have a combustible score? Do objects need a burnt, charred graphic state? RAT HOLE! This will take a LOT of artwork and programming, and never be very convincing. Fires are hard to simulate, the way they interact with the burning object, which changes over time and effects resulting the flames and smoke.

- Other disasters: Blackout, others to be designed
- Electricity supply and usage model. No wires or plugs, maybe a meter. Power outages and brown outs in the whole house, and monthly power usage bills. Lightning strikes and power surges can zap some sensitive appliences that are operating (computer, tv, lights, etc.) Washing machine can blow up and send power surge throughout household.
- Water model. No pipes or sewers, maybe a meter. Cold and hot water, water outage in whole house, droughts, water bill every month, hot water heaters with limited capacity, that recharge slowly and use electricity, tub and shower and dishwasher and washing machine compete for hot water. Interesting game of making sure you have enough hot water for everybody in the household to have a hot shower or bath without running out. Cold showers and baths make people crankey and uncomfortable (though less horny). The "testing the tub water" animation can have a wonderful failure reaction to cold water. Somebody taking a long shower can scream when they run out of hot water and jump out of the shower. This is all very easy stuff to implement given the current system.

Events

Events such as marriages, etc. will be included in game, not yet designed.

Boyfriend seranades girl outside of window. Girl sneaks out bedroom window late at night. Hillbilly Poppa with shotgun comes over with clergy and forces boy to marry his daughter. Bad husband has to sleep in doghouse. Polygamist weddings in Utah, where the first wife presents the husband to the second wife, who presents him to the third wife, etc. Divorced couple saws house in half. Nondenominational same sex mixed race tatooed amputee weddings in California.

Autonomy

Interruptibility

Vocabulary
Player directed is when player gives character instruction

Other character initiated (OCI) is other character in game interaction with selected character

Interaction priority system

1) player directed interrupt of selected character
2) other character interrupt of selected character
3) autonomous behavior

When player directs character interaction

- The player may queue up as many interactions as they wish. There is virtually no limit to the number of interactions in the queue. These interaction, however, are not visible to player.
- Interaction routes and completes one action and then moves on to the next player directed interaction.
- If there is no player directed interaction in the queue, the current interaction loops, decreasing the priority on each loop, until the priority is less than that of the other character initiated interaction in the queue

When other character initiates interaction

- A player directed interaction can interrupt a current OCI during routing or after completion of action.
- A player directed interaction flushes OCI in the queue.

- The interaction routes, completes on action, loops decrease the priority on each loop until the priority is less that of the next OCI in the queue.

Routing interruptibility

- If character is routing to a player directed interaction, then routing is not interruptible.
- If in autonomous mode, a character is interruptible.
- If other character has initiated interaction, no selected character routing is required; the other character has routed him/herself over to the selected character to have the interaction.

Thought Balloon as Visual Feedback

Originally the Visual Queue feature was thought of as a mechanism to acknowledge that the selected character "got the message" from the player of what to do. Since the display size, editability, icon choices and other thorny problems could not be solved to our satisfaction, the

alternative of pie menu animated "yes" of the selected characters head will serve much of the same purpose in a simpler way.

- **When the player gives a command to go do an interaction (i.e. eat), the selected character's head in pie interaction menu will shake head "Yes" to indicate that they have understood the command. (this way , if the character is routing to player's first command, there is no mid-route thought balloon that breaks in momentarily. The acknowledgment of command lives in the pie menu.)**
- **As the selected character is routing to go do that interaction they will display the refrig in a thought balloon. The thought balloon should have a different shape than the motive balloons. Only selected character carry around (display) their thought balloons while routing. Neighbors and other family members do not show their thougth balloons in the same way. By refrig you mean any object? Do all objects have thought icons? What do kiss and goto here and door portal icons look like, or are they incisible? Why not give every ACTION its own icon? Much more informative. Or if each object had an icon, you could display the object icon before the name of the action (from the menu).**

- **The icons in this thought balloon will use the catalog size object graphics (no larger than 45 x45 pixels**).

Advanced Interruptibility feature

For those advanced players who wish to flush the player directed interactions, a keystroke will be noted in the manual.

Feedback

Sound design

SFX

- Objects and their animations in World will use .wav files.
- People, Architecture and Objects Control Panels will have some shared SFX for unified UI, and some specific SFX.
- It's possible to play sound effects at different speeds to change their pitch. This will allow us to get more out of the sounds we include.

Music

Freshness will be considered for music track.

I have some ideas about how the music could effect the game, that I will write up more completely later. In a nutshell, the people in the house could have a cd or record collection to choose from, each record an object that has the sound (audio wave and/or midi) and a "moody" track synchronized with the music. Playing the music also plays the moods into the environment that the people pick up on. Music can subtly effect how people react to the environment, objects, and each other. It can effect their motives and even their skills temporarily. For example, you might be able to clean the house better and faster if you put on some up temp bouncy music. The player should be able to assume the role of disc jocky on the radio, and play from another larger library of music and commercials, that effect the peoples moods and buying habits. The TV of course is another source of mood altering temporal media, with commercials and shows that should effect different people differently. But the most important part of this idea is instead of the game effecting the music that's played, the music effects how the game plays! The ultimate way for the user to effect the game via music, is to insert one of their own CD's into their real computer's CDROM drive, and the game would recognize it, and start playing it (maybe with a simple cd player interface to select the song). There could be a database associating the unique ID number of the CD with a table of contents and "moody" tracks that tell how the song effects the peoples emotions over time, with "percussion" events at dramatic moments of the music that can trigger arbitrary events in the game (like provoking a fight that was brewing, or triggering an orgasm at just the right place in the song). We hire monkeys to listen to well known CD's, and enter time synchronized tracks with semantic meanings in Max (like note tracks, and user defined numeric tracks) or some other timeline editing tool). Put the database up on the web for instant retrieval, so when somebody sticks in a new CD, it downloads our "moody" tracks that go with it, and it starts playing and effecting their game! Streaming emotions over the net! Eventually there should be an end-user tool so people can record their own responses to music as moody tracks they can use in our games. This mechanism could be used in all kinds of games, to varying degrees of effect. I'm not saying that music should be the **only** way to control the game – it's more like a subtle background effect, but there certainly could be a scenario where you try to accomplish some task (like taming a wild beast) by using only your musical taste and timing. The **real** bottom line benefit is that you get to listen to your OWN cd collection of music you want to hear, instead of being driven crazy by the repetitive music bundled with the game.

Voice

We will use either

- a more thoroughly scripted version of the Charlie Brown speak that was used in June milestone.
- An obscure language (Urdu, esperanto, Navajo) for entire script

Professional voice talent will be used.

Before the storyline is set, it is difficult to say how many voices we'll require for characters. Currently, we are assuming one adult male, one adult female, one kid boy, one kid girl for VO. This assumes that all onscreen characters that are adult men will have the same voice.