Design of a technique for selecting and manipulating in 3D

Instructions:

On dispose d’une application 3D interactive. On souhaite fournit un moyen à un utilisateur de sélectionner et de manipuler des cubes 3D pour ensuite les positionner précisément dans une cible de la même forme/taille.

On envisage de comparer deux techniques d’interaction 3D : la main virtuelle et le ray casting (la même technique est utilisée pour la sélection et la manipulation).

La technique utilisée devra être compatible avec une interface de commande type flystick)i.e. 6dof+boutons), mais ne pourra être testée dans un premier temps, qu’avec le clavier et la souris.

Description of the System:

General Description

The system we are designing consist of an educational application for young students to get involved with technology.

The proposed solution is an application for virtually assemble a circuit and test students understanding, letting them learn from mistakes while working virtually and without risks, so later they can reproduce the work with real components.

The system itself has some predefined circuits, which can be selected by the user (for the first version a single circuit will be proposed). Each circuit is composed by different missing components which the student must place correctly. This components are placed in a box of components. During the assembly task the student can place or remove components from the circuit board. After finishing placing all the components he can test the circuit so as to see if it works.

In the real life, in almost any manipulation task the human “selects” the object to manipulate, he grabs it and later he can manipulate it: mainly move it from one place to another and rotate it in any direction.

In our application, both tasks: selection and manipulation must be solved. Some decisions have been made to achieve a valid solution:

- At the beginning the system is in **selection mode**,which means that the user can select any object from the components’ box or the circuit board (for components already placed).
- When the user validates a selection the object is attached to the virtual pointer (“hand”) and the application moves to **manipulation mode**.

- When the user “leaves” an object, the application moves to **selection mode**, allowing to place another component or modify a component already placed.
- On the **selection mode** the user can also execute other tasks, like testing the finished system, reset the system (remove all components), choose another circuit, etc. For this purpose there are menu buttons which are treated as “objects” regarding to the selection task.

Selection

For our application, the selection task will be done with the ray-casting technique, which basically consists of having a virtual pointer (2D in the x,y axis) and a ray which travels along the z axis. The idea is to choose the first object which intersects this ray, and to validate selection the user must focus on the object for about 0.5 seconds. When the validation is correctly done the object is attached to the virtual pointer (in a animated way, so as to avoid a brutal transition). The ray will be slightly orientated so as to be visualize by the user.

Manipulation

Having an object attached to the virtual hand, the manipulation task (rotation and translation of the object) can be done by getting the correct orientation and translation.

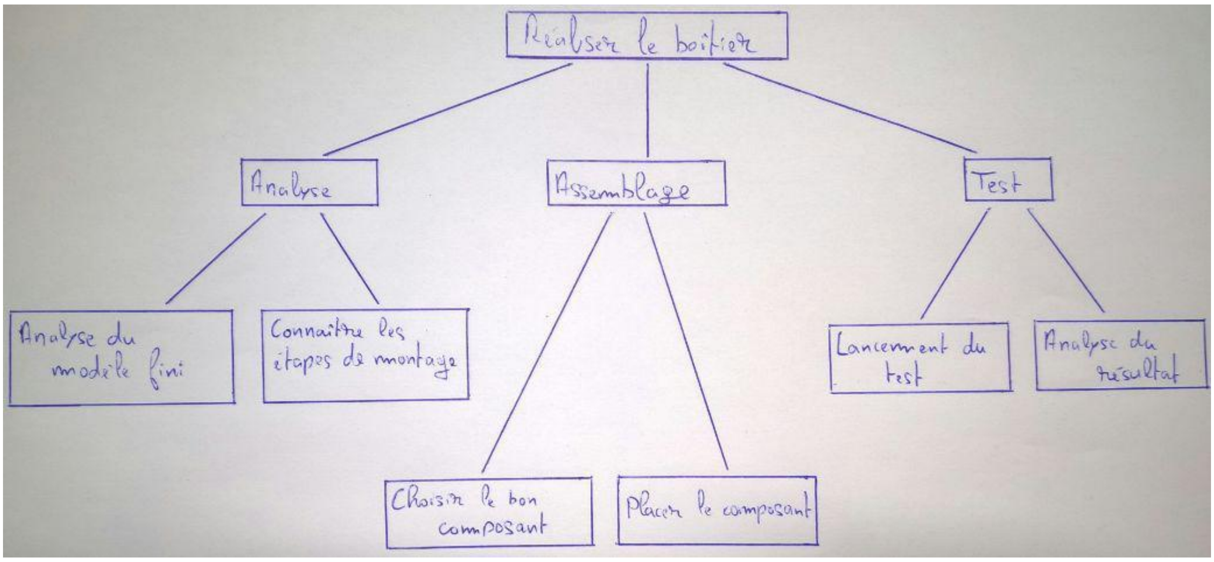
To place an object, when it is correctly located in the board (or the trash bin) it is illuminated (color changes) and the user must stay for 0.5 seconds on that position to validate placement.

- **Translation**
  - **Mouse + Keyboard:** The mouse position (2D) is used to place the virtual pointer in the ***XY-plane***.The same Raycasting technique is used to translate the object.
  - **Flystick**:3D position can be obtained and difference from initial position measured.
- **Rotation**
  - **Mouse + Keyboard:** 2 keys for each rotation axis have been defined. We decided to have only 90 degrees for rotation, since it is enough for components.
  - **Flystick:** 3D orientation can be obtained and difference from initial position measured.

Potential Users:

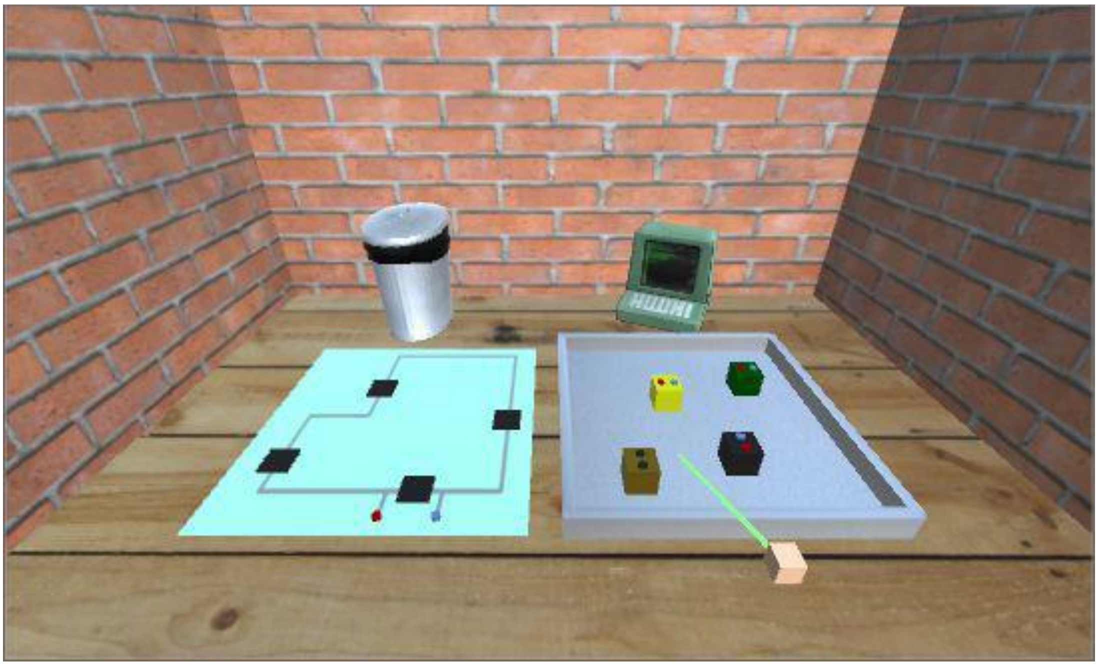
Regarding to the target population, the users for whom the system is designed to, are young students older than **ten years**,with **basic knowledge and manipulation of computers (keyboard and mouse) and technologies**.

Different Tasks in the System:

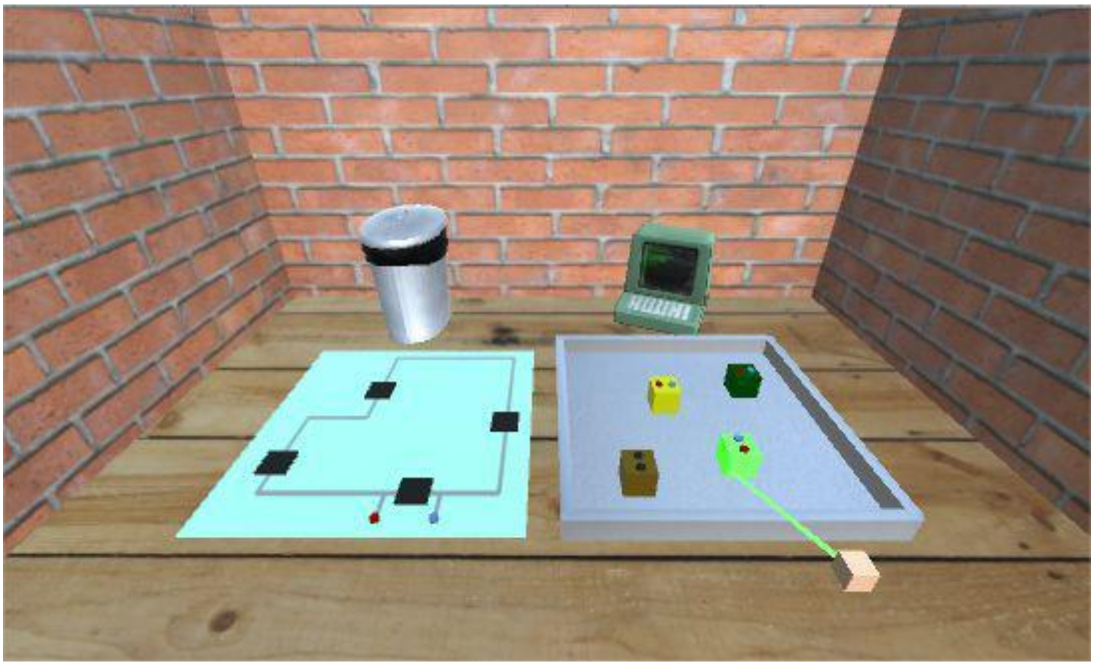


Schemas of the System:

- **Beginning:**

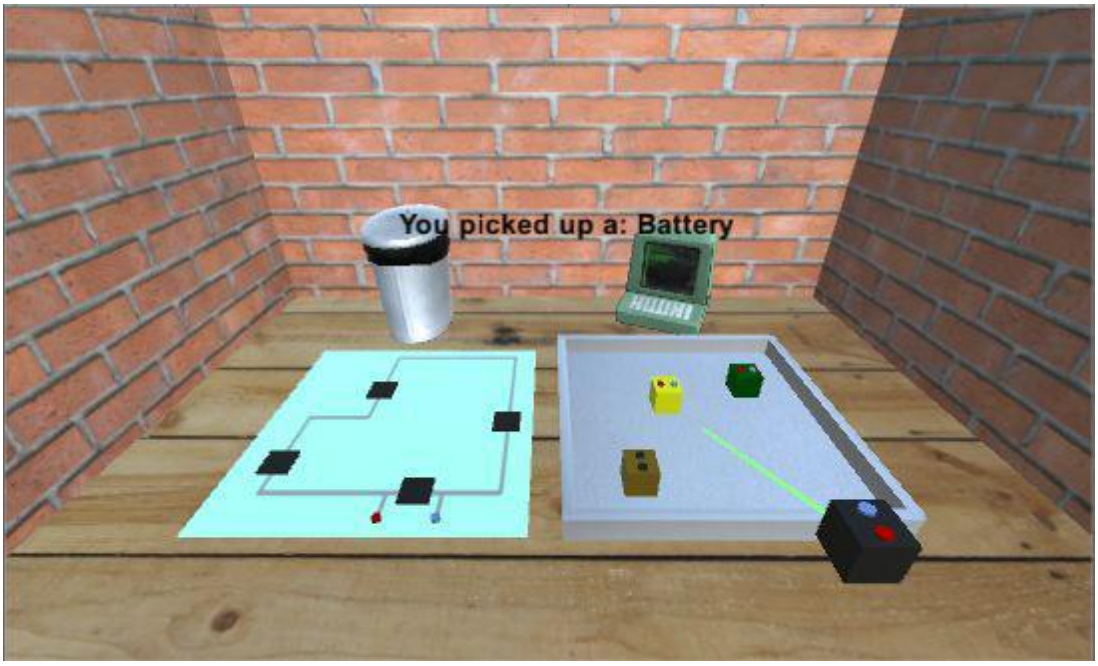


- **Select an object**



Move the pointer (hand) with mouse

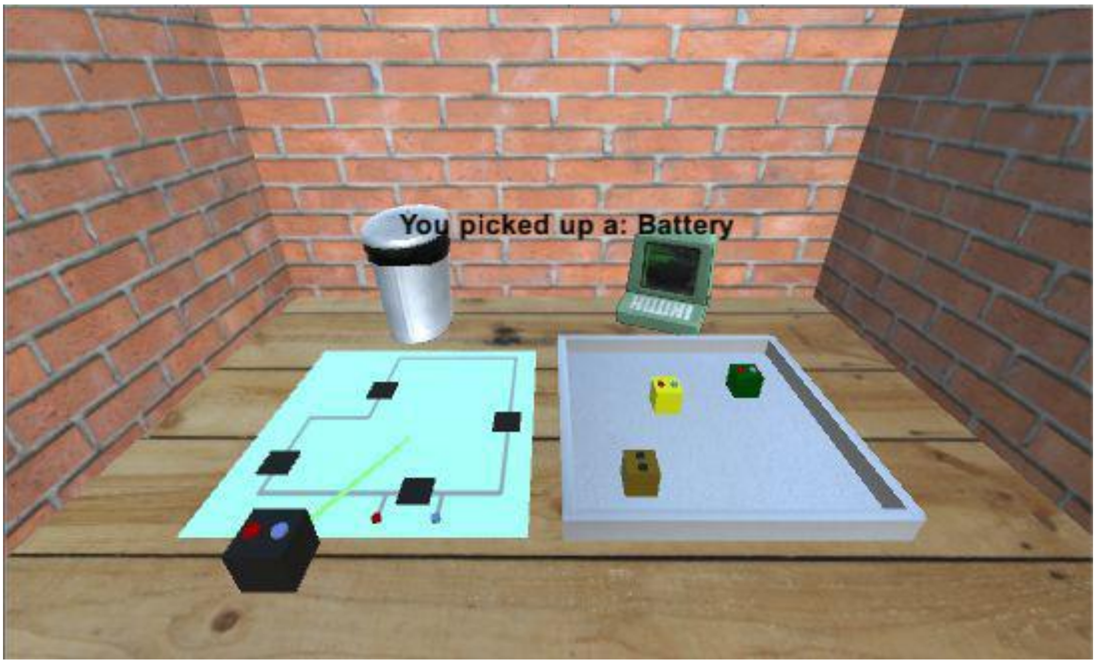
- **Validate selection:**



You picked up a: Battery

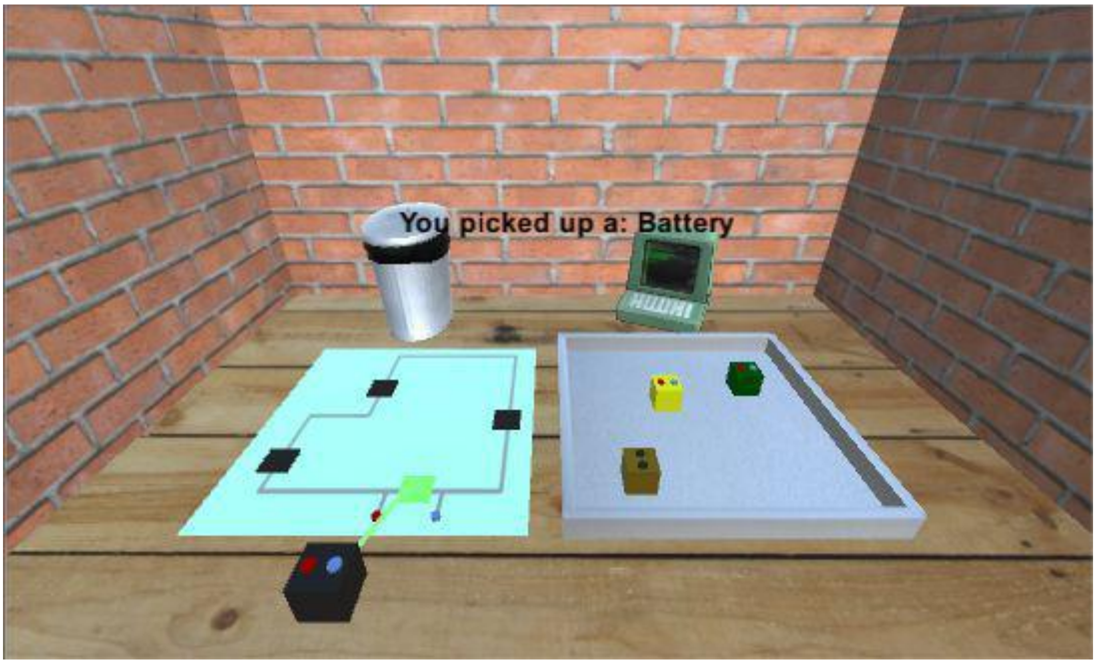
Stay selecting for 0.5s

- **Manipulate object:**

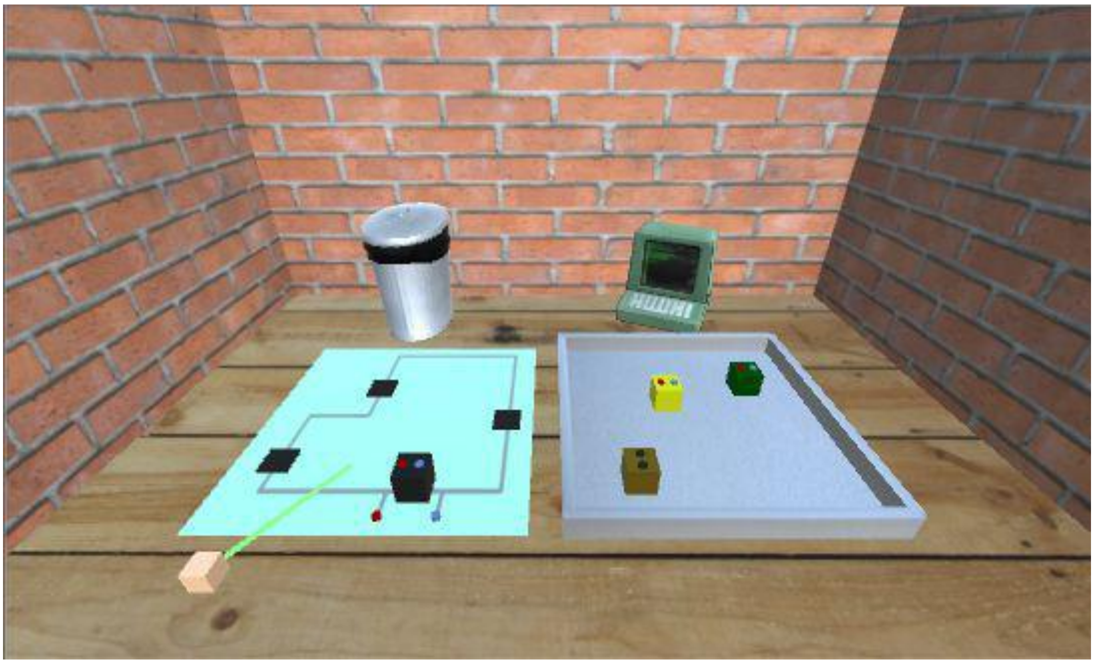


Rotate with arrow keys + Move with mouse

- **Place object:**

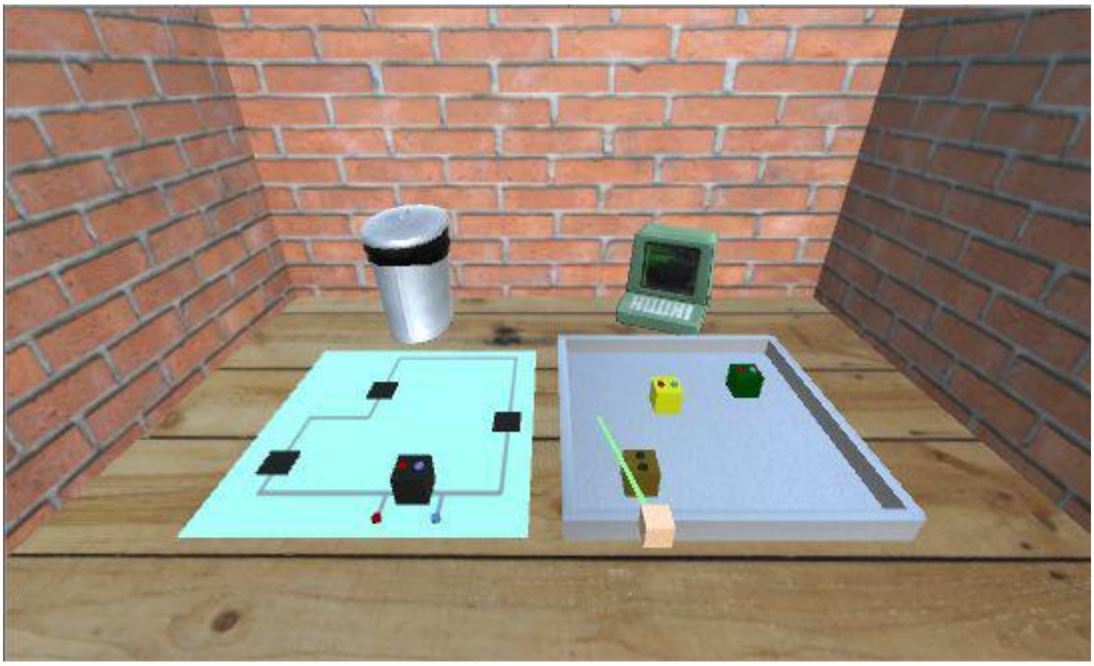


Move with mouse to a target



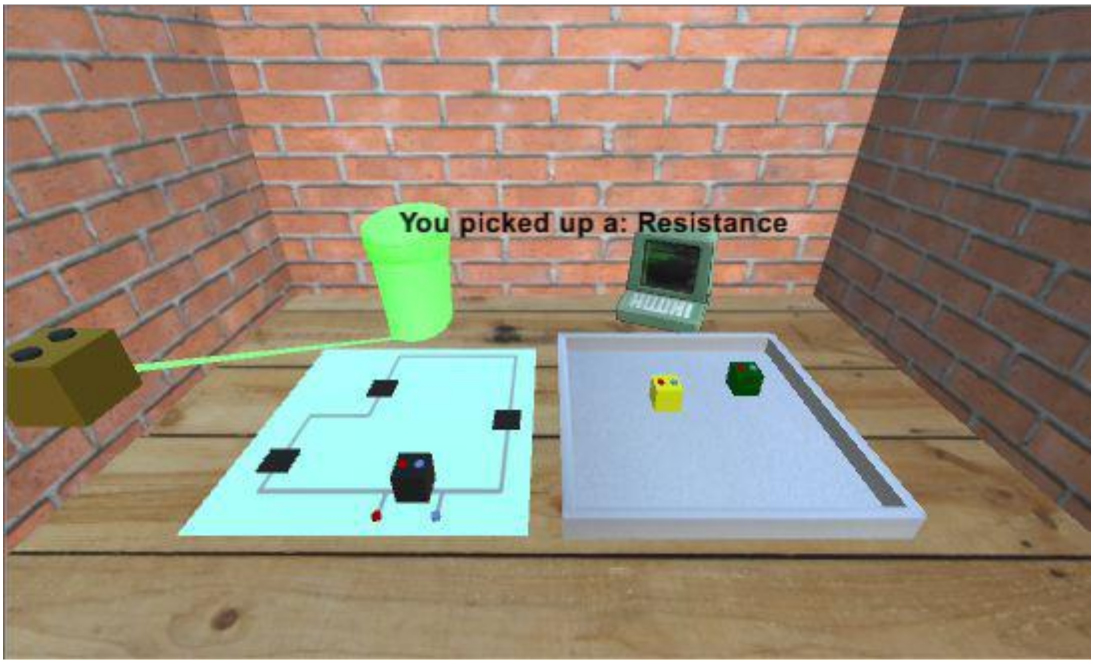
Stay for 0.5s to validate position

- **Return to “beginning”**



We can move the pointer (mouse) to continue selecting and placing the rest of the components

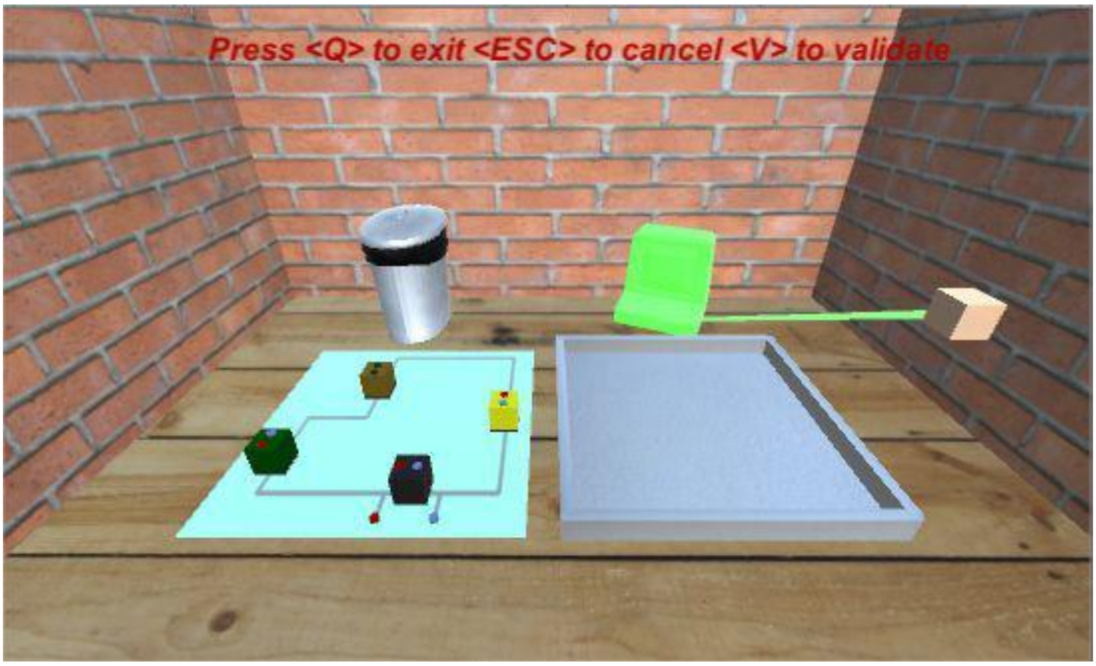
- **Delete a selected object**



You picked up a: Resistance

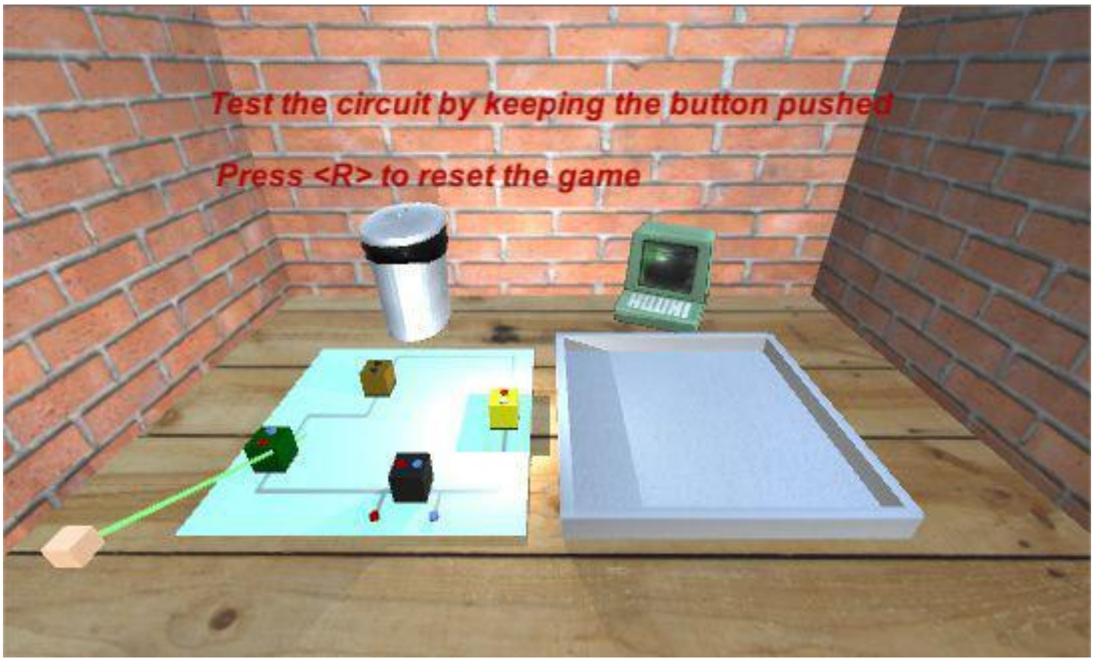
It will move the component back to the components' box

- **Validate the work**



Press <Q> to exit <ESC> to cancel <V> to validate

Key "V" for Validation



Test the circuit by keeping the button pushed

Press <R> to reset the game

**While we press the component "Button" (green component)
the component "LED" shows an animated
light to show everything is ok**

Informal Evaluation:

We have tried to make a system where the "commands" are natural to human, since he must move/rotate the hand in a similar way as the real life. However, it is difficult to achieve a natural interface with Raycasting technique.

Main changes regarding to version 1.1

After a first discussion on the document v1.1, we decided to use the Raycasting technique for both tasks: Selection and Manipulation, since it makes easy for the user to learn how to use the application.

We decided to modelate the components as simple cubes in the first version, but we added text messages to guide the user (tell him which component is each one)

Validations are made in the Testing part, however, more information to the user could be added when he makes mistakes.

Finally, we consider that we must include some counters for errors and some time measurements so as to evaluate the application.