```
#d010>lucas>microcosm:
capacity.equates.incl.pl1
capacity.incl.pl1
class.externals.incl.pl1
defs_action.incl.pl1
defs_class.incl.pl1
defs_helper.incl.pl1
defs_statistics.incl.pl1
defs_struct.incl.pl1
instance_head.incl.pl1
microcosm.incl.pl1
region.structs.incl.pl1
width.incl.pl1
Actions
Classes
Filters
Ghu
Grabthese
Linkable
Misc
Realms
Structs

#d010>lucas>microcosm>Classes:
class_amulet.pl1
class_aquarium.pl1
class_atm.pl1
class_avatar.pl1
class_bag.pl1
class_ball.pl1
class_bed.pl1
class_book.pl1
class_boomerang.pl1
class_bottle.pl1
class_box.pl1
class_bridge.pl1
class_building.pl1
class_bureaucrat.pl1
class_bush.pl1
class_chair.pl1
class_changomatic.pl1
class_chest.pl1
class_club.pl1
class_coke_machine.pl1
class_compass.pl1
class_couch.pl1
class_countertop.pl1
class_crystal_ball.pl1
class_die.pl1
class_display_case.pl1
class_door.pl1
class_dropbox.pl1
class_drugs.pl1
class_elevator.pl1
class_escape_dev.pl1
class_fake_gun.pl1
class_fence.pl1
class_flag.pl1
class_flashlight.pl1
class_flat.pl1
```

```
class_floor_lamp.pl1
class_fortune_machine.pl1
class_fountain.pl1
class_frisbee.pl1
class_game_piece.pl1
class_garbage_can.pl1
class_gate.pl1
class_gemstone.pl1
class_ghost.pl1
class_glue.pl1
class_grenade.pl1
class_ground.pl1
class_gun.pl1
class_hand_of_god.pl1
class_hat.pl1
class_head.pl1
class_hole.pl1
class_hot_tub.pl1
class_house_cat.pl1
class_instant_object.pl1
class_jacket.pl1
class_jukebox.pl1
class_key.pl1
class_knick_knack.pl1
class_knife.pl1
class_magic_lamp.pl1
class_magic_staff.pl1
class_magic_wand.pl1
class_mailbox.pl1
class_matchbook.pl1
class_movie_camera.pl1
class_pants.pl1
class_paper.pl1
class_pawn_machine.pl1
class_picture.pl1
class_plant.pl1
class_plaque.pl1
class_pond.pl1
class_region.pl1
class_ring.pl1
class_river.pl1
class_rock.pl1
class_roof.pl1
class_safe.pl1
class_security_dev.pl1
class_sensor.pl1
class_sex_changer.pl1
class_shirt.pl1
class_short_sign.pl1
class_shovel.pl1
class_sign.pl1
class_skateboard.pl1
class_sky.pl1
class_spray_can.pl1
class_stereo.pl1
class_street.pl1
class_streetlamp.pl1
class_stun_gun.pl1
class_super_trapezoid.pl1
class_switch.pl1
```

```
class_table.pl1
class_tape.pl1
class_teleport.pl1
class_test.pl1
class_ticket.pl1
class_tokens.pl1
class_trapezoid.pl1
class_tree.pl1
class_vendo_front.pl1
class_vendo_inside.pl1
class_wall.pl1
class_window.pl1
class_windup_toy.pl1

#d010>lucas>microcosm>Misc:
bits.pl1
capacity_monitor.pl1
curses.pl1
helpers.pl1
magic.pl1
messages.pl1
width.pl1

#d010>lucas>microcosm>Actions:
actions.pl1
actions_clothing.pl1
actions_container.pl1
actions_door.pl1
actions_gpt.pl1
actions_help.pl1
actions_mail.pl1
actions_music.pl1
actions_oracle.pl1
actions_switch.pl1
actions_weapon.pl1
actions_clothing.incl.pl1
actions_container.incl.pl1
actions_door.incl.pl1
actions_gpt.incl.pl1
actions_help.incl.pl1
actions_magic.incl.pl1
actions_mail.incl.pl1
actions_music.incl.pl1
actions_oracle.incl.pl1
actions_switch.incl.pl1
actions_weapon.incl.pl1

#d010>lucas>microcosm>Structs:
struct_amulet.incl.pl1
struct_aquarium.incl.pl1
struct_atm.incl.pl1
struct_avatar.incl.pl1
struct_bag.incl.pl1
struct_ball.incl.pl1
struct_bed.incl.pl1
struct_book.incl.pl1
struct_boomerang.incl.pl1
struct_bottle.incl.pl1
struct_box.incl.pl1
struct_bridge.incl.pl1
```

```
struct_building.incl.pl1
struct_bureaucrat.incl.pl1
struct_bush.incl.pl1
struct_chair.incl.pl1
struct_changomatic.incl.pl1
struct_chest.incl.pl1
struct_class.incl.pl1
struct_club.incl.pl1
struct_coke_machine.incl.pl1
struct_compass.incl.pl1
struct_couch.incl.pl1
struct_countertop.incl.pl1
struct_crystal_ball.incl.pl1
struct_die.incl.pl1
struct_display_case.incl.pl1
struct_door.incl.pl1
struct_dropbox.incl.pl1
struct_drugs.incl.pl1
struct_elevator.incl.pl1
struct_escape_dev.incl.pl1
struct_fake_gun.incl.pl1
struct_fence.incl.pl1
struct_flag.incl.pl1
struct_flashlight.incl.pl1
struct_flat.incl.pl1
struct_floor_lamp.incl.pl1
struct_fortune_machine.incl.pl1
struct_fountain.incl.pl1
struct_frisbee.incl.pl1
struct_game_piece.incl.pl1
struct_garbage_can.incl.pl1
struct_gemstone.incl.pl1
struct_gen_container.incl.pl1
struct_gen_door.incl.pl1
struct_gen_magic.incl.pl1
struct_gen_object.incl.pl1
struct_gen_player.incl.pl1
struct_gen_switch.incl.pl1
struct_ghost.incl.pl1
struct_glue.incl.pl1
struct_grenade.incl.pl1
struct_ground.incl.pl1
struct_gun.incl.pl1
struct_hand_of_god.incl.pl1
struct_hat.incl.pl1
struct_head.incl.pl1
struct_hole.incl.pl1
struct_hot_tub.incl.pl1
struct_house_cat.incl.pl1
struct_instant_object.incl.pl1
struct_jukebox.incl.pl1
struct_jukebox_catalog.incl.pl1
struct_key.incl.pl1
struct_knick_knack.incl.pl1
struct_knife.incl.pl1
struct_magic_lamp.incl.pl1
struct_magic_staff.incl.pl1
struct_magic_wand.incl.pl1
struct_mailbox.incl.pl1
struct_matchbook.incl.pl1
```

```
struct_movie_camera.incl.pl1
struct_paper.incl.pl1
struct_pawn_machine.incl.pl1
struct_picture.incl.pl1
struct_plant.incl.pl1
struct_plaque.incl.pl1
struct_pond.incl.pl1
struct_ring.incl.pl1
struct_river.incl.pl1
struct_rock.incl.pl1
struct_roof.incl.pl1
struct_safe.incl.pl1
struct_security_dev.incl.pl1
struct_sensor.incl.pl1
struct_sex_changer.incl.pl1
struct_short_sign.incl.pl1
struct_shovel.incl.pl1
struct_sign.incl.pl1
struct_sky.incl.pl1
struct_spray_can.incl.pl1
struct_stereo.incl.pl1
struct_street.incl.pl1
struct_streetlamp.incl.pl1
struct_stun_gun.incl.pl1
struct_super_trapezoid.incl.pl1
struct_switch.incl.pl1
struct_table.incl.pl1
struct_tape.incl.pl1
struct_teleport.incl.pl1
struct_test.incl.pl1
struct_ticket.incl.pl1
struct_tokens.incl.pl1
struct_trapezoid.incl.pl1
struct_tree.incl.pl1
struct_user.incl.pl1
struct_vendo_front.incl.pl1
struct_vendo_inside.incl.pl1
struct_wall.incl.pl1
struct_window.incl.pl1
struct_windup_toy.incl.pl1

%cvideo#d010>lucas>microcosm>Misc>bits.pl1  88-02-29 19:34:52 EST
clear_bit: procedure(num, the_bit);
set_bit: procedure(num, the_bit);
test_bit: procedure(num, the_bit) returns(bit(1) aligned);
and_bit: procedure(num1, num2) returns(binary(15));
or_bit: procedure(num1, num2) returns(binary(15));

%cvideo#d010>lucas>microcosm>Misc>capacity_monitor.pl1  88-02-29 19:34:53 EST
note_object_creation: procedure(class_number, style);
note_instance_creation: procedure(class_number, style);
note_resource_creation: procedure(class_number, style);
note_instance_creation_internal: procedure(class_number);
note_resource_creation_internal: procedure(class_number, style);
note_resource_usage: procedure(resource);
note_object_deletion: procedure(class_number, style);
note_instance_deletion: procedure(class_number, style);
note_resource_deletion: procedure(class_number, style);
note_instance_deletion_internal: procedure(class_number);
note_resource_deletion_internal: procedure(class_number, style);
```

```
note_resource_removal: procedure(resource);
mem_checks_ok: procedure (the_class) returns (bit(1) aligned);
reconstruct_memory_usage: procedure;

%cvideo#d010>lucas>microcosm>Misc>curses.pl1  88-02-29 19:35:01 EST
curse_touch: procedure(curserptr, curseeptr);
activate_head_curse: procedure(victimptr, curse_type) returns(bit(1) aligned);

%cvideo#d010>lucas>microcosm>Misc>helpers.pl1  88-02-29 19:35:03 EST
accessable: procedure(objptr) recursive returns(bit(1) aligned);
announce_object: procedure(objptr);
at_water: procedure returns(bit(1) aligned);
drop_object_in_hand: procedure(whoptr);
auto_teleport: procedure(whoptr, where, entry_mode);
available: procedure(container_noid, x, y) returns(bit(1) aligned);
cancel_event: procedure(event);
change_containers: procedure(obj_noid, new_container_noid, new_position, cp)
heap_space_available: procedure (dmy) returns (bit(1) aligned);
dequeue_player: procedure(whatptr);
destroy_contents: procedure(containerptr);
empty_handed: procedure(whoptr) returns(bit(1) aligned);
enqueue_player: procedure(whatptr);
getable: procedure(objptr) returns(bit(1) aligned);
grabable: procedure(objptr) returns(bit(1) aligned);
goto_new_region: procedure(whoptr, where, direction, transition_type);
lights_off: procedure(whoptr);
lights_on: procedure(whoptr);
holding: procedure(objptr) returns(bit(1) aligned);
wearing: procedure(objptr) returns(bit(1) aligned);
holding_class: procedure(class_number) returns(bit(1) aligned);
item_value: procedure(itemptr) returns(binary(15));
kill_avatar: procedure(victimptr);
object_broadcast: procedure(obj_noid, text);
object_say: procedure(obj_noid, text);
ghost_say: procedure(obj_noid, text);
tset: procedure(tokenptr, amount);
tget: procedure(tokenptr) returns(binary(31));
pay_to: procedure(whoptr, amount) returns(bit(1) aligned);
random: procedure(top) returns(binary(15));
random_time_in_the_future: procedure returns(binary(31));
schedule_event: procedure(objptr, event_procedure, delay) returns(pointer);
     declare event_procedure entry variable;
spend: procedure(amount) returns(binary(15));
spend_check: procedure(amount) returns(bit(1) aligned);
vectorize: procedure(objptr) returns(character(256) varying);
region_entry_daemon: procedure(direction, transition_type, old_orientation, from
_region);
x_invert: procedure(x) returns(binary(15));
y_invert: procedure(y) returns(binary(15));
x_scale: procedure(x) returns(binary(15));
y_scale: procedure(y) returns(binary(15));
inc_record: procedure(whoptr, record);
set_record: procedure(whoptr, record, value);
max_record: procedure(whoptr, record, value);
change_region_fail: procedure(who_noid);
lookfor_string: procedure(sourcestring,substring) returns (binary(15));
lowercase: procedure(mixedstring) returns (character(256) varying);
unescape_string: procedure(string);
     getchar: procedure returns(binary(15));
     ungetc: procedure(c);
     process_escape: procedure returns(binary(15));
     decode_digit: procedure(c, radix) returns(binary(15));
     control_character: procedure(c) returns(binary(15));

%cvideo#d010>lucas>microcosm>Misc>magic.pl1  88-02-29 19:35:30 EST
generic_MAGIC: procedure;
switch_reply: procedure;
initialize_magic: procedure;
generic_HELP_MAGIC: procedure;
magic_vendo_info: procedure(magicptr) returns(character(114) varying);
illegal_magic: procedure;
change_user_height: procedure;
make_target_avatar_jump: procedure;
avatar_target_check: procedure(targetptr) returns(bit(1) aligned);
make_other_avatars_turn_blue: procedure;
send_target_avatar_home: procedure;
switch_give_user_cooties: procedure;
switch_start_end_capture_flag: procedure;
recover_amulet: procedure;
switch_region_rally_winner: procedure;
change_avatar_style: procedure;
switch_gameshow_buzzer: procedure;
make_user_moonwalk: procedure;
switch_reset_chess: procedure;
switch_reset_checkers: procedure;
switch_reset_backgammon: procedure;
reset_generic_boardgame: procedure(pieces, x_init, y_init, o_init, g_init);
tally_vote: procedure;
god_tool: procedure;
god_tool_revisited: procedure;
modify_variable: procedure(changed_field, offset, new_value);
binding_machine: procedure;
bursting_machine: procedure;
copy_machine: procedure;
take_user_to_an_avatar: procedure;
the_vaultkeeper: procedure;
free_dispenser: procedure;
money_tree: procedure;
dispense: procedure(new_class,new_x,new_y) returns(pointer);
magic_opener: procedure;
magic_opener_revisited: procedure;
lottery: procedure;
lottery_revisited: procedure;
lottery_payoff: procedure;
death_magic: procedure;
random_porter: procedure;

%cvideo#d010>lucas>microcosm>Misc>messages.pl1  88-02-29 19:35:46 EST
 *   Procedures to send messages to the home system.
n_msg_0: procedure(to_object, message_number);
n_msg_1: procedure(to_object, message_number, arg1);
n_msg_2: procedure(to_object, message_number, arg1, arg2);
n_msg_3: procedure(to_object, message_number, arg1, arg2, arg3);
n_msg_4: procedure(to_object, message_number, arg1, arg2, arg3, arg4);
n_msg_5: procedure(to_object, message_number, arg1, arg2, arg3, arg4, arg5);
n_msg_6: procedure(to_object, message_number, arg1, arg2, arg3, arg4, arg5, arg6
);
n_msg_1_s: procedure(to_object, message_number, arg1, args);
n_msg_2_s: procedure(to_object, message_number, arg1, arg2, args);
n_msg_3_s: procedure(to_object, message_number, arg1, arg2, arg3, args);
```

```
n_msg_s: procedure(to_object, message_number, args);
send_n_msg: procedure(to_objectptr, message_number, count);
b_msg_0: procedure(to_object, message_number);
b_msg_1: procedure(to_object, message_number, arg1);
b_msg_2: procedure(to_object, message_number, arg1, arg2);
b_msg_3: procedure(to_object, message_number, arg1, arg2, arg3);
b_msg_4: procedure(to_object, message_number, arg1, arg2, arg3, arg4);
b_msg_5: procedure(to_object, message_number, arg1, arg2, arg3, arg4, arg5);
b_msg_7: procedure(to_object, message_number, arg1, arg2, arg3, arg4, arg5, arg6
, arg7);
b_msg_1_s: procedure(to_object, message_number, arg1, args);
b_msg_s: procedure(to_object, message_number, args);
send_b_msg: procedure(to_objectptr, message_number, count);
p_msg_0: procedure(to_object, to_whom, message_number);
p_msg_1: procedure(to_object, to_whom, message_number, arg1);
p_msg_s: procedure(to_object, to_whom, message_number, args);
p_msg_1_s: procedure(to_object, to_whom, message_number, arg1, args);
send_p_msg: procedure(to_objectptr, to_whomptr, message_number, count);
r_msg_1: procedure(arg1);
r_msg_2: procedure(arg1, arg2);
r_msg_3: procedure(arg1, arg2, arg3);
r_msg_4: procedure(arg1, arg2, arg3, arg4);
r_msg_5: procedure(arg1, arg2, arg3, arg4, arg5);
r_msg_1_s: procedure(arg1, args);
r_msg_2_s: procedure(arg1, arg2, args);
r_msg_3_s: procedure(arg1, arg2, arg3, args);
r_msg_s: procedure(args);
send_r_msg: procedure(count);
e_msg_1: procedure(to_object, exclude_whom, message_number, arg1);
e_msg_s: procedure(to_object, exclude_whom, message_number, args);
send_e_msg: procedure(to_objectptr, exclude_whomptr, message_number, count);
send_bogus_track_sector_update: procedure(who_noid, track, sector, val);
send_bogus_customize_response: procedure(who_noid);

%cvideo#d010>lucas>microcosm>Misc>width.pl1  88-02-29 19:36:13 EST
old_check_path: procedure(target_noid, x, y, new_x, new_y, flip_path);
check_path: procedure(target_noid, x, y, new_x, new_y, flip_path);
check_y_line: procedure(x, start_y, end_y) returns(binary(15));
x_overlap: procedure(objptr, avatar_x, tolerance) returns(bit(1) aligned);
old_adjacent: procedure(objptr) returns(bit(1) aligned);
adjacent: procedure(objptr) returns(bit(1) aligned);
old_elsewhere: procedure(objptr) returns(bit(1) aligned);
elsewhere: procedure(objptr) returns(bit(1) aligned);
here: procedure(objptr) returns(bit(1) aligned);

%cvideo#d010>lucas>microcosm>Actions>actions_clothing.pl1  88-02-29 19:36:18 ES
generic_WEAR: procedure;
generic_REMOVE: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_container.pl1  88-02-29 19:36:20 E
generic_CLOSECONTAINER: procedure;
generic_OPENCONTAINER: procedure;
generic_SET_OPEN_BITS: procedure;
lock_HELP: procedure(item_name, key_number, open_flags);

%cvideo#d010>lucas>microcosm>Actions>actions_door.pl1  88-02-29 19:36:21 EST
generic_CLOSE: procedure;
generic_OPEN: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_gpt.old.pl1  88-02-29 19:36:24 EST
```

```
generic_GET: procedure;
generic_PUT: procedure;
generic_THROW: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_gpt.pl1  88-02-29 19:36:26 EST
generic_GET: procedure;
generic_PUT: procedure;
generic_THROW: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_help.pl1  88-02-29 19:36:27 EST
generic_HELP: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_mail.pl1  88-02-29 19:36:28 EST
generic_READMAIL: procedure;
dead_generic_READMAIL_result: procedure (message_id,more_mail);
generic_SENDMAIL: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_music.pl1  88-02-29 19:36:30 EST
generic_OFFPLAYER: procedure;
generic_ONPLAYER: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_oracle.pl1  88-02-29 19:36:32 EST
generic_ASK: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_switch.pl1  88-02-29 19:36:34 EST
generic_OFF: procedure;
generic_ON: procedure;

%cvideo#d010>lucas>microcosm>Actions>actions_weapon.pl1  88-02-29 19:36:35 EST
generic_ATTACK: procedure;
is_ranged_weapon: procedure(class) returns(bit(1));
damage_avatar: procedure(whoptr, weaponptr) returns(binary(15));
damage_object: procedure(targetptr, weaponptr) returns(binary(15));
     /* For now, always destroy the target.  A more sophisticated procedure
damageable: procedure(class) returns(bit(1) aligned);

%cvideo#d010>lucas>microcosm>Actions>old_actions_gpt.pl1  88-02-29 19:36:41 EST
generic_GET: procedure;
generic_PUT: procedure;
generic_THROW: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_amulet.pl1  88-02-29 19:36:42 EST
initialize_class_amulet: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_aquarium.pl1  88-02-29 19:36:43 EST
initialize_class_aquarium: procedure;
aquarium_FEED: procedure;
Fish_Fed: procedure;
Fish_Die: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_atm.pl1  88-02-29 19:36:45 EST
initialize_class_atm: procedure;
atm_DEPOSIT: procedure;
atm_WITHDRAW: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_avatar.host_echo.pl1  88-02-29 19:36
initialize_class_avatar: procedure;
avatar_GRAB: procedure;
avatar_HAND: procedure;
avatar_POSTURE: procedure;
```

```
avatar_SPEAK: procedure;
avatar_ESP: procedure;
avatar_WALK: procedure;
avatar_NEWREGION: procedure;
avatar_IDENTIFY: procedure;
avatar_SITORSTAND: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_avatar.pl1  88-02-29 19:36:51 EST
initialize_class_avatar: procedure;
avatar_DISCORPORATE: procedure;
avatar_GRAB: procedure;
avatar_HAND: procedure;
avatar_POSTURE: procedure;
avatar_SPEAK: procedure;
avatar_ESP: procedure;
avatar_WALK: procedure;
avatar_NEWREGION: procedure;
holding_restricted_object: procedure(whoptr) returns(bit(1));
avatar_IDENTIFY: procedure;
avatar_SITORSTAND: procedure;
avatar_TOUCH: procedure;
buzzify: procedure(text) returns(character(TEXT_LENGTH) varying);
avatar_FNKEY: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_bag.pl1  88-02-29 19:36:57 EST
initialize_class_bag: procedure;
bag_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_ball.pl1  88-02-29 19:36:59 EST
initialize_class_ball: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_bed.pl1  88-02-29 19:37:00 EST
initialize_class_bed: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_book.pl1  88-02-29 19:37:01 EST
initialize_class_book: procedure;
book_READ: procedure;
book_HELP: procedure;
send_book_title: procedure(title, use_flag);
book_vendo_info: procedure(bookptr) returns(character(114) varying);

%cvideo#d010>lucas>microcosm>Classes>class_boomerang.pl1  88-02-29 19:37:04 EST
initialize_class_boomerang: procedure;
boomerang_THROW: procedure;
Boomerang_Return: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_bottle.pl1  88-02-29 19:37:06 EST
initialize_class_bottle: procedure;
bottle_FILL: procedure;
bottle_POUR: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_box.pl1  88-02-29 19:37:09 EST
initialize_class_box: procedure;
box_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_bridge.pl1  88-02-29 19:37:11 EST
initialize_class_bridge: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_building.pl1  88-02-29 19:37:12 EST
initialize_class_building: procedure;
```

```
%cvideo#d010>lucas>microcosm>Classes>class_bureaucrat.pl1  88-02-29 19:37:13 ES
initialize_class_bureaucrat: procedure;
bureaucrat_ASK: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_bush.pl1  88-02-29 19:37:14 EST
initialize_class_bush: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_chair.pl1  88-02-29 19:37:15 EST
initialize_class_chair: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_changomatic.pl1  88-02-29 19:37:17 E
initialize_class_changomatic: procedure;
changomatic_CHANGE: procedure;
changeable: procedure(objptr) returns(bit(1));
neighbor_changeable: procedure(objptr) returns(bit(1));

%cvideo#d010>lucas>microcosm>Classes>class_chest.pl1  88-02-29 19:37:20 EST
initialize_class_chest: procedure;
chest_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_club.pl1  88-02-29 19:37:23 EST
initialize_class_club: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_coke_machine.pl1  88-02-29 19:37:24
initialize_class_coke_machine: procedure;
coke_machine_PAY: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_compass.pl1  88-02-29 19:37:24 EST
initialize_class_compass: procedure;
compass_DIRECT: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_couch.pl1  88-02-29 19:37:27 EST
initialize_class_couch: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_countertop.pl1  88-02-29 19:37:28 ES
initialize_class_countertop: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_crystal_ball.pl1  88-02-29 19:37:29
initialize_class_crystal_ball: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_die.pl1  88-02-29 19:37:30 EST
initialize_class_die: procedure;
die_ROLL: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_display_case.pl1  88-02-29 19:37:32
initialize_class_display_case: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_door.pl1  88-02-29 19:37:34 EST
initialize_class_door: procedure;
door_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_dropbox.pl1  88-02-29 19:37:36 EST
initialize_class_dropbox: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_drugs.pl1  88-02-29 19:37:36 EST
initialize_class_drugs: procedure;
drugs_TAKE: procedure;
initialize_drugs: procedure;
heal_avatar: procedure;
```

```
poison_avatar: procedure;
turn_avatar_black: procedure;
drugs_vendo_info: procedure(drugsptr) returns(character(114) varying);
drugs_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_elevator.pl1  88-02-29 19:37:41 EST
initialize_class_elevator: procedure;
elevator_ZAPTO: procedure;
elevator_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_escape_dev.pl1  88-02-29 19:37:43 ES
initialize_class_escape_dev: procedure;
escape_dev_BUGOUT: procedure;
escape_dev_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_fake_gun.pl1  88-02-29 19:37:45 EST
initialize_class_fake_gun: procedure;
fake_gun_FAKESHOOT: procedure;
fake_gun_RESET: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_fence.pl1  88-02-29 19:37:47 EST
initialize_class_fence: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_flag.pl1  88-02-29 19:37:49 EST
initialize_class_flag: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_flashlight.pl1  88-02-29 19:37:49 ES
initialize_class_flashlight: procedure;
flashlight_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_flat.pl1  88-02-29 19:37:51 EST
initialize_class_flat: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_floor_lamp.pl1  88-02-29 19:37:53 ES
initialize_class_floor_lamp: procedure;
floor_lamp_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_fortune_machine.pl1  88-02-29 19:37:
initialize_class_fortune_machine: procedure;
fortune_machine_PAY: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_fountain.pl1  88-02-29 19:37:57 EST
initialize_class_fountain: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_frisbee.pl1  88-02-29 19:37:58 EST
initialize_class_frisbee: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_game_piece.pl1  88-02-29 19:37:59 ES
initialize_class_game_piece: procedure;
game_piece_CHANGE: procedure;
game_piece_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_garbage_can.pl1  88-02-29 19:38:01 E
initialize_class_garbage_can: procedure;
garbage_can_FLUSH: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_gate.pl1  88-02-29 19:38:04 EST
initialize_class_gate: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_gemstone.pl1  88-02-29 19:38:05 EST
```

```
initialize_class_gemstone: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_ghost.pl1  88-02-29 19:38:05 EST
initialize_class_ghost: procedure;
ghost_WALK: procedure;
ghost_NEWREGION: procedure;
ghost_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_glue.pl1  88-02-29 19:38:07 EST
initialize_class_glue: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_grenade.pl1  88-02-29 19:38:08 EST
initialize_class_grenade: procedure;
grenade_PULLPIN: procedure;
Grenade_Explosion: procedure(arg);
grenade_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_ground.pl1  88-02-29 19:38:11 EST
initialize_class_ground: procedure;
ground_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_gun.pl1  88-02-29 19:38:14 EST
initialize_class_gun: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_hand_of_god.pl1  88-02-29 19:38:15 E
initialize_class_hand_of_god: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_hat.pl1  88-02-29 19:38:16 EST
initialize_class_hat: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_head.pl1  88-02-29 19:38:17 EST
initialize_class_head: procedure;
head_WEAR: procedure;
head_REMOVE: procedure;
head_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_hole.pl1  88-02-29 19:38:19 EST
initialize_class_hole: procedure;
hole_CLOSE: procedure;
hole_OPEN: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_hot_tub.pl1  88-02-29 19:38:22 EST
initialize_class_hot_tub: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_house_cat.pl1  88-02-29 19:38:23 EST
initialize_class_house_cat: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_instant_object.pl1  88-02-29 19:38:2
initialize_class_instant_object: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_jacket.pl1  88-02-29 19:38:26 EST
initialize_class_jacket: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_jukebox.pl1  88-02-29 19:38:27 EST
initialize_class_jukebox: procedure;
jukebox_PAY: procedure;
jukebox_CATALOG: procedure;
jukebox_SELECT: procedure;
lookup_selection: procedure(musicptr, music_max, choice) returns(pointer);
```

```
%cvideo#d010>lucas>microcosm>Classes>class_key.pl1  88-02-29 19:38:30 EST
initialize_class_key: procedure;
key_vendo_info: procedure(keyptr) returns(character(114) varying);

%cvideo#d010>lucas>microcosm>Classes>class_knick_knack.pl1  88-02-29 19:38:33 E
initialize_class_knick_knack: procedure;
knick_knack_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_knife.pl1  88-02-29 19:38:35 EST
initialize_class_knife: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_magic_lamp.pl1  88-02-29 19:38:36 ES
initialize_class_magic_lamp: procedure;
magic_lamp_RUB: procedure;
magic_lamp_WISH: procedure;
Genie_Gets_Impatient: procedure(arg);

%cvideo#d010>lucas>microcosm>Classes>class_magic_staff.pl1  88-02-29 19:38:38 E
initialize_class_magic_staff: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_magic_wand.pl1  88-02-29 19:38:40 ES
initialize_class_magic_wand: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_mailbox.pl1  88-02-29 19:38:41 EST
initialize_class_mailbox: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_matchbook.pl1  88-02-29 19:38:42 EST
initialize_class_matchbook: procedure;
matchbook_README: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_movie_camera.pl1  88-02-29 19:38:43
initialize_class_movie_camera: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_pants.pl1  88-02-29 19:38:46 EST
initialize_class_pants: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_paper.pl1  88-02-29 19:38:46 EST
initialize_class_paper: procedure;
paper_GET: procedure;
generic_READMAIL_result: procedure(message_id, more_mail);
paper_PUT: procedure;
paper_THROW: procedure;
paper_READ: procedure;
paper_WRITE: procedure;
paper_SENDMAIL: procedure;
paper_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_pawn_machine.pl1  88-02-29 19:38:51
initialize_class_pawn_machine: procedure;
pawn_machine_MUNCH: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_picture.pl1  88-02-29 19:38:53 EST
initialize_class_picture: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_plant.pl1  88-02-29 19:38:54 EST
initialize_class_plant: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_plaque.pl1  88-02-29 19:38:55 EST
initialize_class_plaque: procedure;
plaque_READ: procedure;
```

```
plaque_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_pond.pl1  88-02-29 19:38:57 EST
initialize_class_pond: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_region.pl1  88-02-29 19:38:58 EST
initialize_class_region: procedure;
region_DESCRIBE: procedure;
region_LEAVE: procedure;
region_IM_ALIVE: procedure;
region_CUSTOMIZE: procedure;
region_FINGER_IN_QUE: procedure;
region_I_AM_HERE: procedure;
region_PROMPT_REPLY: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_ring.pl1  88-02-29 19:39:02 EST
initialize_class_ring: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_river.pl1  88-02-29 19:39:03 EST
initialize_class_river: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_rock.pl1  88-02-29 19:39:04 EST
initialize_class_rock: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_roof.pl1  88-02-29 19:39:06 EST
initialize_class_roof: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_safe.pl1  88-02-29 19:39:07 EST
initialize_class_safe: procedure;
safe_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_security_dev.pl1  88-02-29 19:39:09
initialize_class_security_dev: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_sensor.pl1  88-02-29 19:39:11 EST
initialize_class_sensor: procedure;
sensor_SCAN: procedure;
initialize_sensors: procedure;
sense_weapons: procedure returns(binary(15));
sensor_HELP: procedure;
sensor_vendo_info: procedure(sensorptr) returns(character(114) varying);

%cvideo#d010>lucas>microcosm>Classes>class_sex_changer.pl1  88-02-29 19:39:15 E
initialize_class_sex_changer: procedure;
sex_changer_SEXCHANGE: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_shirt.pl1  88-02-29 19:39:17 EST
initialize_class_shirt: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_short_sign.pl1  88-02-29 19:39:18 ES
initialize_class_short_sign: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_shovel.pl1  88-02-29 19:39:19 EST
initialize_class_shovel: procedure;
shovel_DIG: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_sign.pl1  88-02-29 19:39:20 EST
initialize_class_sign: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_skateboard.pl1  88-02-29 19:39:21 ES
```

```
initialize_class_skateboard: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_sky.pl1  88-02-29 19:39:23 EST
initialize_class_sky: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_spray_can.pl1  88-02-29 19:39:24 EST
initialize_class_spray_can: procedure;
spray_can_SPRAY: procedure;
spray_can_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_stereo.pl1  88-02-29 19:39:26 EST
initialize_class_stereo: procedure;
stereo_LOAD: procedure;
stereo_UNLOAD: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_street.pl1  88-02-29 19:39:29 EST
initialize_class_street: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_streetlamp.pl1  88-02-29 19:39:31 ES
initialize_class_streetlamp: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_stun_gun.pl1  88-02-29 19:39:31 EST
initialize_class_stun_gun: procedure;
stun_gun_STUN: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_super_trapezoid.pl1  88-02-29 19:39:
initialize_class_super_trapezoid: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_switch.pl1  88-02-29 19:39:35 EST
initialize_class_switch: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_table.pl1  88-02-29 19:39:36 EST
initialize_class_table: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_tape.pl1  88-02-29 19:39:37 EST
initialize_class_tape: procedure;
tape_READLABEL: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_teleport.pl1  88-02-29 19:39:39 EST
initialize_class_teleport: procedure;
teleport_PAY: procedure;
teleport_ZAPTO: procedure;
squish: procedure(instr) returns(character(256) varying);
activate_teleporter: procedure (destination, x_value, y_value);
area_code: procedure(portptr) returns(character(20) varying);
teleport_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_test.pl1  88-02-29 19:39:44 EST
initialize_class_test: procedure;
generic_TEST: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_ticket.pl1  88-02-29 19:39:45 EST
initialize_class_ticket: procedure;
ticket_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_tokens.pl1  88-02-29 19:39:47 EST
initialize_class_tokens: procedure;
tokens_PAY: procedure;
tokens_SPLIT : procedure;
tokens_HELP: procedure;
```

```
%cvideo#d010>lucas>microcosm>Classes>class_trapezoid.pl1  88-02-29 19:39:49 EST
initialize_class_trapezoid: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_tree.pl1  88-02-29 19:39:49 EST
initialize_class_tree: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_vendo_front.pl1  88-02-29 19:39:51 E
initialize_class_vendo_front: procedure;
vendo_VSELECT: procedure;
select_out_of_order: procedure;
vendo_PAY: procedure;
clone: procedure(objnoid, new_x, new_y) returns(pointer);
vendo_HELP: procedure(frontptr);
vendo_front_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_vendo_inside.pl1  88-02-29 19:39:55
initialize_class_vendo_inside: procedure;
vendo_inside_HELP: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_wall.pl1  88-02-29 19:39:57 EST
initialize_class_wall: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_window.pl1  88-02-29 19:39:58 EST
initialize_class_window: procedure;

%cvideo#d010>lucas>microcosm>Classes>class_windup_toy.pl1  88-02-29 19:40:00 ES
initialize_class_windup_toy: procedure;
windup_toy_WIND: procedure;

%cvideo#d010>lucas>microcosm>microcosm.incl.pl1  88-02-29 19:41:12 EST

/*
 *   microcosm.incl.pl1
 *
 *   General purpose include file for MicroCosm(TM).
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   8-April-1986
 */

%include ’defs_message.incl.pl1’;
%include ’defs_class.incl.pl1’;
%include ’defs_struct.incl.pl1’;
%include ’defs_statistics.incl.pl1’;

%replace NULL  by 0;
%replace FALSE by 0;
%replace BOING_FAILURE by 2;
%replace TRUE  by 1;
%replace false by ’0’b;
%replace true  by ’1’b;

%replace TEXT_LENGTH by 256;
%replace PAPER_LENGTH by 640;

/* Avatar constants */
%replace MAIL_SLOT by 4;
%replace HANDS by 5;
```

```
%replace HEAD by 6;
%replace AVATAR_CAPACITY by 8;
%replace UNWEARABLE by 0; /* historical aberration */

/* Container constants */
%replace OPEN_BIT by 1;
%replace UNLOCKED_BIT by 2;

/* Curse constants */
%replace CURSE_NONE    by 0;
%replace CURSE_COOTIES by 1;
%replace CURSE_SMILEY  by 2;
%replace CURSE_MUTANT  by 3;
%replace CURSE_FLY     by 4;

/* Magic lamp constants */
%replace MAGIC_LAMP_WAITING by 0;
%replace MAGIC_LAMP_GENIE by 1;

/* instance_head general flag constants */
%replace RESTRICTED by 1;
%replace MODIFIED   by 2;

/* region nitty_bits constants */
%replace WEAPONS_FREE by 1;
%replace STEAL_FREE by 2;

/* avatar nitty_bit constants */
%replace VOTED_FLAG by 3;
%replace GOD_FLAG by 4;
%replace MISC_FLAG1 by 5;
%replace MISC_FLAG2 by 6;
%replace MISC_FLAG3 by 7;

/* object nitty-bits constants */
%replace DOOR_AVATAR_RESTRICTED_BIT by 32;
%replace DOOR_GHOST_RESTRICTED_BIT by 31;

%replace THE_REGION by 0;

%replace ObjectsPerRegion by 255;
%replace UsersPerRegion by 6;
%replace regions_per_process by 10;

%include 'capacity.equates.incl.pl1';
/*%replace C64_HEAP_SIZE                  by 14470;   Ver 4.3 - 3/13/87 */
/*%replace C64_HEAP_SIZE                  by 14214;   Ver 5.2 - 5/22/87 */
/*%replace C64_HEAP_SIZE                  by 14982;   Ver 5.6 - 6/22/87 */
/*%replace C64_HEAP_SIZE                  by 16500;   Ver 5.9 - 7/17/87 */
%replace C64_HEAP_SIZE                    by 16244;   /* Ver 6.3 - 10/5/87 */
%include 'region.structs.incl.pl1';

declare 1 o based %include struct_gen_object;
declare 1 u based %include struct_user;

declare 1 Class_Table(0:255) external %include struct_class;

declare 1 avatar based(avatarptr) %include struct_avatar;
declare 1 self based(selfptr) %include struct_gen_object;
declare c(0:255) binary(15) based; /* Pointer qualifier for contents array */

%include 'instance_head.def.incl.pl1';

declare request(258) character(1) defined(request_string);
declare put_success bit(1) aligned external;
declare throw_success bit(1) aligned external;
%replace FIRST by 3;
%replace SECOND by 4;
%replace THIRD by 5;
%replace FOURTH by 6;
%replace FIFTH by 7;

declare COLLISION_ON bit(1) external init('1'b);
declare ADJACENCY_ON bit(1) external init('1'b);

%replace C64_XPOS_OFFSET           by  7;
%replace C64_ypos_offset           by  8;
%replace C64_orient_offset         by  9;
%replace C64_gr_state_offset       by  10;
%replace C64_contained_offset      by  11;
%replace C64_TOKEN_denom_offset    by  15;
%replace C64_text_offset           by  15;
%replace C64_customize_offset      by  26;
%replace C64_destx_offset          by  28;
%replace C64_desty_offset          by  29;

%replace OPERATE        by 152;

%replace AUTO_TELEPORT_DIR by 4;
%replace WALK_ENTRY by 0;
%replace TELEPORT_ENTRY by 1;
%replace DEATH_ENTRY by 2;



%cvideo#d010>lucas>microcosm>region.structs.incl.pl1  88-02-29 19:41:51 EST

%page;

declare RoomNumber                       binary(15) external initial(0);
declare RoomPtr                          pointer external;
declare RoomPtrs(regions_per_process)    pointer external;
declare CapMonPtr                        pointer external;
declare CapMonPtrs(regions_per_process) pointer external;

declare 1 RoomDBank                      based (RoomPtr),
          2  Region                      binary(31),
          2  Region_name                 character(20),
          2  RoomQId                     binary(31),
          2  RoomBQId                    binary(31),
          2  last_noid                   binary(15),
          2  total_ghosts                binary(15),
          2  Pending                     pointer,
          2  flags,
             3 private                   bit(1),
             3 owner_here                bit(1),
             3 initialized               bit(1),
             3 filler_flags              bit(13),
          2  current_region,
             3 lighting                 binary(15),
             3 depth                    binary(15),
             3 neighbor(4)              binary(31),
             3 exit_type(4)             binary(15),
             3 restriction(4)           bit(1),
             3 nitty_bits(28)           bit(1),
             3 max_avatars              binary(15),
             3 owner                    binary(31),
             3 entry_proc               binary(15),
             3 exit_proc                binary(15),
             3 class_group              binary(15),
             3 orientation              binary(15),
             3 object_count             binary(15),
             3 space_usage              binary(15),
             3 town_dir                 character(1),
             3 port_dir                 character(1),
          2  oracle,
             3 object                   binary(15),
             3 person                   binary(15),
             3 control                  pointer,
          2  UserList(UsersPerRegion)   pointer,
          2  ObjList(0:ObjectsPerRegion) pointer,
          2  GhostList                  pointer,
          2  Block_addr                 pointer;

declare 1 RoomCMon                      based (CapMonPtr),
          2  class_ref_count(0:MAX_CLASS_NUMBER)      binary(15),
          2  resource_ref_count(NUMBER_OF_RESOURCES)  binary(15);

declare 1 Memory_Block                  based,
          2  free                       bit(64),
          2  entry(64)                  char(40);

declare  DayNight   bin(15) external initial(0);

declare 1         player    based                  /* entry for avatar    */
%include 'struct_user.incl.pl1';

declare 1         object    based                  /* entry for object    */
%include 'instance_head.incl.pl1';,
         2        param1    pointer,               /* to contents list    */
         2        param2    char(1);               /* depends on class    */


declare  current_noid                   binary (15) external;
declare  current_request                binary (15) external;
declare  current_header                 char(1) external;
declare  current_qid                    binary (31) external;
declare  selfptr                        pointer external initial(null());
declare  avatarptr                      pointer external initial(null());
declare  userptr                        pointer external initial(null());

declare  request_string char(646) var external;

declare fan_cnt                         binary(15) external;
declare fan_list(UsersPerRegion+200)    pointer external;  /* + 200 ghosts */

declare 1 now_in    external,
          2 last    bin(31)        initial(0),
          2 count   bin(15)        initial(0),
          2 line    (4) char(40) var;

declare 1 enter_info  based,              /* remember info for later retry */
          2 room         binary(31),
          2 user         binary(31),
          2 que          binary(31),
          2 attempts     binary(15),
          2 params       char(254) var;

declare 1 fountain  based,
          2 type         binary(15),
          2 which_room   binary(15),
          2 start_time   binary(31),
          2 end_time     binary(31),
          2 interval     binary(31),
          2 msg_text     char(100) var;

declare bit_mask(UsersPerRegion)   bit(32) external;   /* user index bit mask */


%replace Separation_Char by 144;


%cvideo#d010>lucas>microcosm>Structs>struct_user.incl.pl1  88-02-29 19:43:03 ES

/*
 *   struct_user.incl.pl1
 *
 *   Struct stub for UserList structure.
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   9-April-1986
 *
 */
,    2    U_Name              character(10) varying,
     2    U_Id                binary(31),
     2    U_Q_Id              binary(31),
     2    U_Q                 pointer,
     2    U_version           binary(15),
     2    object_slot         binary(15),
     2    esp                 ,
          3 to_uid            binary(31),
          3 to_qid            binary(31),
          3 que               pointer,
          3 lines             binary(15),
     2    last_mail_ts        binary(31),
     2    auto_destination    binary(31),     /* TEMP - should be in context */
     2    auto_mode           binary(31),     /* TEMP - should be in context */
     2    flags               ,
          3    U_mail              bit(1),
          3    cr_pending          bit(1),
          3    online              bit(1),
          3    incoming            bit(1),
          3    new_session         bit(1),
          3    ck_last_login       bit(1),
          3    filler              bit(10);

%cvideo#d010>lucas>microcosm>Structs>struct_vendo_front.incl.pl1  88-02-29 19:4
```

```
 /*
 *   struct_vendo_front.incl.pl1
 *
 *   Struct stub for vendo_front instance descriptor.
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   20-June-1986
 *
 */
,    2    common_head         like instance_head,
     2    contents            pointer,
     2    class_specific      ,
           3   open_flags     binary(15),
           3   key_hi         binary(15),
           3   key_lo         binary(15),
           3   item_price     binary(15),
           3   display_item   binary(15),
/* Internal use below here */
           3   prices(0:9)    binary(15),
           3   take           binary(31);


%cvideo#d010>lucas>microcosm>Structs>struct_ball.incl.pl1  88-02-29 19:43:43 ES

/*
 *   struct_ball.incl.pl1
 *
 *   Struct stub for ball instance descriptor.
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   9-April-1986
 *
 */
,    2    common_head         like instance_head
; /* terminates struct header from include file */

%cvideo#d010>lucas>microcosm>Classes>class_ball.pl1  88-02-29 19:44:05 EST

/*
 *   class_ball.pl1
 *
 *   Ball object behavior module for MicroCosm(TM).
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   9-April-1986
 */

%include 'microcosm.incl.pl1';

%include 'defs_action.incl.pl1';

initialize_class_ball: procedure;

     %replace BALL_REQUESTS by 3;

     declare a(0:BALL_REQUESTS) entry based;
     declare class_ball_actions pointer;
     declare 1 ball based %include struct_ball;

     %replace I by CLASS_BALL;

     Class_Table(I).capacity = 0;
     Class_Table(I).max_requests = BALL_REQUESTS;
     Class_Table(I).alloc_size = size(ball);
     Class_Table(I).pc_state_bytes = 0;
     Class_Table(I).known = true;
     Class_Table(I).opaque_container = false;
     Class_Table(I).filler = false;

     allocate a set(class_ball_actions);
     Class_Table(I).actions = class_ball_actions;

     Class_Table(I).actions->a(HELP) = generic_HELP; /* 0 */
     Class_Table(I).actions->a(GET)   = generic_GET;   /* 1 */
     Class_Table(I).actions->a(PUT)   = generic_PUT;   /* 2 */
     Class_Table(I).actions->a(THROW) = generic_THROW; /* 3 */

end initialize_class_ball;


%cvideo#d010>lucas>microcosm>Classes>class_vendo_front.pl1  88-02-29 19:44:37 E

/*
 *   class_vendo_front.pl1
 *
 *   Vendo front object behavior module for MicroCosm(TM).
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   20-June-1986
 */

%replace VENDO_FRONT_CAPACITY by 10;

%include 'microcosm.incl.pl1';
%include 'defs_helper.incl.pl1';
%include 'defs_action.incl.pl1';
declare book_vendo_info entry(pointer) returns(character(114) varying);
declare drugs_vendo_info entry(pointer) returns(character(114) varying);
declare magic_vendo_info entry(pointer) returns(character(114) varying);
declare key_vendo_info entry(pointer) returns(character(114) varying);
declare sensor_vendo_info entry(pointer) returns(character(114) varying);

initialize_class_vendo_front: procedure;

     %replace VENDO_FRONT_REQUESTS by 5;

     declare a(0:VENDO_FRONT_REQUESTS) entry based;
     declare class_vendo_front_actions pointer;
     declare 1 vendo_front based %include struct_vendo_front;

     %replace I by CLASS_VENDO_FRONT;

     Class_Table(I).capacity = VENDO_FRONT_CAPACITY;
     Class_Table(I).max_requests = VENDO_FRONT_REQUESTS;
     Class_Table(I).alloc_size = size(vendo_front);
     Class_Table(I).pc_state_bytes = 5;
     Class_Table(I).known = true;
     Class_Table(I).opaque_container = true;
     Class_Table(I).filler = false;

     allocate a set(class_vendo_front_actions);
     Class_Table(I).actions = class_vendo_front_actions;

     Class_Table(I).actions->a(HELP)    = vendo_front_HELP;       /* 0 */
     Class_Table(I).actions->a(1)       = illegal;          /* 1 */
     Class_Table(I).actions->a(2)       = illegal;          /* 2 */
     Class_Table(I).actions->a(3)       = illegal;          /* 3 */
     Class_Table(I).actions->a(PAY)     = vendo_PAY;        /* 4 */
     Class_Table(I).actions->a(VSELECT) = vendo_VSELECT;    /* 5 */

end initialize_class_vendo_front;

%replace VENDO_DISPLAY by 1;

vendo_VSELECT: procedure;
     declare 1 vinside based(vinsideptr) %include struct_vendo_inside;
     declare vinsideptr pointer;
     declare 1 self based(selfptr) %include struct_vendo_front;
     declare dummy bit(1) aligned;
     declare save_display_item binary(15);
     declare save_noid binary(15);
     declare loop_counter binary(15);

     vinsideptr = ObjList(self.container);
     if (vinside.contents->c(VENDO_DISPLAY) = NULL) then do;
         call select_out_of_order;
         return;
     end;
     save_display_item = self.display_item;
     loop_counter = 0;
foon:self.display_item = self.display_item + 1;
     if (self.display_item > 9) then self.display_item = 0;
     if (self.contents->c(self.display_item) = NULL) then do ;
          loop_counter = loop_counter + 1;
          if (loop_counter = VENDO_FRONT_CAPACITY) then do;
               self.display_item = save_display_item;
          end; else
               goto foon;
     end;
     save_noid = vinside.contents->c(VENDO_DISPLAY);
     dummy = change_containers(save_noid,
          self.noid, save_display_item, true); /* never fails (decreases mem use
) */
     if (^dummy) then
          call trace_msg ('IMPOSSIBLE VENDO ERROR:  args=' || ltrim(save_noid) |
|
                         ',' || ltrim(self.noid) || ',' ||
                         ltrim(save_display_item) || ', vendo=' ||
                         ltrim(self.obj_id));
     if (^ change_containers((self.contents->c(self.display_item)),
               vinside.noid, VENDO_DISPLAY, true)) then do;
          dummy = change_containers(save_noid, vinside.noid, VENDO_DISPLAY, true
);
          self.display_item = save_display_item;
          call select_out_of_order;
          return;
     end;
     self.item_price = self.prices(self.display_item);
     self.gen_flags(MODIFIED) = true;
     call checkpoint_object (0, self.noid);  /* To insure consistent vendo */
     call r_msg_3(mod(self.item_price, 256),
            divide(self.item_price, 256, 15), self.display_item);
     call n_msg_1(avatarptr, POSTURE$, OPERATE);
     call n_msg_3(selfptr, VSELECT$, mod(self.item_price, 256),
            divide(self.item_price, 256, 15), self.display_item);
end vendo_VSELECT;

select_out_of_order: procedure;
     call object_say(self.noid, ’This machine is out of order.’);
     call r_msg_3(0, 0, 255);           /* fail */
     call trace_msg (’Broken vendo: ’ || ltrim(self.obj_id));
end select_out_of_order;

vendo_PAY: procedure;
     declare 1 vinside based(vinsideptr) %include struct_vendo_inside;
     declare vinsideptr pointer;
     declare 1 self based(selfptr) %include struct_vendo_front;
     declare obj_vector character(256) varying;
     declare new_x binary(15);
     declare new_y binary(15);
     declare junk binary(15);
     declare item_price_lo binary(15);
     declare item_price_hi binary(15);
     declare objptr pointer;

     vinsideptr = ObjList(self.container);
     item_price_lo = mod(self.item_price, 256);
     item_price_hi = divide(self.item_price, 256, 15);
     if (spend_check((self.item_price))) then do;
          new_x = vinside.x + 8;
          new_y = vinside.y;
          call set_bit(new_y, 8);
          objptr = clone(vinside.contents->c(VENDO_DISPLAY), new_x, new_y);
          if (objptr = null()) then do;
               call r_msg_1(BOING_FAILURE);
               return;
          end;
          junk = spend((self.item_price));
          self.take = self.take + self.item_price;
          self.gen_flags(MODIFIED) = true;
          obj_vector = vectorize(objptr);
          call r_msg_3_s(TRUE, item_price_lo, item_price_hi, obj_vector);
          call n_msg_3_s(selfptr, SELL$, avatar.noid, item_price_lo, item_price_
hi, obj_vector);
     end; else do;
          call object_say(self.noid, ’You don’’t have enough money.  The item on
 display costs $’ || ltrim(self.item_price) || ’.’);
          call r_msg_1(FALSE);
     end;
end vendo_PAY;

clone: procedure(objnoid, new_x, new_y) returns(pointer);
     declare objnoid binary(15);
     declare new_x binary(15);
     declare new_y binary(15);
     declare 1 object based(objptr) %include struct_gen_object;
     declare objptr pointer;
     declare 1 newobj based(newobjptr) %include struct_gen_container;
     declare newobjptr pointer;
     declare save_noid binary(15);
     declare save_obj_id binary(31);
     declare save_ptr pointer;
     declare i binary(15);
     declare funnyout(256) character based(funnyoutptr);
     declare funnyoutptr pointer;
     declare funnyin(256) character based(funnyinptr);
     declare funnyinptr pointer;

     objptr = ObjList(objnoid);
     newobjptr = create_object(object.class, 0, 0, 0, 0, 0, 0, 0);
     if (newobjptr = null()) then return(null());
     save_noid = newobj.noid;
     save_obj_id = newobj.obj_id;
     save_ptr = newobj.contents;
     funnyinptr = objptr;
     funnyoutptr = newobjptr;
     do i=1 to divide(Class_Table(object.class).alloc_size, 8, 15);
          funnyout(i) = funnyin(i);
     end;
     newobj.noid = save_noid;
     newobj.obj_id = save_obj_id;
     if (Class_Table(object.class).capacity ^= 0)
          then newobj.contents = save_ptr;
     newobj.x = new_x;
     newobj.y = new_y;
     call set_bit(newobj.y, 8);
     newobj.container = THE_REGION;
     newobj.gen_flags(MODIFIED) = true;
     return(newobjptr);
end clone;

vendo_HELP: procedure(frontptr);
     declare frontptr pointer;
     declare 1 front based(frontptr) %include struct_vendo_front;
     declare displayptr pointer;
     declare interiorptr pointer;
     declare 1 interior based(interiorptr) %include struct_vendo_inside;
     declare 1 display based(displayptr) %include struct_gen_object;
     declare the_message character(114) varying;
     declare info_messages(0:158) character(80) varying static init(
'i',                                    /*   0 -- region */
'i',                                    /*   1 -- avatar */
'm',                                    /*   2 -- amulet */
'-',                                    /*   3 */
'i',                                    /*   4 -- atm */
'GAME PIECE, for board games of all kinds.', /*   5 -- game piece */
'BAG, a useful container.',             /*   6 -- bag */
'BALL, for throwing and playing.',      /*   7 -- ball */
'-',                                    /*   8 */
'-',                                    /*   9 */
'b',                                    /*  10 -- book */
'BOOMERANG, non-functional.',           /*  11 -- boomerang */
'BOTTLE, holds water.',                 /*  12 -- bottle */
'BOX, a useful container.',             /*  13 -- box */
'-',                                    /*  14 */
'-',                                    /*  15 */
```

```
'CLUB.',                                /*  16 -- club */
'COMPASS, shows direction of West Pole.',/*  17 -- compass */
'i',                                    /*  18 -- countertop */
'-',                                    /*  19 */
'i',                                    /*  20 -- crystal ball */
'DIE, for immediate acess to random numbers.', /*  21 -- die */
'i',                                    /*  22 -- display case */
'i',                                    /*  23 -- door */
'i',                                    /*  24 -- dropbox */
'd',                                    /*  25 -- drugs */
'ESCAPE DEVICE, takes you home in a panic.', /*  26 -- escape device */
'GUN.',                                 /*  27 -- fake gun */
'i',                                    /*  28 -- elevator */
'i',                                    /*  29 -- flag */
'LIGHT, illuminates the dark places.',  /*  30 -- flashlight */
'FRISBEE, for throwing and playing',    /*  31 -- frisbee */
'i',                                    /*  32 -- garbage can */
'm',                                    /*  33 -- gemstone */
'-',                                    /*  34 */
'GRENADE.',                             /*  35 -- grenade */
'i',                                    /*  36 -- ground */
'GUN',                                  /*  37 -- gun */
'i',                                    /*  38 -- hand of god */
'-',                                    /*  39 -- hat */
'INSTANT OBJECT PILL',                  /*  40 -- instant object pill */
'-',                                    /*  41 -- jacket */
'k' ,                                   /*  42 -- key */
'KNICK-KNACK of some sort',             /*  43 -- knick knack */
'KNIFE.',                               /*  44 -- knife */
'i',                                    /*  45 -- magic lamp */
'm',                                    /*  46 -- magic staff */
'm',                                    /*  47 -- magic wand */
'i',                                    /*  48 -- mailbox */
'i',                                    /*  49 -- matchbook */
'-',                                    /*  50 */
'-',                                    /*  51 */
'MOVIE CAMERA.',                        /*  52 -- movie camera */
'-',                                    /*  53 */
'PAPER, for notes and mail.',           /*  54 -- paper */
'-',                                    /*  55 */
'i',                                    /*  56 -- short sign */
'i',                                    /*  57 -- sign */
'i',                                    /*  58 -- plant */
'-',                                    /*  59 */
'm',                                    /*  60 -- ring */
'i',                                    /*  61 -- rock */
'-',                                    /*  62 */
'SECURITY DEVICE.',                     /*  63 -- security device */
's',                                    /*  64 -- sensor */
'-',                                    /*  65 */
'-',                                    /*  66 */
'-',                                    /*  67 */
'-',                                    /*  68 */
'i',                                    /*  69 -- sky */
'i',                                    /*  70 -- stereo */
'i',                                    /*  71 -- tape */
'-',                                    /*  72 */
'-',                                    /*  73 */
'i',                                    /*  74 -- teleport booth */
'i',                                    /*  75 -- ticket */
```

```
'i',                                    /*  76 -- tokens */
'-',                                    /*  77 */
'-',                                    /*  78 */
'-',                                    /*  79 */
'i',                                    /*  80 -- wall */
'-',                                    /*  81 */
'WINDUP TOY.',                          /*  82 -- windup toy */
'-',                                    /*  83 */
'CHANGE-O-MATIC, lets you change your Turf.', /*  84 -- changomatic */
'i',                                    /*  85 -- vendo front */
'i',                                    /*  86 -- vendo inside */
'i',                                    /*  87 -- trapezoid */
'i',                                    /*  88 -- hole */
'SHOVEL, for digging holes.',          B/*  89 -- shovel */
'i',                                    /*  90 -- sex changer */
'STUN GUN.',                            /*  91 -- stun gun */
'i',                                    /*  92 -- super trapezoid */
'i',                                    /*  93 -- flat */
'TEST OBJECT!',                         /*  94 -- test */
'BODY SPRAYER, lets you change your body colors.', /*  95 -- spray can */
'i',                                    /*  96 -- pawn machine */
'i',                                    /*  97 -- switch / immobile magic */
'i',                                    /*  98 -- "glue" */
'-',                                    /*  99 */
'-',                                    /* 100 */
'-',                                    /* 101 */
'-',                                    /* 102 */
'-',                                    /* 103 */
'-',                                    /* 104 */
'-',                                    /* 105 */
'-',                                    /* 106 */
'-',                                    /* 107 */
'-',                                    /* 108 */
'-',                                    /* 109 */
'-',                                    /* 110 */
'-',                                    /* 111 */
'-',                                    /* 112 */
'-',                                    /* 113 */
'-',                                    /* 114 */
'-',                                    /* 115 */
'-',                                    /* 116 */
'-',                                    /* 117 */
'-',                                    /* 118 */
'-',                                    /* 119 */
'-',                                    /* 120 */
'-',                                    /* 121 */
'-',                                    /* 122 */
'-',                                    /* 123 */
'-',                                    /* 124 */
'-',                                    /* 125 */
'-',                                    /* 126 */
'HEAD.',                                /* 127 -- head */
'-',                                    /* 128 */
'i',                                    /* 129 -- aquarium */
'i',                                    /* 130 -- bed */
'i',                                    /* 131 -- bridge */
'i',                                    /* 132 -- building */
'i',                                    /* 133 -- bush */
'i',                                    /* 134 -- chair */
'i',                                    /* 135 -- chest */
```

```
'i',                                    /* 136 -- coke machine */
'i',                                    /* 137 -- couch */
'i',                                    /* 138 -- fence */
'i',                                    /* 139 -- floor lamp */
'i',                                    /* 140 -- fortune machine */
'i',                                    /* 141 -- fountain */
'-',                                    /* 142 */
'i',                                    /* 143 -- house cat */
'i',                                    /* 144 -- hot tub */
'i',                                    /* 145 -- jukebox */
'-',                                    /* 146 */
'i',                                    /* 147 -- pond */
'i',                                    /* 148 -- river */
'i',                                    /* 149 -- roof */
'i',                                    /* 150 -- safe */
'-',                                    /* 151 */
'i',                                    /* 152 -- picture */
'i',                                    /* 153 -- street */
'i',                                    /* 154 -- streetlamp */
'i',                                    /* 155 -- table */
'i',                                    /* 156 -- tree */
'i',                                    /* 157 -- window */
'i'                                     /* 158 -- bureaucrat */
     );

     interiorptr = ObjList(front.container);
     displayptr = ObjList(interior.contents->c(VENDO_DISPLAY));
     the_message = info_messages(display.class);
     if (the_message = '-') then /* non-existant objects */
          the_message = 'This object does not exist.';
     else if (the_message = 'b') then
          the_message = book_vendo_info(displayptr);
     else if (the_message = 'd') then
          the_message = drugs_vendo_info(displayptr);
     else if (the_message = 'm') then
          the_message = magic_vendo_info(displayptr);
     else if (the_message = 'k') then
          the_message = key_vendo_info(displayptr);
     else if (the_message = 's') then
          the_message = sensor_vendo_info(displayptr);
     else if (the_message = 'i') then do; /* impossible to get messages */
          call trace_msg('Impossible vendo help request, class '||ltrim(front.cl
ass));
          the_message = 'Hey!  This thing shouldn''t be in a VenDroid!';
     end;
     call r_msg_s('VENDO: DO displays next selection.  PUT tokens here to purcha
se item on display.');
     call object_say(front.noid, the_message || '  $' || ltrim(front.item_price)
);
end vendo_HELP;

vendo_front_HELP: procedure;
     call vendo_HELP(selfptr);
end vendo_front_HELP;

%cvideo#d010>lucas>microcosm>Structs>struct_pawn_machine.incl.pl1  88-02-29 21:

/*
 *   struct_pawn_machine.incl.pl1
 *
 *   Struct stub for pawn_machine instance descriptor.
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   6-October-1986
 *
 */
,    2    common_head         like instance_head,
     2    contents            pointer,
     2    class_specific      ,
           3   open_flags     binary(15),
           3   key_hi         binary(15),
           3   key_lo         binary(15);

%cvideo#d010>lucas>microcosm>Classes>class_pawn_machine.pl1  88-02-29 21:26:30

/*
 *   class_pawn_machine.pl1
 *
 *   Behavior module for object class pawn_machine.
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   6-October-1986
 */

%replace PAWN_MACHINE_CAPACITY by 1;

%include ’microcosm.incl.pl1’;
%include ’defs_helper.incl.pl1’;
%include ’defs_action.incl.pl1’;

initialize_class_pawn_machine: procedure;

     %replace PAWN_MACHINE_REQUESTS by 6;

     declare a(0:PAWN_MACHINE_REQUESTS) entry based;
     declare class_pawn_machine_actions pointer;
     declare 1 pawn_machine based %include struct_pawn_machine;

     %replace I by CLASS_PAWN_MACHINE;

     Class_Table(I).capacity = PAWN_MACHINE_CAPACITY;
     Class_Table(I).max_requests = PAWN_MACHINE_REQUESTS;
     Class_Table(I).alloc_size = size(pawn_machine);
     Class_Table(I).pc_state_bytes = 3;
     Class_Table(I).known = true;
     Class_Table(I).opaque_container = true;
     Class_Table(I).filler = false;

     allocate a set(class_pawn_machine_actions);
     Class_Table(I).actions = class_pawn_machine_actions;

     Class_Table(I).actions->a(HELP)  = generic_HELP;      /* 0 */
     Class_Table(I).actions->a(1)     = illegal;           /* 1 */
     Class_Table(I).actions->a(2)     = illegal;           /* 2 */
     Class_Table(I).actions->a(3)     = illegal;           /* 3 */
     Class_Table(I).actions->a(4)     = illegal;           /* 4 */
     Class_Table(I).actions->a(5)     = illegal;           /* 5 */
     Class_Table(I).actions->a(MUNCH) = pawn_machine_MUNCH;/* 6 */
```

```
end initialize_class_pawn_machine;

pawn_machine_MUNCH: procedure;
     declare 1 self based(selfptr) %include struct_pawn_machine;

     if (adjacent(selfptr) & self.contents->c(0) ^= NULL) then do;
          if (pay_to(avatarptr, item_value(ObjList(self.contents->c(0))))) then
do;
               call n_msg_1(selfptr, MUNCH$, avatar.noid);
               call n_msg_1(null(), GOAWAY_$, self.contents->c(0));
               call destroy_contents(selfptr);
               call r_msg_1(TRUE);
               return;
          end;
          call r_msg_1(BOING_FAILURE);
          return;
     end;
     call r_msg_1(FALSE);
end pawn_machine_MUNCH;

%cvideo#d010>quantum>stratus>include_library>microcosm>instance_head.def.incl.p

/*
 *   instance_head.def.incl.pl1
 *
 *   The common header shared by ALL object instance descriptors.
 *
 *   Chip Morningstar
 *   Lucasfilm Ltd.
 *   9-April-1986
 *
 */
declare 1 instance_head       based,
          2    avatarslot     binary(15),
          2    obj_id         binary(31),
          2    noid           binary(15),
          2    class          binary(15),
          2    style          binary(15),
          2    x              binary(15),
          2    y              binary(15),
          2    position       binary(15),
          2    orientation    binary(15),
          2    gr_state       binary(15),
          2    container      binary(15),
          2    gr_width       binary(15),
          2    gen_flags(32)  bit(1);
```