Modeling and Navigating a Simulated Semiconductor Fabrication Laboratory using a Game Engine

Preliminary Report

Craig Brown and Walt Scacchi
Institute for Software Research
and
Game Culture and Technology Laboratory
University of California, Irvine
Irvine, CA 92697-3455 USA
wscacchi@ics.uci.edu

12 October 2007

Purpose:

The purpose of this system is to assist in the identification of spilled puddles of liquid in microchip, semiconductor, or nanotechnology device manufacturing plants (FabLabs). This is an important problem to solve because while some puddles in the plant may be relatively harmless (leaking water from a fire sprinkler pipe), others may be potential biohazards (any of the over sixty dangerous acids, solvents, caustics, etc.). This danger means that any puddle not immediately recognized must be treated as a potential biohazard. This may cause the FabLab to be closed down until a cleanup team can determine the puddle’s origin, composition, and means for remediation. This may result in a shut down of the FabLab down for the remainder of the day, resulting in monetary loss for the company that owns and operates the FabLab. Being able to determine the composition of a liquid spill without shutting down the plant and waiting for a cleanup intervention effectively increases the daily utilization and productivity of the FabLab.

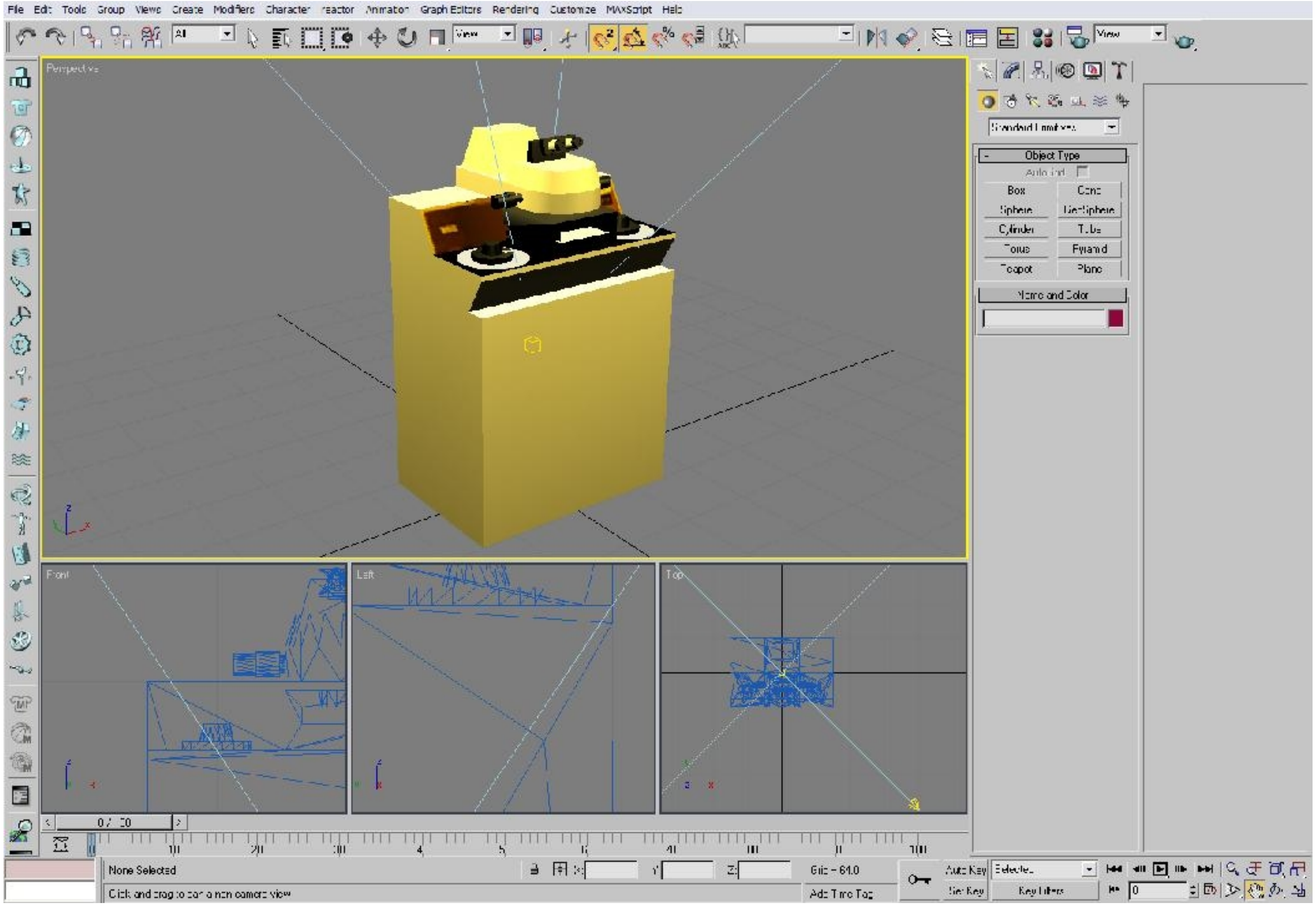
Features:

Background

We have used the *UnrealEd* 2.0 to recreate a microchip manufacturing plant. UnrealEd is a software development kit (SDK) and content development or modification environment for the Unreal 2.0 Game Engine (circa 2004). The Unreal Tournament 2004 game engine (packaged with the *Unreal Tournament 2004* game that retails for $30 or less) allows the user to construct/import, navigate, and interact with object located within a FabLab, as well as with in-FabLab characters, that are also visually modeled and assigned behavioral capabilities (via character animation scripts) via a custom character model (for example, an Intel employee in a “bunny suit”). Although FabLab objects and textures for their visual appearance can be modeled using the UnrealEd SDK, it is often more productive to use a more capable computer-aided design (CAD) modeling package like Discreet 3DStudio to model constructed objects and characters. In our efforts below, 3DStudio is used to create different FabLab objects, which are then placed into a FabLab layout using UnrealEd.

Models

We first start with an overview of some of the objects that have been modeled. The following screenshot depicts the modeling and texturing of an object used to depict a scanning electron microscope.



One overall goal of our effort is to model, texture, and display the following FabLab facilities and possible spill dangers:

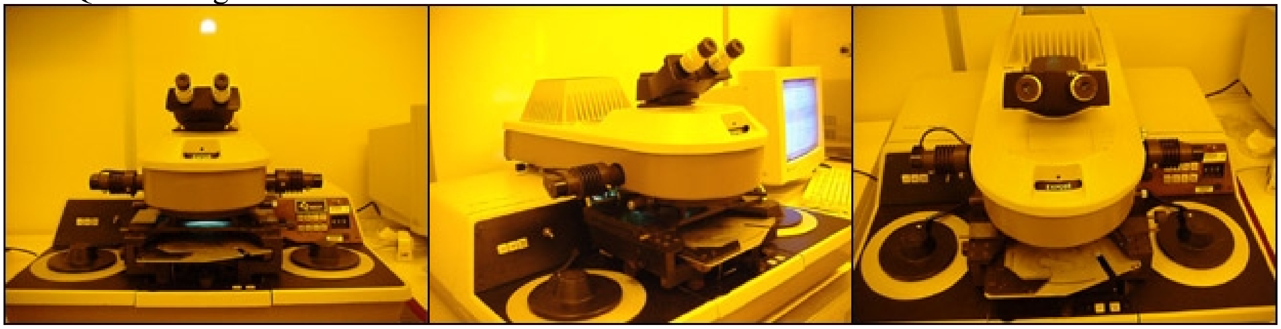
- FabLab manufacturing equipment, machines or devices
- Pipes (sprinkler, coolants, etc)
- Robotic carriage tracks
- Human ignorance

In what follows, we focus just on the first of these items.

Many of the models used will be either from the standard UnrealEd package, or are created by using the built-in CSG geometry and texturing them with standard textures. The rest of the models will be created custom using 3DStudio Max 7. The model shown above was created with 3DStudio Max.

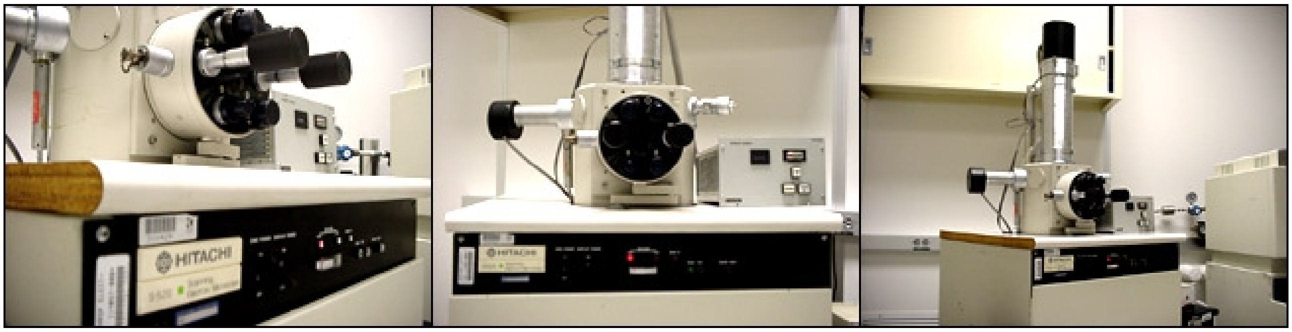
The virtual FabLab we have created uses the following equipment models, whose images are taken from the UCI Integrated Nanosystems Research Facility website.

Quintel Aligner



Technics Plasma Etcher

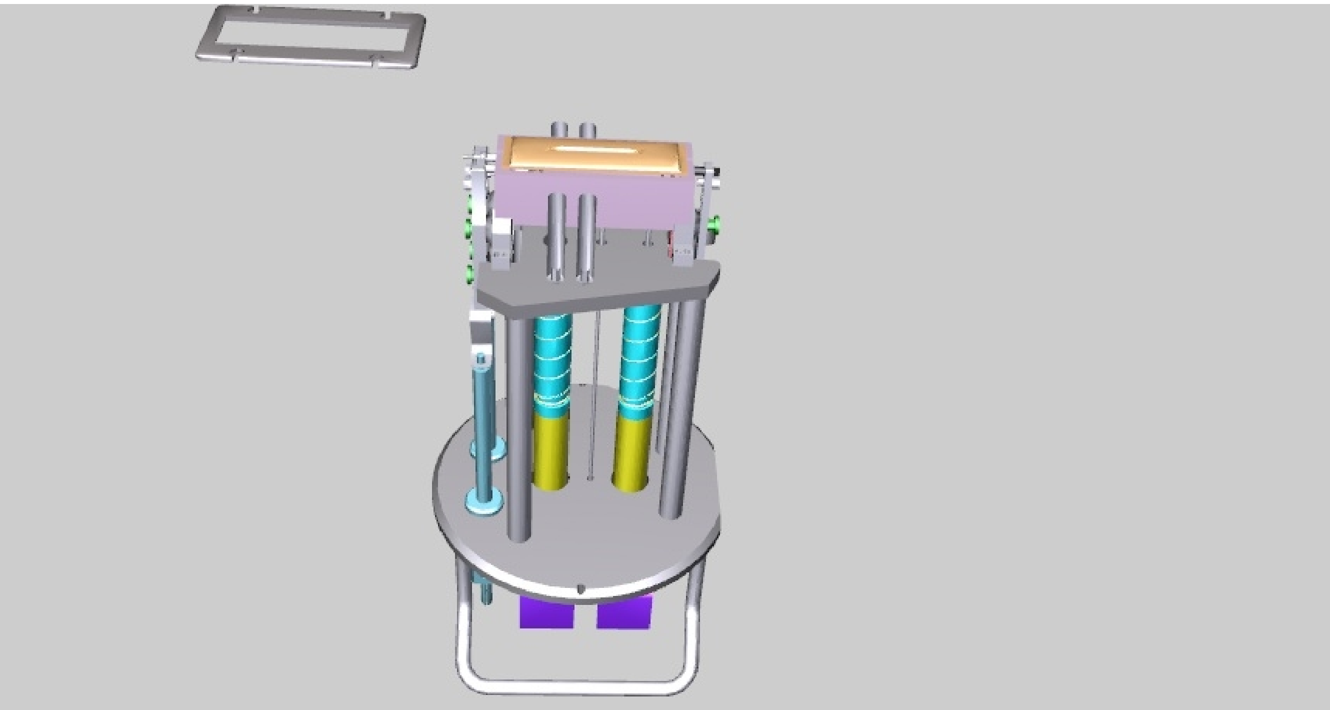
Hitachi Scanning Electron Microscope

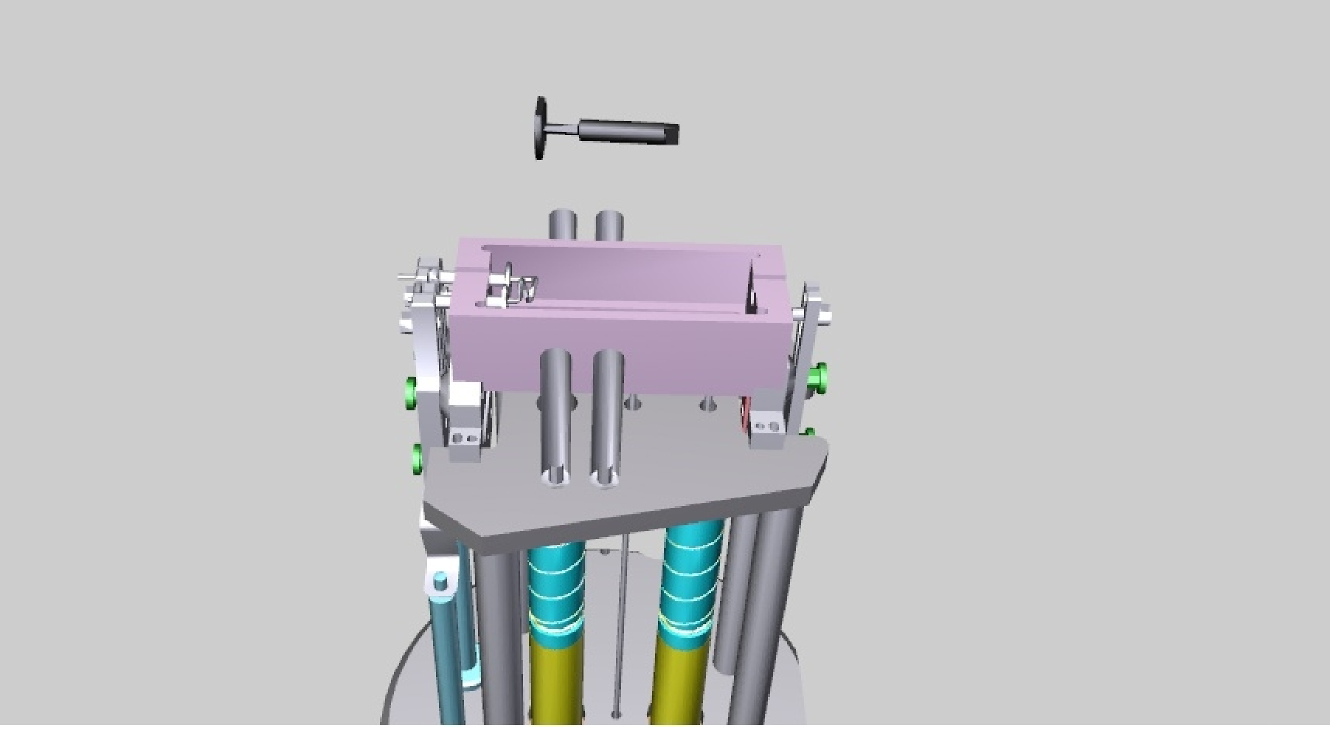


The system can be modified to recreate any FabLab layout and portray any set of machinery that can be modeled as 3D objects. Actual engineering design models of a device or piece of equipment would enable the modeled object to be (a) disassembled/reassembled for service purposes by adding an corresponding disassembly/assembly animation script coded in UnrealScript, or (b) functional or behavioral simulation (using an external simulation package). The system could easily be modified to function for other manufacturing plants.

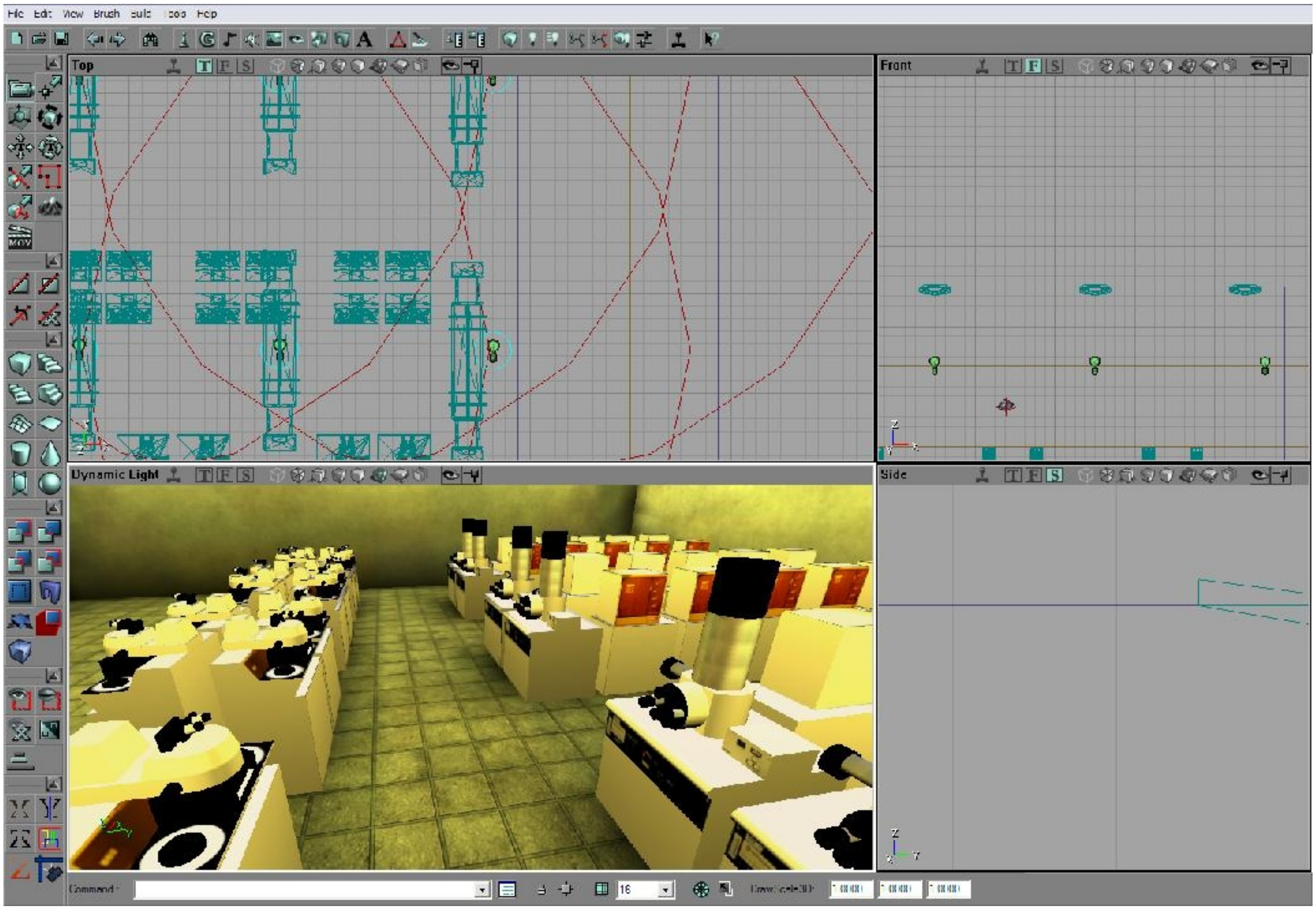
Beyond this, if actual 3D CAD models of FabLab devices were made available to us (in a standard data exchange format that UnrealEd can import), then we would be able to model not only the external appearance of devices, but also their internal subsystems. For example, if the scanning electron microscope incorporated a subsystem that was separately model, that subsystem model could be included within the FabLab 3D device model like those we have created. It should further be possible to also animate the subsystem's disassembly/reassembly, for instance, in response to a device service request or device fault detection.

The following is a 3D FabLab device subsystem model that features a partial subsystem disassembly showing a subsystem component.

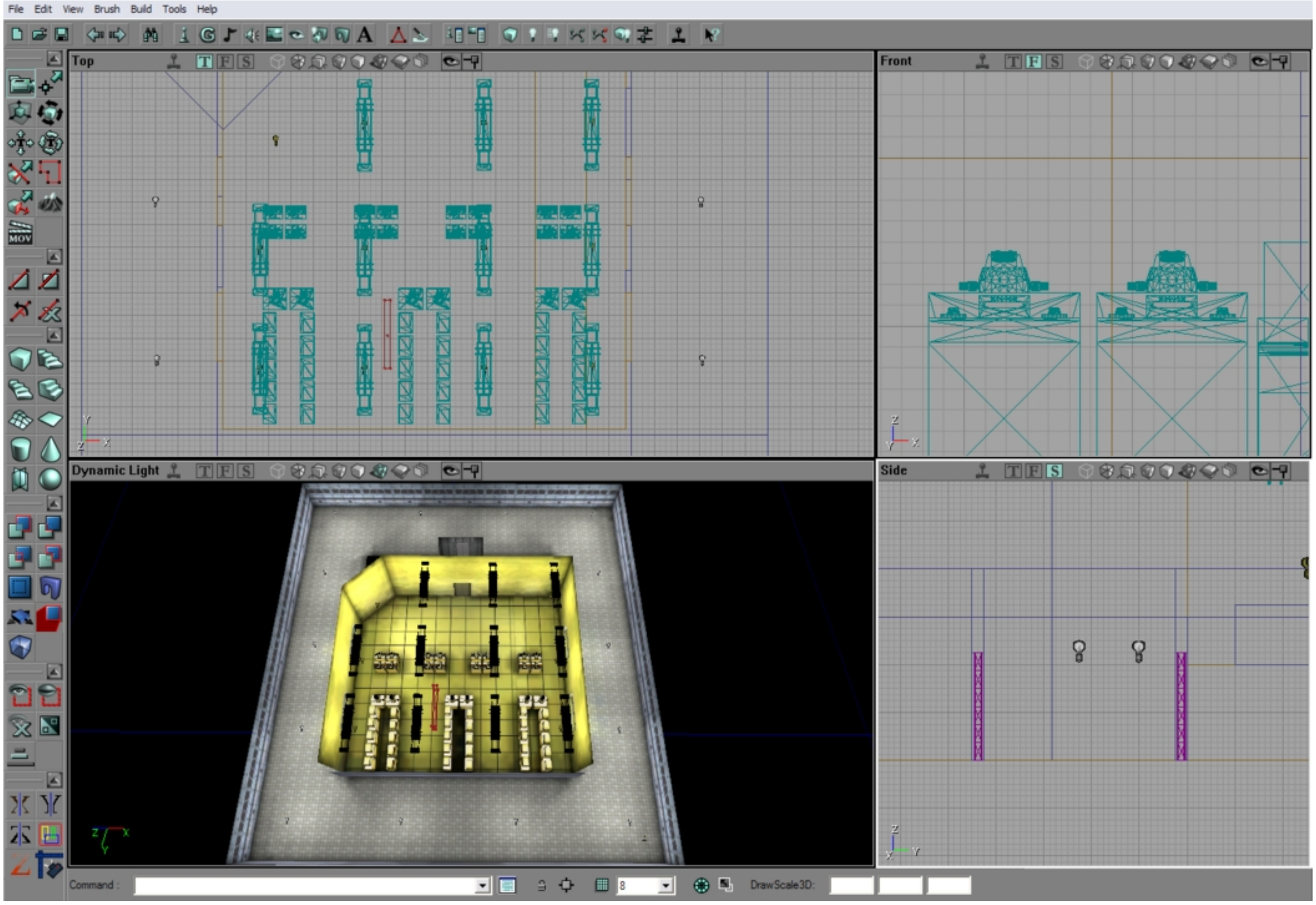




Moving forward in the FabLab modeling process, the following screenshot depicts work in progress in the placement of equipment and overall layout of the FabLab using UnrealEd.



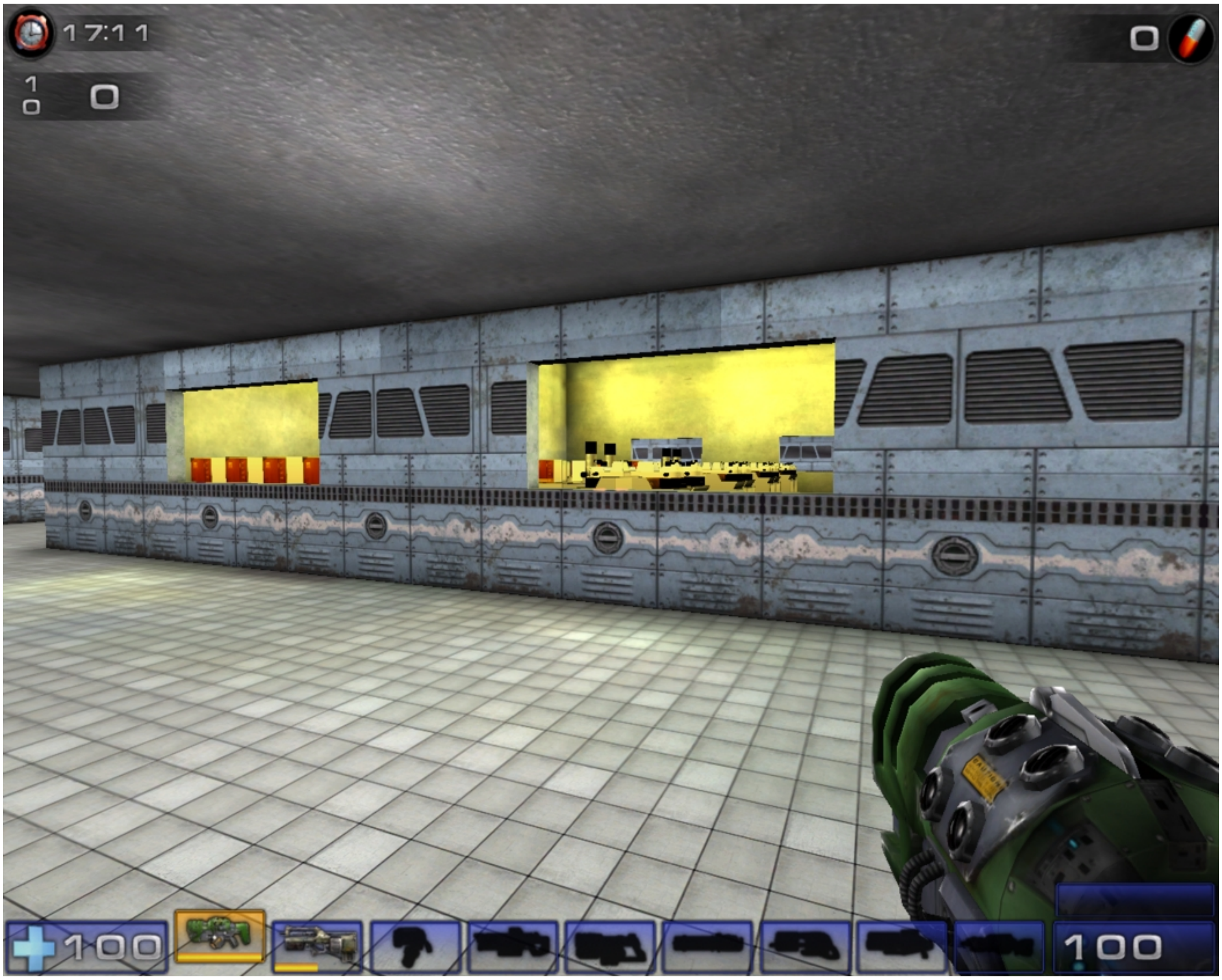
The next screenshot now displays an overall perspective view of the FabLab situated within a larger surrounding hallway, to suggests it location within a larger building.



Walkthrough

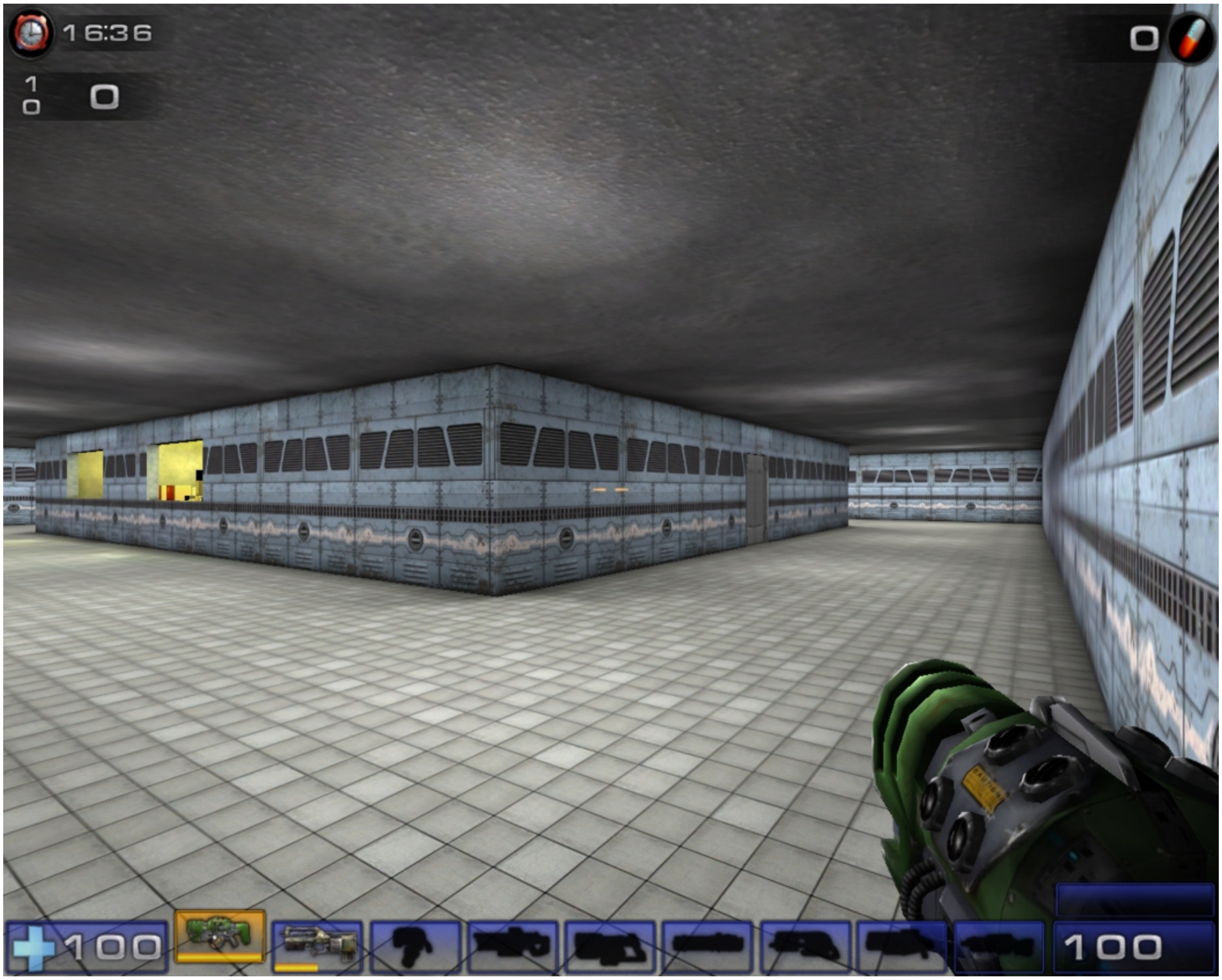
At this point, we turn to provide a series of images that depict an interactive navigational walkthrough of the FabLab that we have modeled, using the facility devices identified above. We start from a location (or viewpoint) in the hallway a larger facility as we approach the FabLab.

From the hallway, we see the FabLab has windows that allow for a hallway-based view of activities or circumstances within the FabLab.

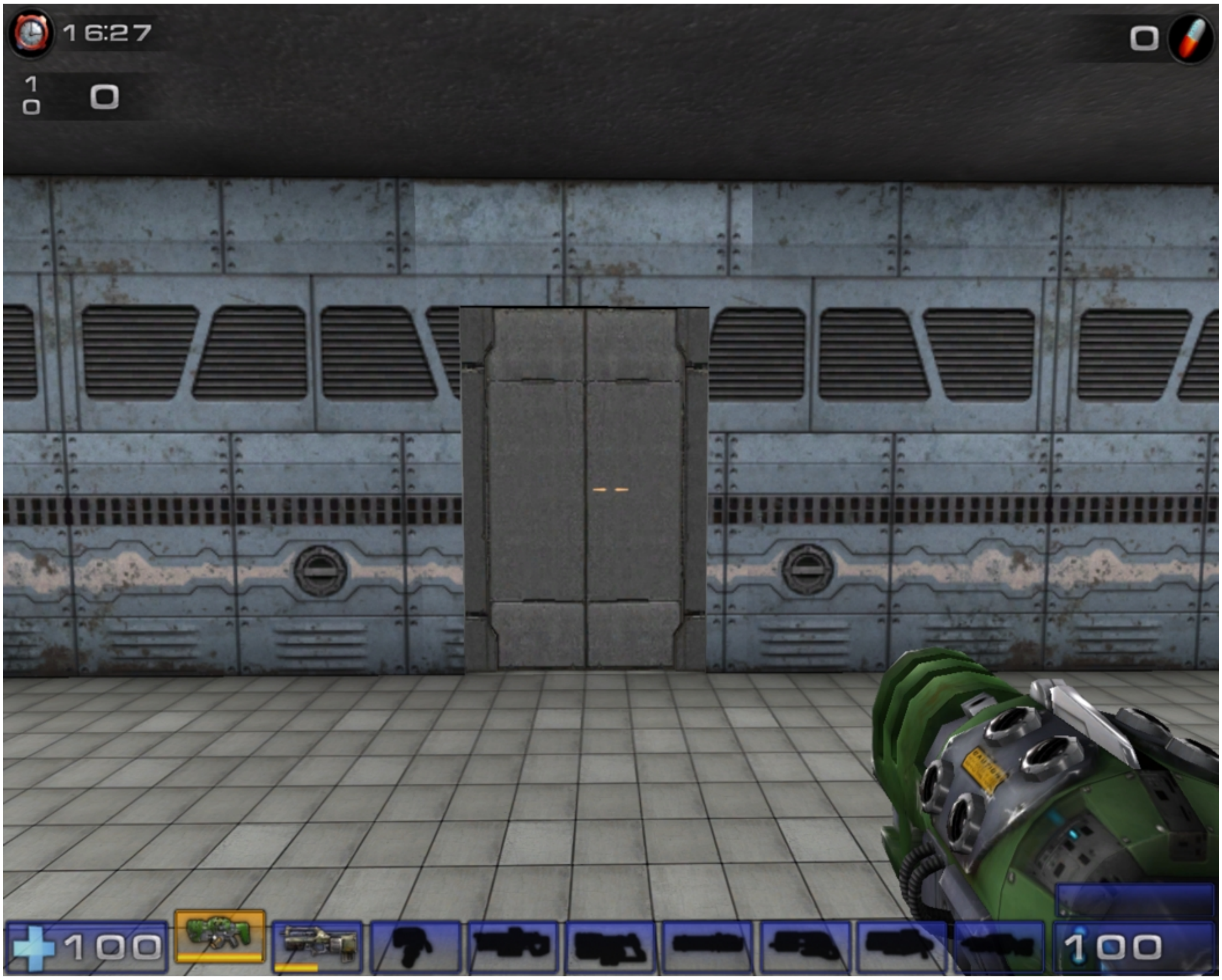


Information indicated at the top and bottom of the screenshot, and those that follow, are extraneous items from the Unreal game, unrelated to our demonstration. However, the countdown timer in the upper-right corner may be of value in the context of time-limited problem-solving tasks.

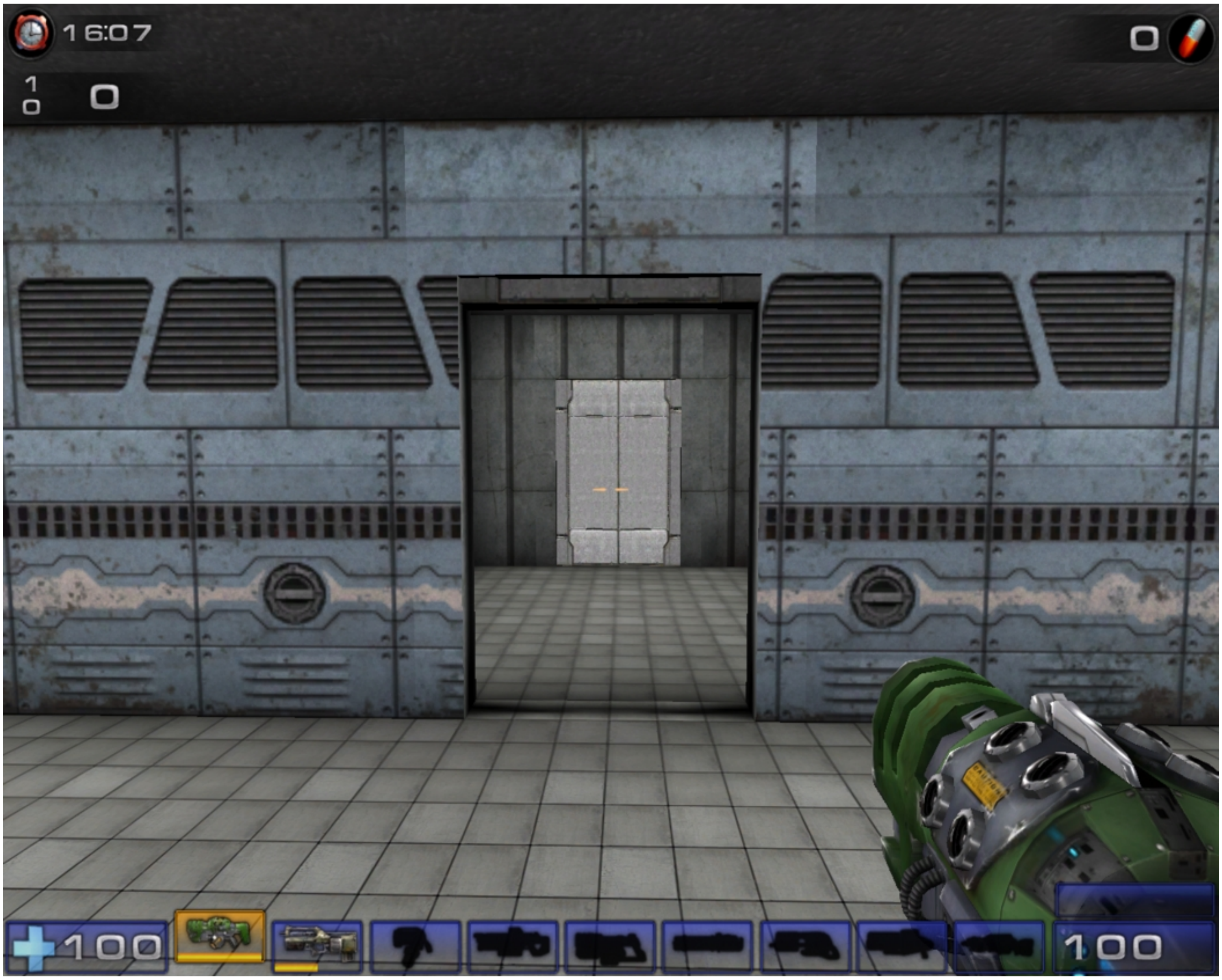
Next, we move away from the window and around the hall to see the door entrance to the Fablab (grey vertical doors).



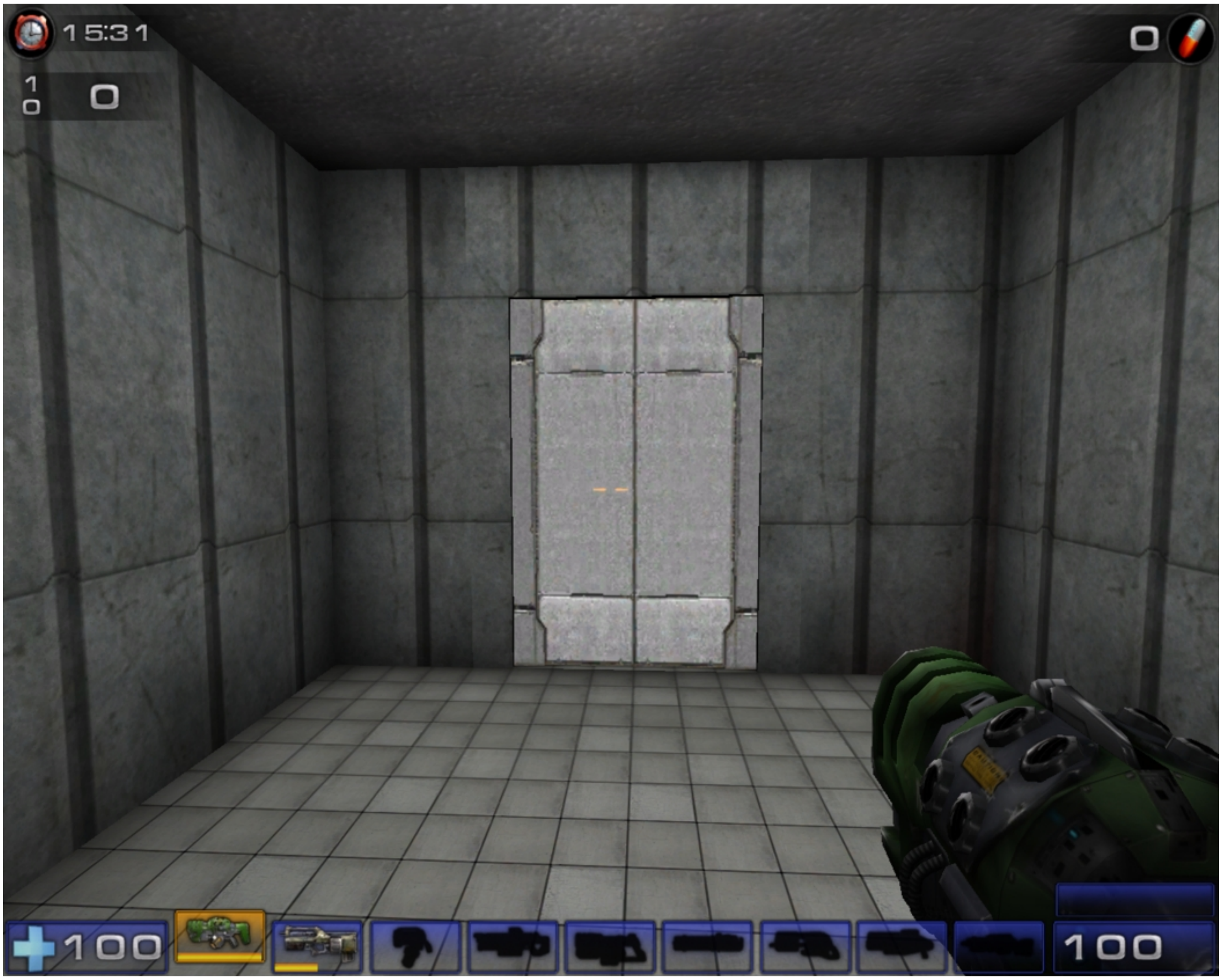
Now facing the door to enter the FabLab from the hallway, as shown below.



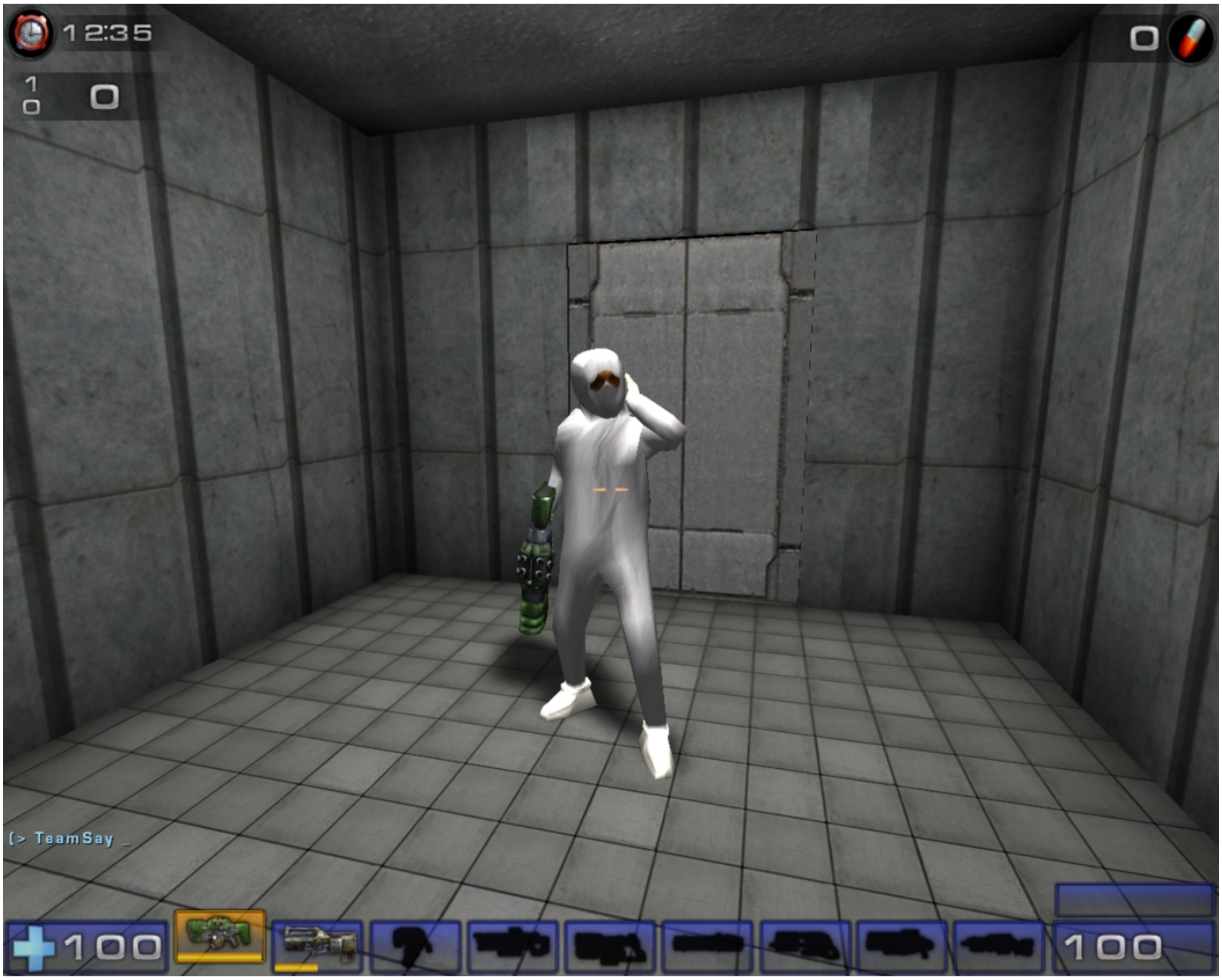
The outer door to the FabLab opens, reveals an interlock room where a cleanroom bodysuit ("bunny suit") is acquired.



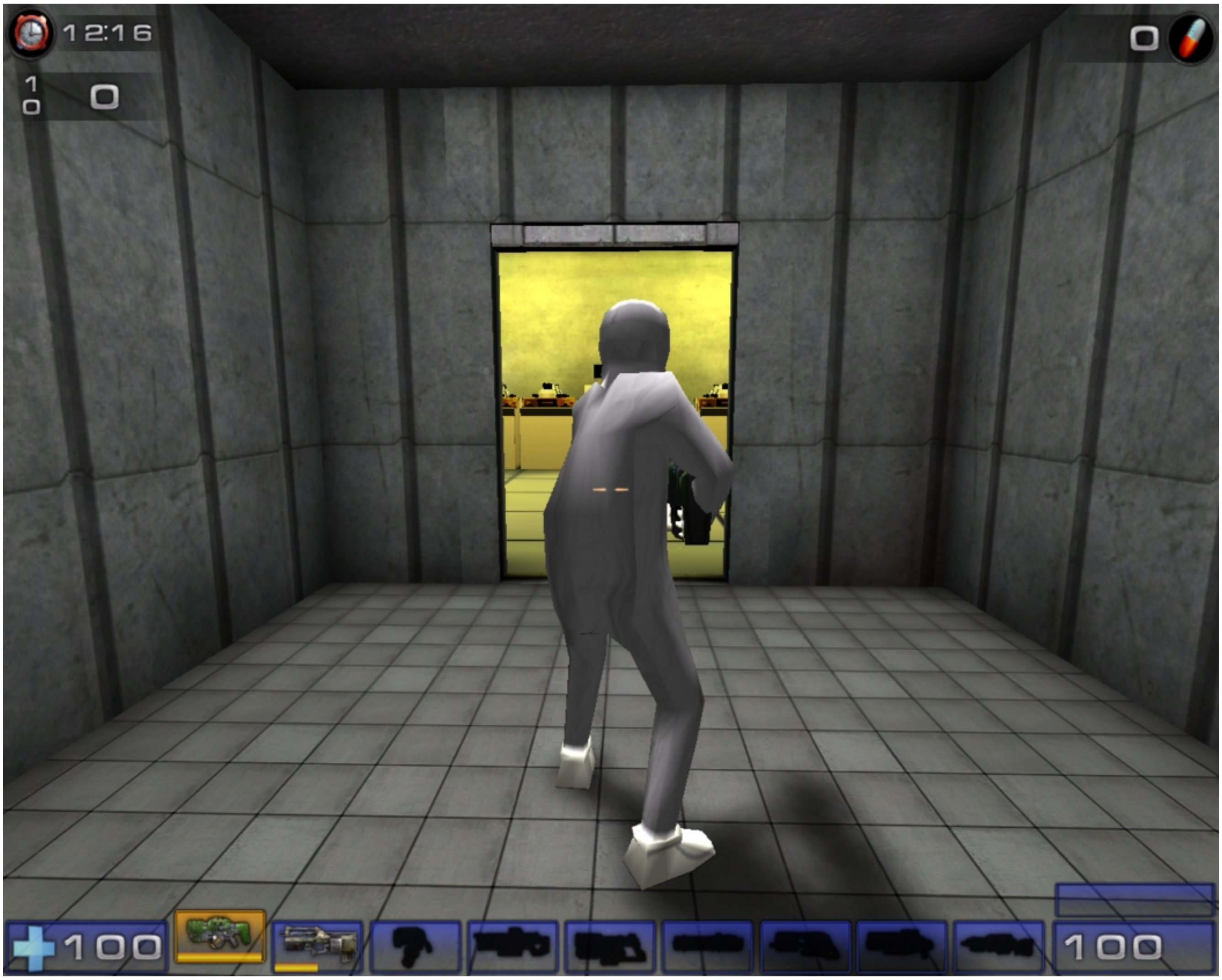
Inside the interlock room, facing the inner door to the FabLab.



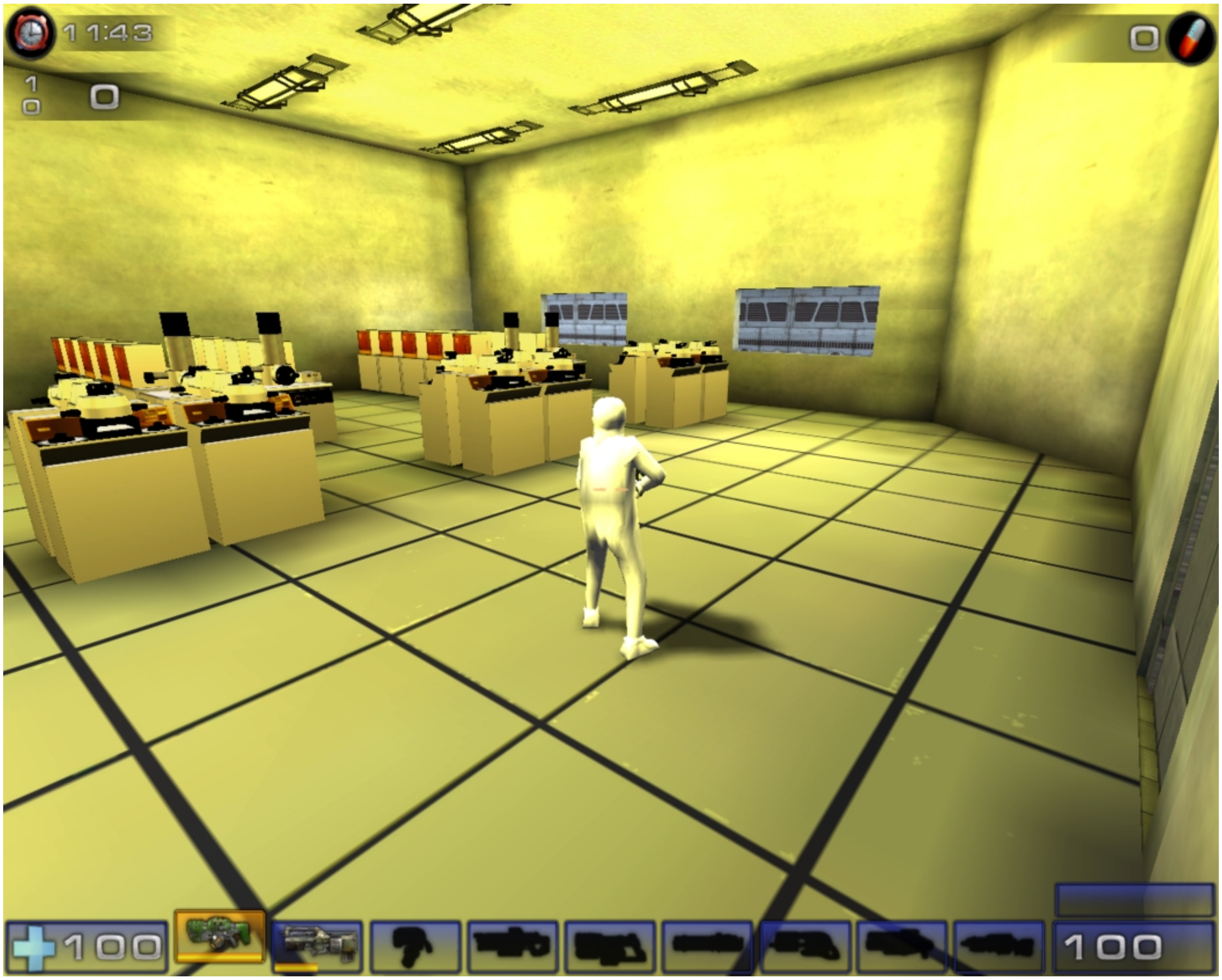
FabLab worker (as a fully articulated, user controlled character/avatar) wearing a clean room body suit, shown next.



Preparing to enter the FabLab in a clean room body suit.

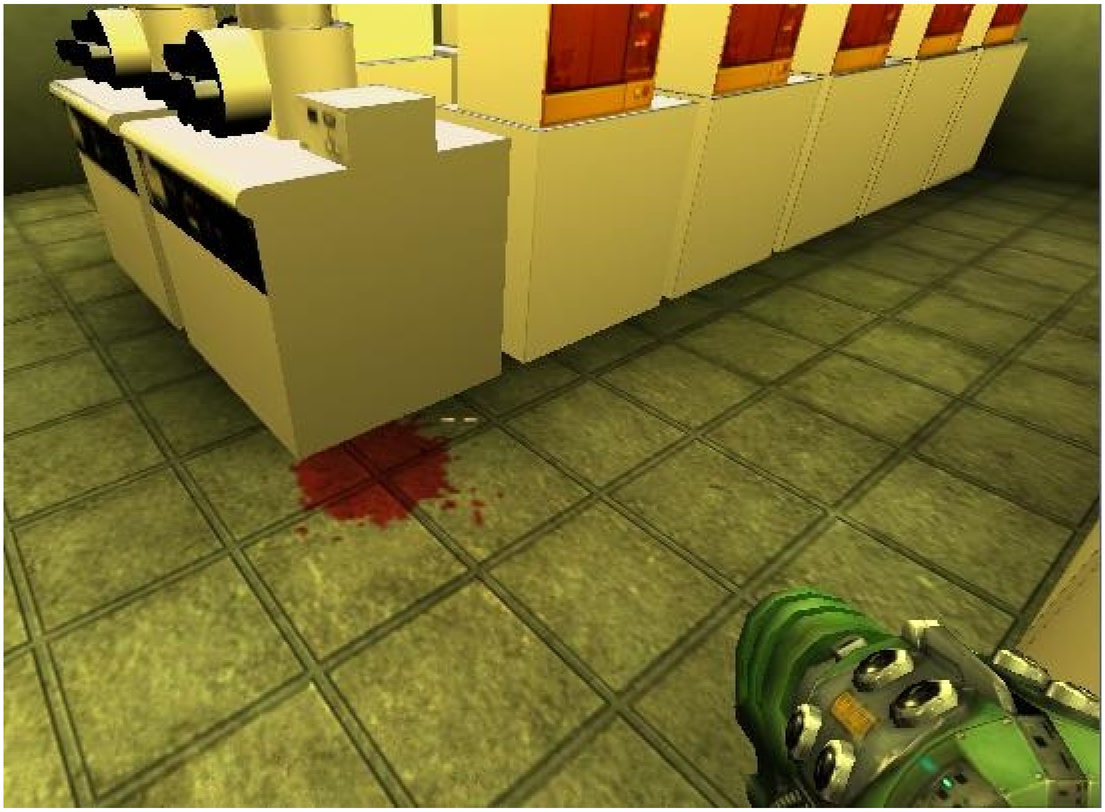


Inside the Fablab, noting windows into hallway in the distance, and semiconductor fabrication facilities nearby. The room color (or background illumination) results from an attempt to replicate the color-safe lighting used in a FabLab.

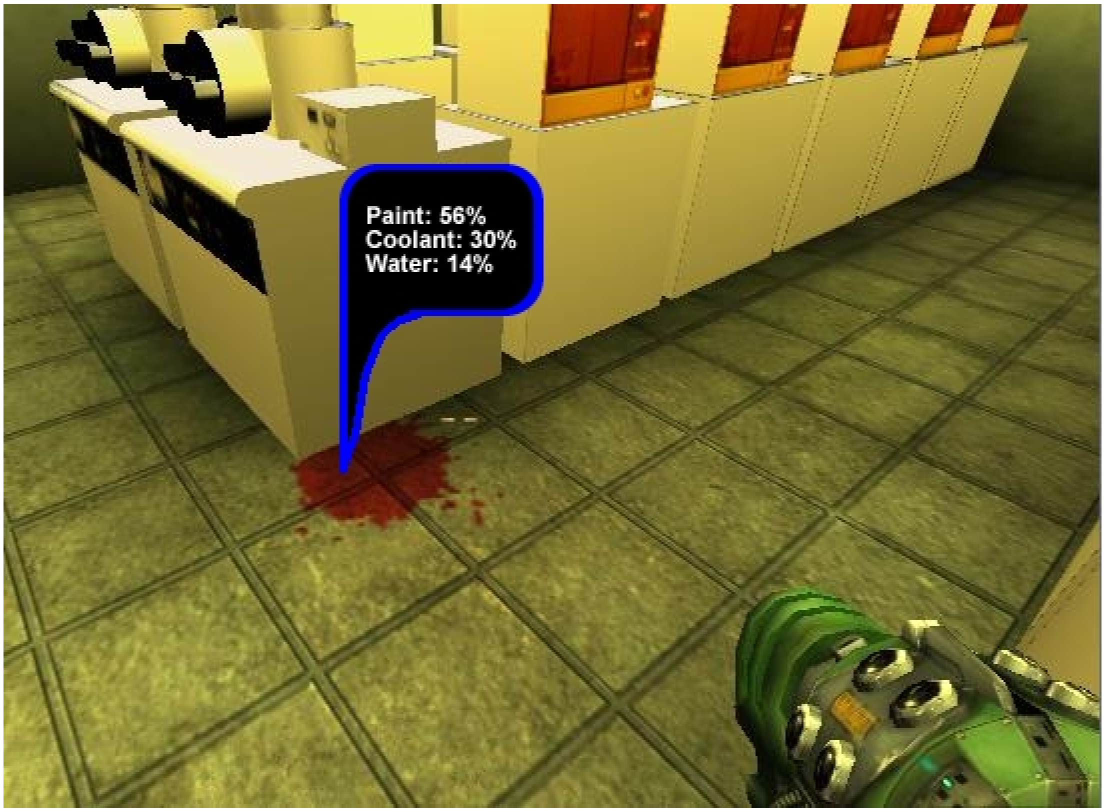


The Spill Squirtgun

As part of a training regimen for teaching workers how to safely diagnose a liquid spill in the FabLab, the trainer/worker is provided with a virtual device that enables her to place a liquid spill anywhere in the FabLab where such a liquid spill might occur. Spills are placed using a specially designed and programmed "spill gun" (the Squirtgun). When the trainer/worker fires the Squirtgun, it places a spill puddle wherever the reticle is aimed.



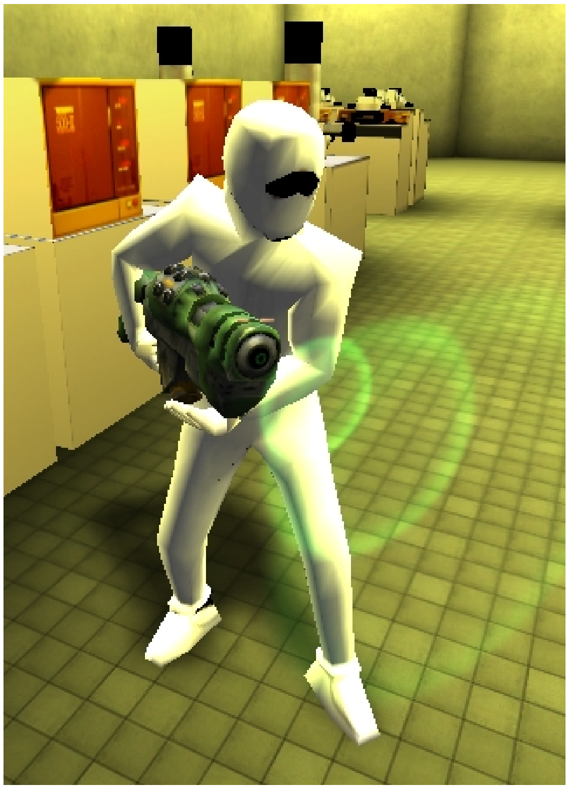
The Squirtgun (shown in the lower right corner of the screenshot above) has a simple targeting reticle, and no recoil when fired. On impact, the puddle uses its location to calculate the most likely composition a puddle in that location would have, and displays this information.



A limited number (to be determined at a later time) of puddles may be placed at once; when the limit is reached, the oldest puddle (and all its information) will disappear.

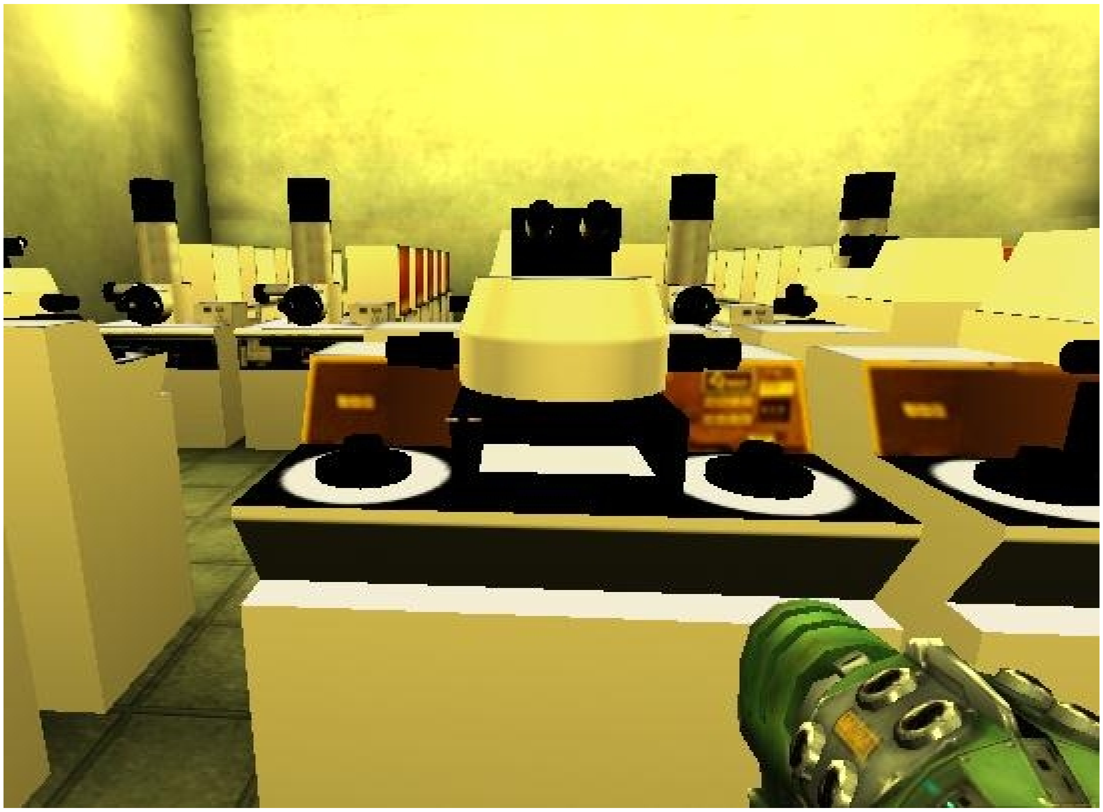
How to use:

The system is very easy to use, as all that the trainer is required to do is place a puddle. The user navigates through the plant, using standard Unreal Tournament 2004 movement ("W-A-S-D" keystrokes to move forward, left, back, right, and mouse click to look), and fires the Squirtgun (left mouse button) to place a puddle in an appropriate spot.



The spill puddle will appear where the Squirtgun is aimed, and will display the three most likely compositions of the puddle. If there are fewer than three possible composition/source pairs, then the one or two possible will be displayed, leaving the third space blank. If there is no possible composition/source pair for a puddle's location, the puddle will display the message "No composition or source found".

Preparing to place/shot a spill on a piece of FabLab equipment, in order to help determine what kinds of known liquids might be associated with a particular piece of equipment, given its overall 3D location within the FabLab. Note, some equipment might not normally have spills associated with their operation, but such devices might have another unrelated liquid spilled on them (e.g., from a neary "leaky" device, or from some remote source, such as overhead plumbing).



End of walkthrough.