

SCAPE FROM IRONCRYPT



Sara Salmón and Alfredo Mpeso



```

def get_next_room_of_door(door, current_room):
    """
    From object_relations, find the two rooms connected to the given door.
    Return the room that is not the current_room.
    """
    connected_rooms = object_relations[door["name"]]
    for room in connected_rooms:
        if(not current_room == room):
            return room

def examine_item(item_name):
    """
    Examine an item which can be a door or furniture.
    First make sure the intended item belongs to the current room.
    Then check if the item is a door. Tell player if key hasn't been
    collected yet. Otherwise ask player if they want to go to the next
    room. If the item is not a door, then check if it contains keys.
    Collect the key if found and update the game state. At the end,
    play either the current or the next room depending on the game state
    to keep playing.
    """
    current_room = game_state["current_room"]
    next_room = ""
    output = None

    for item in object_relations[current_room["name"]]:
        if(item["name"] == item_name):
            output = "Has examinado " + item_name + ". "
            if(item["type"] == "door"):
                have_key = False
                for key in game_state["keys_collected"]:
                    if(key["target"] == item):
                        have_key = True
                if(have_key):
                    output += "Usa conmina para ir al siguiente habitació"

```

Project overview

GAME'S STRUCTURE
 FUNCTIONS IMPORTED
 FEATURES ADDED

```

for spec in grid:
    queue = new_queue(hwm, True, pack)
    self._queues[spec.model.hid].append(queue)
    self._workers.append(
        Process(self._estimate, self._load_estimator, spec,
                self._scorer, self._to_seq, self._seq_attr,
                self._seq_delta, self._update_info, queue, fstore,
                log_every, store_every).start())
# Pass to another process to create a fit function.
self._ffit = partial(self._ffit, self._queues, pack)
self._store = fstore and fstore()
join(self):
    for worker in self._workers:
        worker.join()
resume_from(self):
try:
    from_time = min(spec.from_time for spec in self._grid
                    if spec.from_time is not None)
except ValueError: # All from_times are None: min with empty sequence
    from_time = None
if self._seq_attr == 'time':
    return from_time, None
sinfos = self._store.get_infos(self._grid)
to_sids = [info.to_sid for _, info in sinfos]
from_sid = None if len(to_sids) < len(self._grid) else min(to_sids)
return from_time, from_sid

best(self, rank=lambda info: info.r2):
    sinfos = list(self._store.get_infos(self._grid))
    ranks = (rank(info) for _, info in sinfos)
    ranked = [(rank, sinfo) for rank, sinfo in zip(ranks, sinfos)
              if rank is not None]
    if not ranked:

```

```

# Laboratorio de Xisca

xisca_lab = {
    "name": "xisca lab",
    "type": "room",
}

xisca = {
    "name": "xisca",
    "type": "furniture",
}

sistema_seguridad = {
    "name": "sistema de seguridad",
    "type": "door",
}

contraseña = {
    "name": "contraseña",
    "type": "key",
    "target": sistema_seguridad,
}

puerta_hall = {
    "name": "puerta hall",
    "type": "door",
}

```

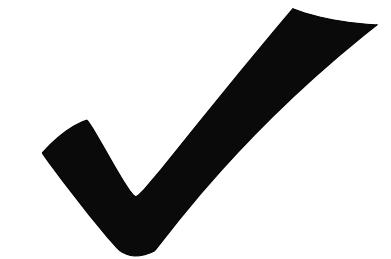
TECHNICAL CHALLENGE

BIG MISTAKE



Key-value changes
**Importance of
dictionaries**

HUGE LESSON



**Organization
Architecture
Hierarchy
Indentation**

РЕСАНЕ АСТОМ

ALL IN, WIN AN AND ESCAPE, OR LOSE ⚡ DIE,



RDIDA

LET'S PLAY HACKOUT !

PLAYER XISCA

ALL IN.
WIRE MAN
DECAPE
ESCAPE

LOSE AND DIE

SCAPE FROM IRONCRYPT

THANK YOU!

SARA SALMÓN
AND ALFREDO MPESO

