

# Appendix A

## A.1 Animation Scenario: Bar

A bar has been chosen as scenario to illustrate the proposed framework. In this scenario the agents are stereotyped entities with different roles (e.g., waiter, barman, and customer). The objects of the scenario are passive entities subject to the agents' actions, but can influence the course of the agents' activities. The environment is divided into regions such as the "bar area" and the "restaurant area". The initial scenario is depicted in Figure A.1. There are several camera viewing positions in the scene (8 from the sides and one from the top) that can be selected and included in the detailed script which is generated by the *Controller* and subsequently executed by the BAS.

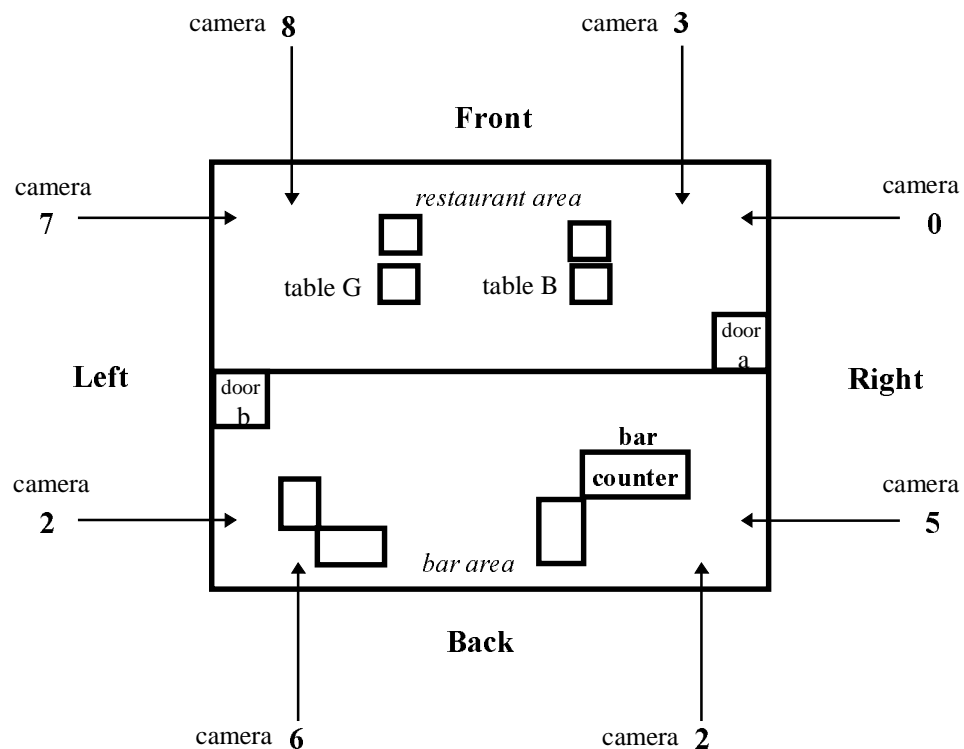


Figure A.1: Layout of the animation scenario with 9 camera viewing positions.

## A.2 An Example of a Script

The specification of a script for the animation is shown in the listing below. The first and the largest part of the script comprises the commands (i.e., *attr*) for setting up the animation scenario. The remainder of the listing comprises the actual motion commands of the script. The initial scenario starts with a barman at the bar counter, a waiter in the restaurant area, and a customer. The customer in particular is instructed to drink at a table which is indicated by the *script* clause. The other characters, the barman and the waiter, perform default activities which obviously are not specified in the script. In the later stages of the animation, at times 48 and 180, two other customers are created and assigned with the explicit actions of “having a drink at the counter”. The customers come from different “doors”, which are drawn as small rectangle areas, and, by default, exit at the “door a”.

```
object( peter, barman ),           % Create instance of agent peter as barman
object( pinky, waiter ),           % Create instance of agent pinky as waiter
object( bar_counter, counter ),    % Create “bar counter” managed the barman
object( exc_counter, shelf ),      % Create “exchange counter” as a service
counter
object( sup_counter, shelf ),      % “supply counter” provides supliment of “new
glasses”
object( wst_counter, shelf ),      % “waste counter” to store “used glasses”.

object( black_table, table ),      object( black_chair, chair ), object( green_table, table ),
object( green_chair, chair ),      object( glass_bb1, glass ), object( glass_bb2, glass ),
object( glass_bx1, glass ),        object( glass_gx1, glass ), object( glass_gx2, glass ),
object( glass_gs1, glass ),        object( glass_gs2, glass ), object( glass_bw1, glass ),
object( glass_bgt1, glass ),       object( glass_bbt1, glass ), object( glass_bbt2, glass ),

region( world, surface ),          % Define region
region( bar_area, surface ),
region( restaurant, surface ),
region( door_a, door ),
region( door_b, door ),

% Set attributes to the animated instances.

attr( world, area, a(-500, 1000, -500, 800) ), % Xmin, Xmax, Zmin, Zmax.
attr( bar_area, area, a(-500, 300, -500, 800) ),
attr( bar_area, place, world ),
attr( bar_area, colour, salmon ),
attr( restaurant, area, a( 300, 1000, -500, 800) ),
attr( restaurant, place, world ),
attr( restaurant, colour, green ),
attr( door_a, area, a( 100,300, -500, -350) ), attr( door_a, direction, -90 ),
attr( door_a, place, bar_area ), attr( door_a, colour, yellow ),
attr( door_b, area, a( 300, 500, 650, 800) ), attr( door_b, direction, 90 ),
attr( door_b, place, restaurant ), attr( door_b, colour, blue ),
```



```

attr( pinky, location, pos( 400, 0, -86 ) ),
attr( pinky, direction, 0.0 ),
attr( pinky, colour, red ),
attr( pinky, behaviour, waiter ),
attr( pinky, shelf, sup_counter ),
attr( pinky, waste, wst_counter ),
attr( pinky, place, restaurant ),

attr( peter, colour, red ),
attr( peter, supplier, mike ),
attr( peter, counter, bar_counter ),
attr( peter, shelf, exc_counter ),
attr( peter, location, pos(-200, 0, 490 ) ),
attr( peter, direction, 0.0 ),
attr( peter, behaviour, barman ),
attr( peter, place, bar_area ),

attr( bar_counter, colour, black ),
attr( bar_counter, location, pos(0,0,500) ),
attr( bar_counter, direction, 0.0 ),
attr( bar_counter, contents, 0 ),
attr( bar_counter, objects_list, [] ),
attr( bar_counter, place, bar_area ),

attr( exc_counter, colour, black ),
attr( exc_counter, location, pos(-150,0,270) ),
attr( exc_counter, direction, 90.0 ),
attr( exc_counter, contents, 0 ),
attr( exc_counter, place, bar_area ),

attr( sup_counter, colour, blue ),
attr( sup_counter, location, pos(-400,0,-180) ),
attr( sup_counter, direction, -20.0 ),
attr( sup_counter, contents, 0 ),
attr( sup_counter, place, bar_area ),

attr( wst_counter, colour, blue ),
attr( wst_counter, location, pos(-270,0,-380) ),
attr( wst_counter, direction, -45.0 ),
attr( wst_counter, contents, 0 ),
attr( wst_counter, place, bar_area ),

attr( glass_bb1, colour, black ),
attr( glass_bb1, location, pos(-5,300,420) ),
attr( glass_bb1, place, bar_counter ),
attr( glass_bb1, condition, usedd ),
attr( glass_bb1, volume, 1 ),

attr( glass_bb2, colour, black ),
attr( glass_bb2, location, pos(0,300,550) ),
attr( glass_bb2, place, bar_counter ),
attr( glass_bb2, condition, usedd ),

attr( glass_bx1, colour, black ),
attr( glass_bx1, location, pos(-200,260,250) ),
attr( glass_bx1, place, exc_counter ),
attr( glass_bx1, condition, usedd ),

attr( glass_gx1, colour, green ),
attr( glass_gx1, location, pos(-170,260,270) ),

```

```

attr( glass_gx1, place, exc_counter ),

attr( glass_gx2, colour, green ),
attr( glass_gx2, location, pos(-100,260,270) ),
attr( glass_gx2, place, exc_counter ),

attr( glass_gs1, colour, green ),
attr( glass_gs1, location, pos(-380,260,-180) ),
attr( glass_gs1, place, sup_counter ),

attr( glass_gs2, colour, green ),
attr( glass_gs2, location, pos(-385,260,-170) ),
attr( glass_gs2, place, sup_counter ),

attr( glass_bw1, colour, black ),
attr( glass_bw1, location, pos(-300,260,-390) ),
attr( glass_bw1, place, wst_counter ),

attr( glass_bgt1, colour, black ), % black glass on green table.
attr( glass_bgt1, location, pos(540,190,-90) ),
attr( glass_bgt1, place, green_table ),
attr( glass_bgt1, condition, usedd ),

attr( glass_bgt2, colour, black ), % black glass on green table.
attr( glass_bgt2, location, pos(520,190,-110) ),
attr( glass_bgt2, place, green_table ),
attr( glass_bgt2, condition, usedd ),

attr( glass_bbt1, colour, black ),
attr( glass_bbt1, location, pos(530,190,300) ),
attr( glass_bbt1, place, black_table ),
attr( glass_bbt1, condition, usedd ),

attr( glass_bbt2, colour, black ),
attr( glass_bbt2, location, pos(510,190,320) ),
attr( glass_bbt2, place, black_table ),
attr( glass_bbt2, condition, usedd ),

attr( black_chair, colour, black ),
attr( black_chair, location, pos(700,0,300) ),
attr( black_chair, direction, 180.0 ),
attr( black_chair, place, restaurant ),

attr( black_table, colour, black ),
attr( black_table, location, pos(550,0,300) ),
attr( black_table, direction, 180.0 ),
attr( black_table, place, restaurant ),
attr( black_table, chair, black_chair ),
attr( black_table, current_user, senna ),

attr( green_chair, colour, green ),
attr( green_chair, location, pos(700,0,-180) ),
attr( green_chair, direction, -160.0 ),
attr( green_chair, place, restaurant ),

attr( green_table, colour, green ),
attr( green_table, location, pos(550,0,-120) ),
attr( green_table, direction, -160.0 ),
attr( green_table, place, restaurant ),
attr( green_table, chair, green_chair ),

```

```

.....
.....
% The script begins here.

script( 1, time(1),    create(john, customer, [attr(colour,blue), attr(place,door_a)]) ),
script( 1, time(1),    action(john, drink_at_table_i, [black_table], script_source) ),
script( 2, time(48),   create(mary, customer, [attr(colour,blue), attr(place,door_b)]) ),
script( 2, time(48),   action(mary, have_drink_i, [], script_source) ),
script( 3, time(180),  create(lucy, customer, [attr(colour,black), attr(place,door_a)]) ),
script( 3, time(180),  action(lucy, have_drink_i, [], script_source) )

```

**Figure A.2: The animator's script.**

### A.3 Goal-directed Actions Derived from the Script

The script guides the main developments of the animation. As the animation develops, new goal-directed actions are automatically started by the animation system. These actions take part as background actions, in the form of default actions, reactions to the environment, and interactions between agents. The listing given in Figure A.3 is a list of goal-directed actions produced by the system in response to the developments of the script. These actions will be further expanded into detailed actions before being sent to the BAS, as shown in section A.5. There are occasions that new actions are created and scheduled and are only started when the resources become available.

The instruction *do\_leisure\_i* occurs frequently. It comprises, or develops into several different activities such as: looking around at people, having a drink at a table, stretching the arms, or leaving the area. It is actually a default activity that in most cases is used to keep an agent alive while he is waiting the resumption of an action or has no prescribed activity.

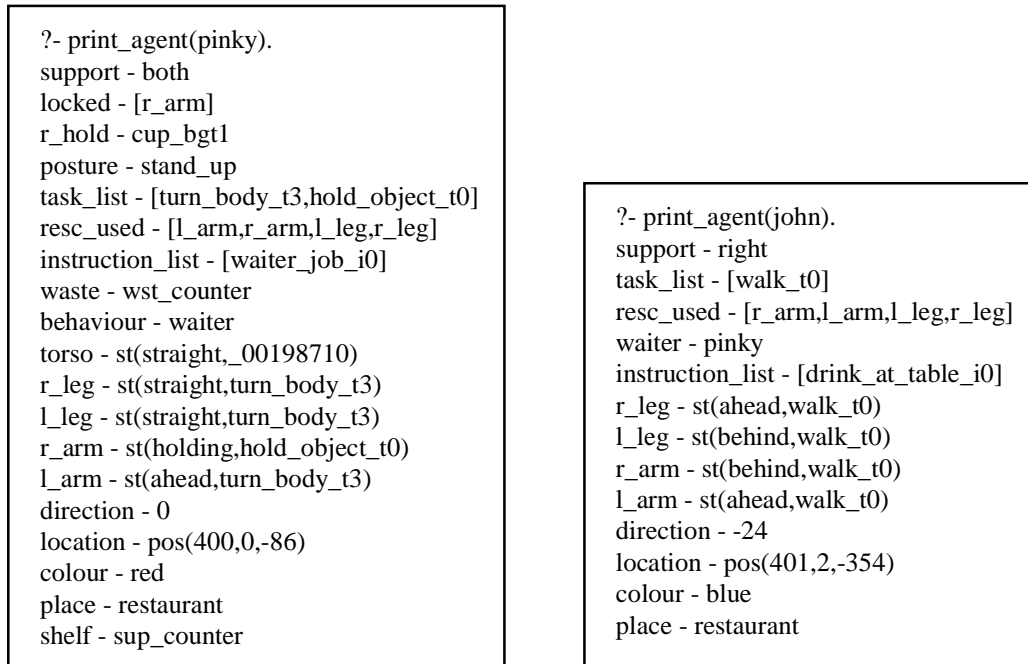
The instruction *give\_attention\_i* is an action instigated for all agents in an area when an agent enters that area.

|     |       |                   |     |       |                   |
|-----|-------|-------------------|-----|-------|-------------------|
| 1   | john  | drink_at_table_i0 | 190 | mary  | look_at_i10       |
| 1   | peter | barman_job_i0     | 190 | peter | give_attention_i4 |
| 1   | pinky | waiter_job_i0     | 191 | john  | look_at_i9        |
| 19  | pinky | give_attention_i0 | 197 | john  | do_leisure_i17    |
| 20  | pinky | give_attention_i0 | 202 | peter | give_attention_i4 |
| 44  | peter | give_attention_i1 | 203 | john  | do_leisure_i18    |
| 48  | mary  | have_drink_i0     | 206 | john  | do_leisure_i19    |
| 53  | peter | give_attention_i1 | 218 | pinky | signal_done_i1    |
| 67  | pinky | give_attention_i2 | 218 | john  | thank_i1          |
| 67  | peter | give_attention_i3 | 234 | peter | barman_job_i2     |
| 71  | pinky | give_attention_i2 | 247 | peter | serve_drink_i2    |
| 72  | peter | give_attention_i3 | 247 | lucy  | do_leisure_i16    |
| 72  | peter | barman_job_i1     | 248 | peter | serve_drink_i2    |
| 82  | peter | serve_drink_i0    | 253 | lucy  | do_leisure_i20    |
| 82  | mary  | do_leisure_i1     | 256 | lucy  | do_leisure_i21    |
| 83  | peter | serve_drink_i0    | 270 | pinky | give_attention_i5 |
| 88  | mary  | do_leisure_i2     | 270 | john  | look_at_i16       |
| 91  | mary  | do_leisure_i3     | 271 | pinky | give_attention_i5 |
| 92  | pinky | serve_drink_i1    | 280 | lucy  | do_leisure_i22    |
| 92  | john  | do_leisure_i0     | 286 | lucy  | do_leisure_i23    |
| 94  | pinky | serve_drink_i1    | 289 | lucy  | do_leisure_i24    |
| 98  | john  | do_leisure_i4     | 313 | john  | look_at_i16       |
| 101 | john  | do_leisure_i5     | 313 | lucy  | do_leisure_i25    |
| 115 | mary  | do_leisure_i6     | 319 | john  | do_leisure_i26    |
| 121 | mary  | do_leisure_i7     | 322 | john  | do_leisure_i27    |
| 124 | mary  | do_leisure_i8     | 328 | lucy  | do_leisure_i28    |
| 125 | john  | do_leisure_i9     | 334 | lucy  | do_leisure_i29    |
| 131 | john  | do_leisure_i10    | 337 | lucy  | do_leisure_i30    |
| 134 | john  | do_leisure_i11    | 339 | peter | signal_done_i2    |
| 148 | mary  | do_leisure_i12    | 339 | lucy  | thank_i2          |
| 151 | peter | signal_done_i0    | 400 | lucy  | look_at_i23       |
| 151 | mary  | thank_i0          | 400 | peter | give_attention_i6 |
| 158 | john  | do_leisure_i13    | 412 | peter | give_attention_i6 |
| 164 | john  | do_leisure_i14    | 440 | peter | barman_job_i3     |
| 167 | john  | do_leisure_i15    | 531 | pinky | give_attention_i7 |
| 180 | lucy  | have_drink_i1     | 552 | peter | barman_job_i4     |
| 185 | john  | look_at_i9        |     |       |                   |

**Figure A.3: Goal-directed actions generated by the animation system in response to the animator’s script.**

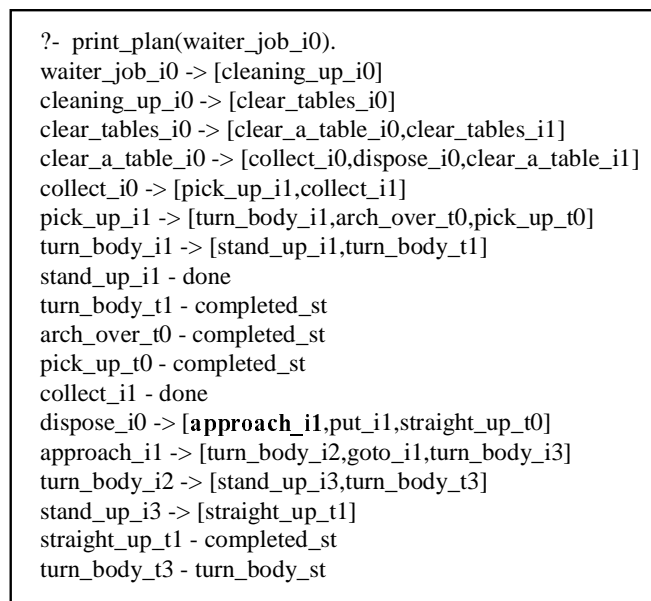
#### **A.4 Examining Instances During Runtime**

It is interesting to examine the contents of the data instances of an ongoing animation. For example, the animation is paused at the frame 19 of the animation sequence when the waitress (*Pinky*), is engaged in the action *waiter\_job\_i0* and is about to start the task *turn\_body\_t3* as part of the sub-action *approach\_i1*. Also at that moment *John*, the first customer, is engaged in the action *drink\_at\_table\_i0* and is currently performing the task *walk\_i0*. Figure A.4 displays the information about both agents at that stage.



**Figure A.4: Instances of the active agents frames.**

Figure A.5 presents the partial planning of the goal actions, *waiter\_job\_i0* and *drink\_at\_table\_i0*, which are performed by the agents *Pinky* and *John* respectively. The arrows indicate the expansion of the instructions into plans. In the specific case of instructions yielding to null plans, these are concluded without further expansion and this is indicated by the *done* literals. The task actions that have been concluded are indicated by *completed\_st* literals.



**Figure A.4: The actions hierarchies. (continued)**



```

?- print_plan(drink_at_table_i0).
drink_at_table_i0 -> [sit_i0,hold_place_i0,look_at_i0,signal_arm_t0,order_drink_m0,drink_i0]
sit_i0 -> [goto_i0,turn_body_t0,approach_t0,sit_t0]
goto_i0 -> [walk_i0]
walk_i0 -> [walk_t0]
walk_t0 - walk_st

```

**Figure A.5: The actions hierarchies.**

Figure A.6 shows the action performed by *Pinky*: *approach\_i1*, and the task performed by *John*: *walk\_t0*. It can be observed that the default parameters are not shown. Figure A.7 presents the state of processes performed by both characters, while Figure A.8 presents information about the objects involved in these actions.

```

?- print_action(approach_i1).
target - wst_counter
obj_ref - false
object - glass_bgt1
closeness - too_far
goal_location - pos(-143,260,-323)
child - turn_body_i2
orig_plan - [turn_body_i2,goto_i1,turn_body_i3]
state - next_st
plan_seq - [goto_i1,turn_body_i3]
param - [place,object,false]
parent - dispose_i0
plan_name - approach_3
root - root_2
direction - 156.42045911921
place - wst_counter

```

```

?- print_action(walk_t0).
sub_goal - pos(593,90,-269)
subgoalist - []
focus - on
start_time - 1
used_resc - [r_arm,l_arm,l_leg,r_leg]
goal_location - pos(593,90,-269)
state - walk_st
param - []
parent - walk_i0
root - root_0

```

**Figure A.6: Two sub-actions internal to the correspondent goal-directed actions.**

```

?- print_process(root_2).
current_instance - turn_body_t3
top_instance - waiter_job_i0
category - primary
rating - 3
source - idle_source
agent - pinky

```

```

?- print_process(root_0).
current_instance - walk_t0
top_instance - drink_at_table_i0
category - primary
rating - 11
source - script_source
agent - john

```

**Figure A.7: Two processes correspondent to the goal-directed actions.**

|   |  |   |
|---|--|---|
| <pre>?- print_agent(glass_bgt1). holder - pinky state - held condition - usedd location - pos(540,190,-90) colour - black place - pinky</pre> | <pre>?- print_agent(wst_counter). objects_list - [glass_bw1] contents - 1 direction - -45 location - pos(-270,0,-380) colour - blue place - bar_area</pre> | <pre>?- print_agent(black_table). current_user - [] objects_list -     [glass_bbt2,glass_bbt1] . contents - 2 direction - 180 location - pos(550,0,300) colour - black place - restaurant chair - black_chair</pre> |
|---|--|---|

**Figure A.8: Printout of instances of some of the animated objects.**

## A.5 The Detailed Script

The detailed script is a list of simple commands generated by the *Controller* which is sent to the BAS. It is the expanded form of the goal-directed actions shown in Figure A.3. These commands are recorded in a file by the BAS; a partial listing of the file is given below. Each command line listed below has three parts: the start time, the name of the procedure to be invoked in the BAS, and the data supplied to the procedure. Some commands start with an X which indicates that they have been commented out and have been used by the Controller to enquire from BAS about the animated entities. They are included in the file for documentation purposes and can be used during the debugging of the BAS procedures.

% The animation begins with the generation of the animated objects.

```
0 CreateRegion salmon -500 300 -500 800
0 CreateRegion green 300 1000 -500 800
0 CreateRegion yellow 100 300 -500 -350
0 CreateRegion blue 300 500 650 800
0 CreateActor peter person red -200 0 490 0 bar_area
0 CreateActor pinky person red 400 0 -86 0 restaurant
0 CreateObject bar_counter counter black 0 0 500 0 bar_area
0 CreateObject exc_counter shelf black -150 0 270 90 bar_area
0 CreateObject sup_counter shelf blue -400 0 -180 -20 bar_area
0 CreateObject wst_counter shelf blue -270 0 -380 -45 bar_area
0 CreateObject black_table table black 550 0 300 180 restaurant
0 CreateObject black_chair chair black 700 0 300 180 restaurant
0 CreateObject green_table table green 550 0 -120 -160 restaurant
0 CreateObject green_chair chair green 700 0 -180 -160 restaurant
0 CreateObject glass_bb1 glass black -5 300 420 0 bar_counter
0 CreateObject glass_bb2 glass black 0 300 550 0 bar_counter
0 CreateObject glass_bx1 glass black -200 260 250 0 exc_counter
0 CreateObject glass_gx1 glass green -170 260 270 0 exc_counter
0 CreateObject glass_gx2 glass green -100 260 270 0 exc_counter
0 CreateObject glass_gs1 glass green -380 260 -180 0 sup_counter
```

```

0 CreateObject glass_gs2 glass green -385 260 -170 0 sup_counter
0 CreateObject glass_bw1 glass black -300 260 -390 0 wst_counter
0 CreateObject glass_bgt1 glass black 540 190 -90 0 green_table
0 CreateObject glass_bbt1 glass black 530 190 300 0 black_table
0 CreateObject glass_bbt2 glass black 510 190 320 0 black_table
0 CreateActor john person blue 200 0 -425 -90 door_a

% The Controller enquire BAS using the following commands.
X0 GetObjectPoint john green_chair right
X0 GetApproachPoint peter bar_counter glass_bb2
X0 CanReach pinky right glass_bgt1
X0 TestPathObstruction john 200 -425 593 -269
X0 TestPathObstruction john 200 -425 593 -269
X0 TestProximity pinky
X0 TestProximity peter

% Set camera view from the above.
% Start of the animation.
% Customer 1 (john) enters from door B.
% John heads to the table.
0 SetCamera 8
0 SendIA pinky 0 arch_over_r 1
0 SendIA peter 1 walkf_app_lleg 0 0.428571428571429 0.75
0 SendIA john 2 walkf_co_lleg 65 0.3
0 SendIA john -1 l_arm_bacw
0 SendIA john -1 r_arm_forw
X5 CanReach pinky right glass_bgt1

% Waitress (pinky) picks up glass_bgt1 with the right arm after having done
'arch_over_r'.
5 SendIA pinky 3 r_arm_reach_goal 540 190 -90
8 ActorHoldObject pinky right glass_bgt1

% How far pinky is from wst_counter?
X8 GetObjectPoint pinky wst_counter front
8 SendIA pinky 4 r_arm_hold_forw
8 SendIA pinky 5 straight_up

% Is there anybody obstructing john's path to the table?
X9 TestPathObstruction john 237 -427 593 -269
9 SendIA peter 6 walkf_app_rleg 0 6.19047619047619E-002 0.05
9 SendIA john 7 walkf_co_rleg 1.0673836919085 1
9 SendIA john -1 l_arm_forw
9 SendIA john -1 r_arm_bacw

% Test the proximity of any object in the vicinity around pinky.
X13 TestProximity pinky

% pinky walk a step backwards.
13 SendIA pinky 8 walksr_back 80 -0.3 0
13 SendIA pinky -1 bl_arm_bacw
17 SendIA pinky 9 r_arm_hold_forw

% For every new step test for obstruction in the john's path.
X18 TestPathObstruction john 401 -354 593 -269
X18 CanReach peter right glass_bb2
18 SendIA peter 10 r_raise_pick
18 SendIA john 11 walkf_co_lleg 0.120628635655141 0.999875275668633
18 SendIA john -1 l_arm_bacw
18 SendIA john -1 r_arm_forw

% john has entered in the restaurant area which instigates pinky to turn and look at him.

```

```

19  SendIA pinky 12 turn_head_torso_r 11.2970413251273
% Checks if peter can reach glass_bb2.
% He is moving used glasses from bar_counter to exc_counter.
X21  CanReach peter right glass_bb2
21  SendIA peter 13 r_arm_reach_goal 0 300 550
24  ActorHoldObject peter right glass_bb2
X24  GetPlacingPoint peter exc_counter back
X24  GetObjectPoint peter exc_counter back
X24  TestProximity peter
24  SendIA peter 14 r_arm_hold_forw
24  SendIA peter 15 walkf_co_lleg 80 0.3
24  SendIA peter -1 l_arm_bacw
% pinky goes to the wst_counter to dispose the glass held on hand.
X25  TestProximity pinky
25  SendIA pinky 16 walkf_co_lleg 60 0.3
25  SendIA pinky -1 l_arm_bacw
26  SendIA pinky 17 r_arm_hold_forw
X27  TestPathObstruction john 564 -282 593 -269
27  SendIA john 18 walkf_co_rleg -0.145 0.105
27  SendIA john -1 l_arm_forw
27  SendIA john -1 r_arm_bacw
33  SendIA peter 19 r_arm_hold_forw
X34  TestProximity pinky
X34  TestProximity peter
X34  TestPathObstruction pinky 372 -99 -143 -323
X34  TestPathObstruction pinky 372 -99 -143 -323
34  SendIA peter 20 walkf_app_rleg 20.477 0.561 -5.817E-002
34  SendIA pinky 21 walkf_co_rleg 16.4932986081618 1
34  SendIA pinky -1 l_arm_forw
35  SendIA pinky 22 r_arm_hold_forw
X36  TestProximity john
36  SendIA john 23 walksl_back -50 -0.3 0
36  SendIA john -1 bl_arm_forw
36  SendIA john -1 br_arm_bacw
X42  TestProximity john
42  SendIA john 24 walkf_co_rleg -60 0.3
42  SendIA peter 25 r_arm_hold_forw
42  SendIA john -1 l_arm_forw
42  SendIA john -1 r_arm_bacw
X43  TestPathObstruction pinky 201 -175 -143 -323
43  SendIA peter 26 walkf_app_lleg 0.47 1.66E-002 -3.00E-002
43  SendIA pinky 27 walkf_co_lleg 0.721050801997308 1
43  SendIA pinky -1 l_arm_bacw
44  SendIA pinky 28 r_arm_hold_forw
% New character, mary, is created in the area door_b.
47  CreateActor mary person blue 400 0 725 90 door_b
% Find the approaching point at the front_right of the bar_counter.
X47  GetObjectPoint mary bar_counter front_right
X47  TestProximity mary
47  SendIA mary 29 walkf_co_lleg 60 0.3
47  SendIA mary -1 l_arm_bacw
47  SendIA mary -1 r_arm_forw
X48  TestProximity mary
48  SendIA mary 30 walkf_co_rleg 41.622037810351 0.3
48  SendIA mary -1 l_arm_forw
48  SendIA mary -1 r_arm_bacw
X51  TestProximity john

```

```

% peter keeps holding the glass while walking towards exc_counter.
51 SendIA peter 31 r_arm_hold_forw
51 SendIA john 32 walksr_back -26 -0.124123277511553 -1
X52 TestPathObstruction pinky 36 -244 -143 -323

% peter turn to see mary entering the bar_area.
52 SendIA peter 33 turn_head_torso_r -27.3603177356609
52 SendIA pinky 34 walkf_co_rleg -0.813823559600849 0.931704117388282
52 SendIA pinky -1 l_arm_forw
53 SendIA pinky 35 r_arm_hold_forw
X57 TestProximity mary
X57 TestPathObstruction mary 332 670 120 575
57 SendIA john 36 walkf_app_lleg 0 0.299741939379343 -1

% mary walks and swing both arms.
57 SendIA mary 37 walkf_co_lleg 3.86221854061623 1
57 SendIA mary -1 l_arm_bacw
57 SendIA mary -1 r_arm_forw

% peter arch_over the exc_counter to place the glass (glass_bb2) of the right hand.
60 SendIA peter 38 arch_over_r 2
X61 TestPathObstruction pinky -117 -311 -143 -323
61 SendIA pinky 39 walkf_co_lleg -0.775140568831887 9.54521404218423E-002
61 SendIA pinky -1 l_arm_bacw
62 SendIA pinky 40 r_arm_hold_forw
65 SendIA peter 41 r_arm_reach_goal -150 260 270
X66 TestPathObstruction mary 171 597 120 575

% john is approaching the chair with a step rather sideways (-0.292962594392564).
66 SendIA john 42 walksr_back 0 -1.72890144223994E-002 -0.292962594392564
66 SendIA mary 43 walkf_co_rleg 1.66595872329023 0.185142587693318
66 SendIA mary -1 l_arm_forw
66 SendIA mary -1 r_arm_bacw
68 ActorReleaseObject peter glass_bb2
68 SendIA peter 44 r_arm_retract
X70 TestProximity pinky
X70 GetPlacingPoint pinky wst_counter front
X71 GetApproachPoint peter bar_counter glass_bb1
71 SendIA pinky 45 arch_over_r 2

% peter straight himself after having arched over.
71 SendIA peter 46 straight_up

% john sits and put the arms 'on the legs'.
72 SendIA john 47 sit_on_chair 80
72 SendIA john -1 l_put_on_leg
72 SendIA john -1 r_put_on_leg
X75 TestProximity mary
75 SendIA mary 48 turn_head_torso_r -10.5034369824142
X76 TestProximity peter
76 SendIA pinky 49 r_arm_reach_goal -256 260 -394
76 SendIA peter 50 walksl_back -50 -0.3 0
76 SendIA peter -1 bl_arm_forw
76 SendIA peter -1 br_arm_bacw
79 ActorReleaseObject pinky glass_bgt1
79 SendIA pinky 51 r_arm_retract
79 SendIA john 52 turn_head_torso_r -32.7056956048207
81 SendIA mary 53 turn_head_torso_r -100.363702691457
X82 GetApproachPoint peter exc_counter glass_bb2
X82 TestProximity peter
82 SendIA pinky 54 straight_up
82 SendIA peter 55 walksr_back 36.0489110211884 -0.3 0

```

```

82  SendIA peter -1 bl_arm_bacw
82  SendIA peter -1 br_arm_forw
85  SendIA john 56 signal_r_arm

% john has ordered a drink to the pinky who is going to deliver at the black_table.
X87  CanReach pinky right glass_bbt1
X87  GetApproachPoint pinky black_table glass_bbt1
X87  TestProximity pinky
87   SendIA pinky 57 walksr_back 80 -0.3 0
87   SendIA mary 58 turn_back
87   SendIA pinky -1 bl_arm_bacw
87   SendIA pinky -1 br_arm_forw

% mary has ordered a drink from the bar_counter to peter who respond promptly.
% She takes her time by involuntary stretching her arms lazily.
X88  TestProximity peter
X88  GetApproachPoint peter exc_counter glass_gx2
X88  TestProximity peter
88   SendIA peter 59 walkf_app_lleg 3 0.228571428571429 0.825
90   SendIA mary 60 lazy_stretch2

% john takes his time by looking at people around.
91   SendIA john 61 turn_head_torso_r 36.6362973085426
X93  CanReach pinky right glass_gs2
X93  GetApproachPoint pinky sup_counter glass_gs2
X93  TestProximity pinky
93   SendIA pinky 62 walkf_app_lleg -65 0.330850406427138 -1
97   SendIA peter 63 walkf_app_rleg 0 3.80952380952381E-002 0.0625

% john takes more time because the waitress is still doing his order.
97   SendIA john 64 turn_back
100  SendIA john 65 lazy_stretch

% pink walks few steps sideways from wst_counter to the sup_counter where the
supply of drinks are.
102  SendIA pinky 66 walksr_back -11 -0.054272334252425 -1
X106 CanReach peter right glass_gx2
106  SendIA peter 67 r_raise_pick
108  SendIA pinky 68 walkf_app_lleg 0 3.20799028101365E-002 -
0.536134282496295
X109 CanReach peter right glass_gx2
109  SendIA peter 69 r_arm_reach_goal -100 260 270
112  ActorHoldObject peter right glass_gx2
X112 GetObjectPoint peter bar_counter back_right
112  SendIA peter 70 r_arm_hold_forw
112  SendIA peter 71 walksr_back -65 -9.52380952380952E-002 1
114  SendIA mary 72 turn_head_torso_r -38.9946041101963
117  SendIA pinky 73 walksr_back 0 -4.556E-002 -3.627E-002
118  SendIA peter 74 walksr_back -25 -9.52380952380952E-002 0.8375
120  SendIA mary 75 turn_back
121  SendIA peter 76 r_arm_hold_forw
X123 CanReach pinky right glass_gs2
123  SendIA pinky 77 r_raise_pick

% mary and john are still waiting for their respective drinks. They take more time...
123  SendIA mary 78 lazy_stretch2
124  SendIA peter 79 walkf_app_lleg 0 7.14285714285714E-002 -0.0375
124  SendIA john 80 turn_head_torso_r 24.1865243261871
X126 CanReach pinky right glass_gs2
126  SendIA pinky 81 r_arm_reach_goal -385 260 -170
129  ActorHoldObject pinky right glass_gs2

```

```

X129  GetObjectPoint pinky green_table front
X129  TestPathObstruction pinky -290 -135 430 -129
X129  TestPathObstruction pinky -290 -135 430 -129
129   SendIA pinky 82 r_arm_hold_forw
129   SendIA pinky 83 walkf_co_lleg -65 0.3
129   SendIA pinky -1 l_arm_bacw
130   SendIA peter 84 r_arm_hold_forw
130   SendIA john 85 turn_back
X133  GetPlacingPoint peter bar_counter front_right

% john is still waiting calmly...
133   SendIA john 86 lazy_stretch
138   SendIA pinky 87 r_arm_hold_forw
X139  TestPathObstruction pinky -314 -220 430 -129
139   SendIA peter 88 r_arm_reach_goal 0 300 575
139   SendIA pinky 89 walkf_co_rleg -65 0.3
139   SendIA pinky -1 l_arm_forw
142   ActorReleaseObject peter glass_gx2
142   SendIA peter 90 r_arm_retract
145   SendIA peter 91 straight_up
147   SendIA pinky 92 r_arm_hold_forw
147   SendIA mary 93 turn_head_torso_r -100.363702691457
X148  TestPathObstruction pinky -277 -222 430 -129
148   SendIA pinky 94 walkf_co_lleg -37.4937612737174 0.7
148   SendIA pinky -1 l_arm_bacw
X150  CanReach mary right glass_gx2
X150  TestProximity peter
X150  TestProximity mary

% peter (barman) and mary acknowledge to each other when the order is delivered.
150   SendIA peter 95 signal_r_arm
150   SendIA mary 96 signal_r_arm
150   SendIA peter 97 walkf_app_lleg 0 2.38095238095238E-002 -1

% pinky holds the drink in the way to john's table.
156   SendIA pinky 98 r_arm_hold_forw

% mary can finally pick up the drink.
156   SendIA mary 99 arch_over_r 2
X157  TestPathObstruction pinky -131 -208 430 -129
157   SendIA john 100 turn_head_torso_r 36.6362973085426
157   SendIA pinky 101 walkf_co_rleg -1.56825771264266E-002 1
157   SendIA pinky -1 l_arm_forw
159   SendIA peter 102 walksr_back 0 -6.19047619047619E-002 -1
X161  CanReach mary right glass_gx2
161   SendIA mary 103 r_arm_reach_goal 5.152093 301.173526 574.505949
163   SendIA john 104 turn_back
164   ActorHoldObject mary right glass_gx2

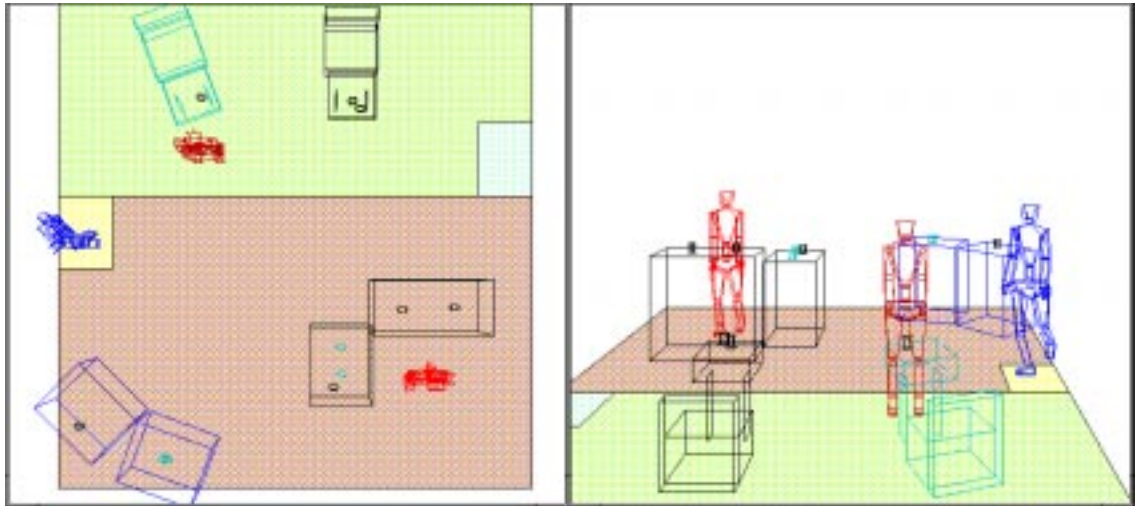
% mary can now enjoy the drink glass_gx2.
164   SendIA mary 105 r_arm_hold_forw
164   SendIA mary 106 straight_up
165   SendIA pinky 107 r_arm_hold_forw
165   SendIA peter 108 walksr_back 0 -4.76190476190476E-003 -0.075
.....

```

**Figure A.9: Detailed script generated by the Controller and set to the BAS.**

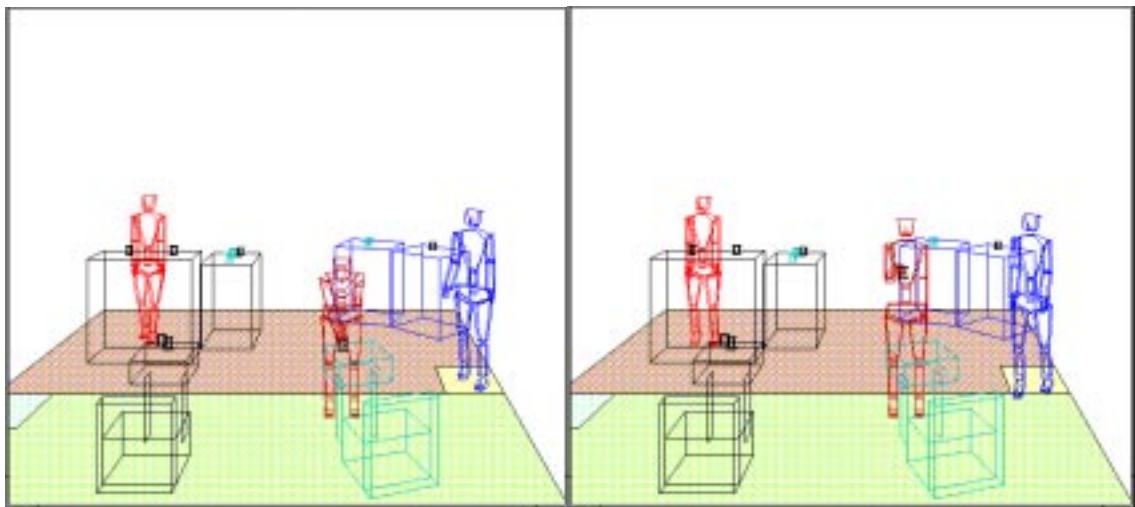
## A.6 The Animated Scene

A number of frames is selected from the animated scene with the purpose of illustration.



Frame 1: View from above.  
Start of the animation.  
Customer 1 in the door B  
Waiter is in front of table G

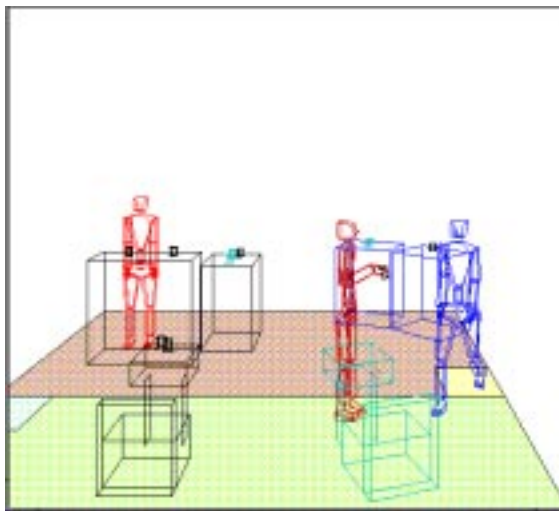
Frame 2: View changed to Camera 8.  
Barman collects "used glasses"  
Waiter collects "used glasses"  
Customer heads to table G.



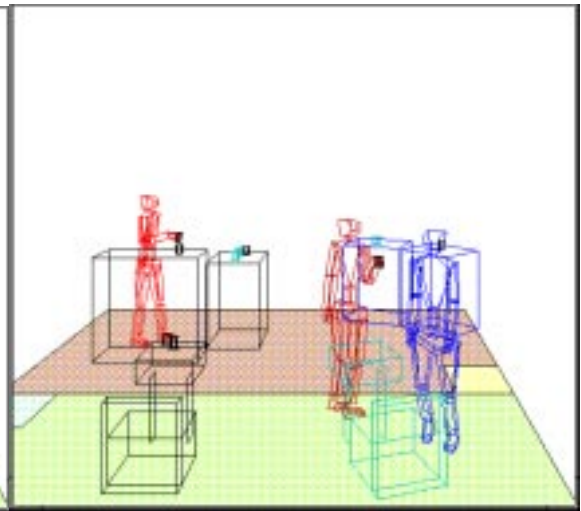
Frame 7: Waiter picking up a glass.

Frame 12: Customer entering  
the restaurant area.

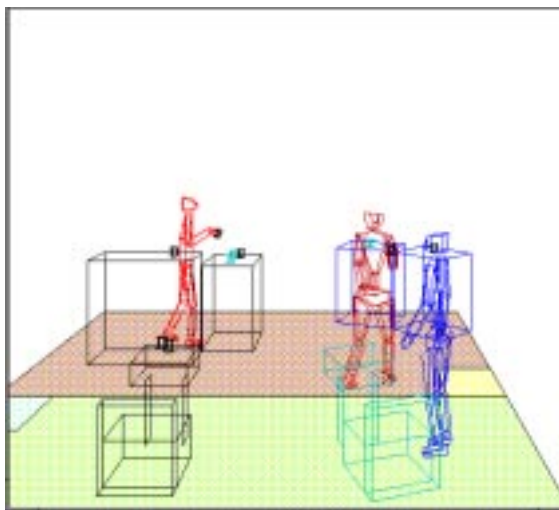




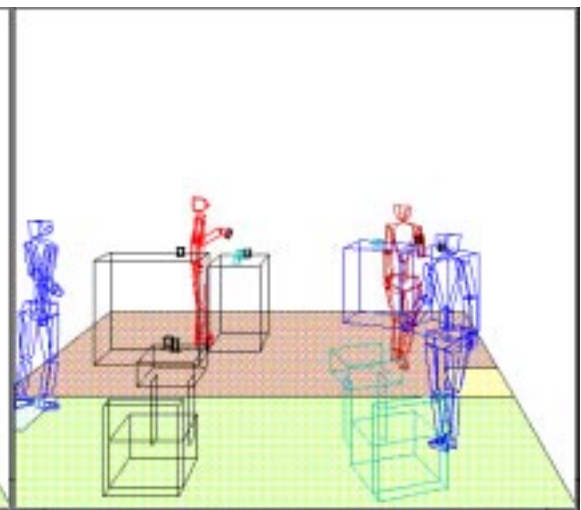
Frame 17: Waiter turning.



Frame 29: Barman carrying a glass.

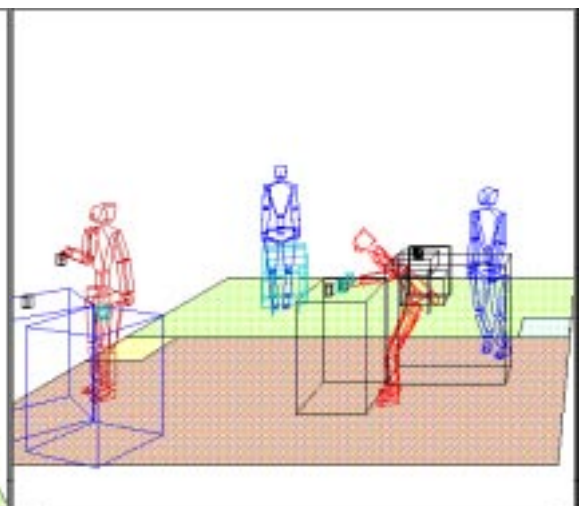
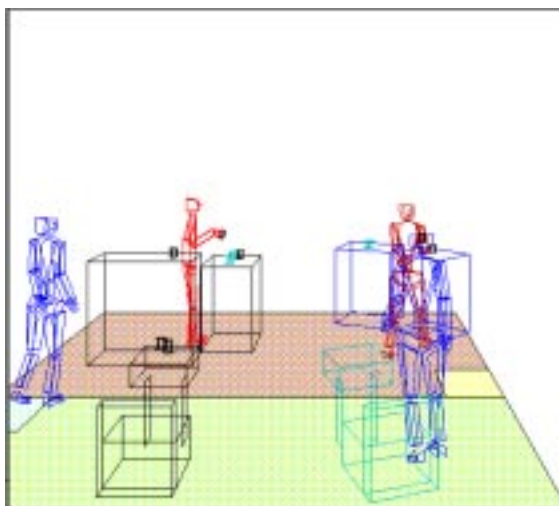


Frame 43: Customer A preparing to sit



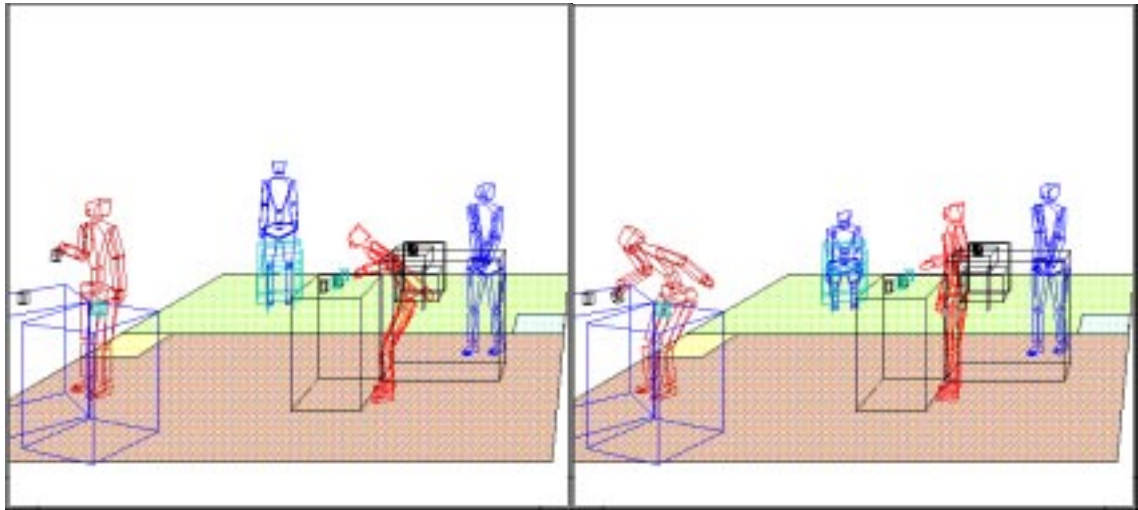
Frame 52: Customer B at the door A

**Figure A.10: Selected frames of the animated scene. (cont.)**



Frame 58: Waiter approaching “waste counter”.

Frame 68: View changed to camera 2.  
Barman puts glass on the “exchange counter”.



Frame 72: Customer A sitting.

Frame 76: Waiter putting a glass.

**Figure A.10: Selected frames of the animated scene.**

|   |            |
|---|------------|
| <b>APPENDIX A</b> .....   | <b>176</b> |
| A.1 ANIMATION SCENARIO: BAR .....   | 176        |
| A.2 AN EXAMPLE OF A SCRIPT .....  | 177        |
| A.3 GOAL-DIRECTED ACTIONS DERIVED FROM THE SCRIPT .....   | 180        |
| A.4 EXAMINING INSTANCES DURING RUNTIME.....   | 181        |
| A.5 THE DETAILED SCRIPT .....   | 184        |
| A.6 THE ANIMATED SCENE.....   | 190        |
| <br>  |            |
| FIGURE A.1: LAYOUT OF THE ANIMATION SCENARIO WITH 9 CAMERA VIEWING POSITIONS. ....                                | 176        |
| FIGURE A.2: THE ANIMATOR’S SCRIPT. ....   | 180        |
| FIGURE A.3: GOAL-DIRECTED ACTIONS GENERATED BY THE ANIMATION SYSTEM IN RESPONSE TO THE<br>ANIMATOR’S SCRIPT. .... | 181        |
| FIGURE A.4: INSTANCES OF THE ACTIVE AGENTS FRAMES. ....   | 182        |
| FIGURE A.5: THE ACTIONS HIERARCHIES. ....   | 183        |
| FIGURE A.6: TWO SUB-ACTIONS INTERNAL TO THE CORRESPONDENT GOAL-DIRECTED ACTIONS. ....                             | 183        |
| FIGURE A.7: TWO PROCESSES CORRESPONDENT TO THE GOAL-DIRECTED ACTIONS.....   | 183        |
| FIGURE A.8: PRINTOUT OF INSTANCES OF SOME OF THE ANIMATED OBJECTS.....  | 184        |
| FIGURE A.9: DETAILED SCRIPT GENERETED BY THE CONTROLLER AND SET TO THE BAS.....                                   | 189        |
| FIGURE A.10: SELECTED FRAMES OF THE ANIMATED SCENE. (CONT.).....  | 191        |