

C2 Level Design -Mechanics and Layout

C2 Level Design -Mechanics and Layout

This isn't easy, a common theme of chemistry seems to be forcing me to use visually bigger puzzles to get simple results. Like the B3 level, C2 features a lot of speech from the payer to describe what is going on.

The C2 level takes its themes from Topic 4: Groups in the periodic table:

- Broken circuit (Metallic bonding and transition metals).
- The fire room (Noble gasses).
- Water drop (Alkali Metals)

Name: C2

Class: Stage

States: Incomplete, Complete

Can transfer states: no

Starting state: Incomplete

Algorithm(s):

```
If [C.Teleporter]: {on},  
And [C2]: not {Incomplete},  
enter {Incomplete}.  
While: {Incomplete},  
[C2] will be assigned to main instance.  
If [Battery]: {Win},  
enter {complete}.  
While: {complete},  
[C2] will be deleted.
```

Name: C2.Teleporter

Class: Stage object

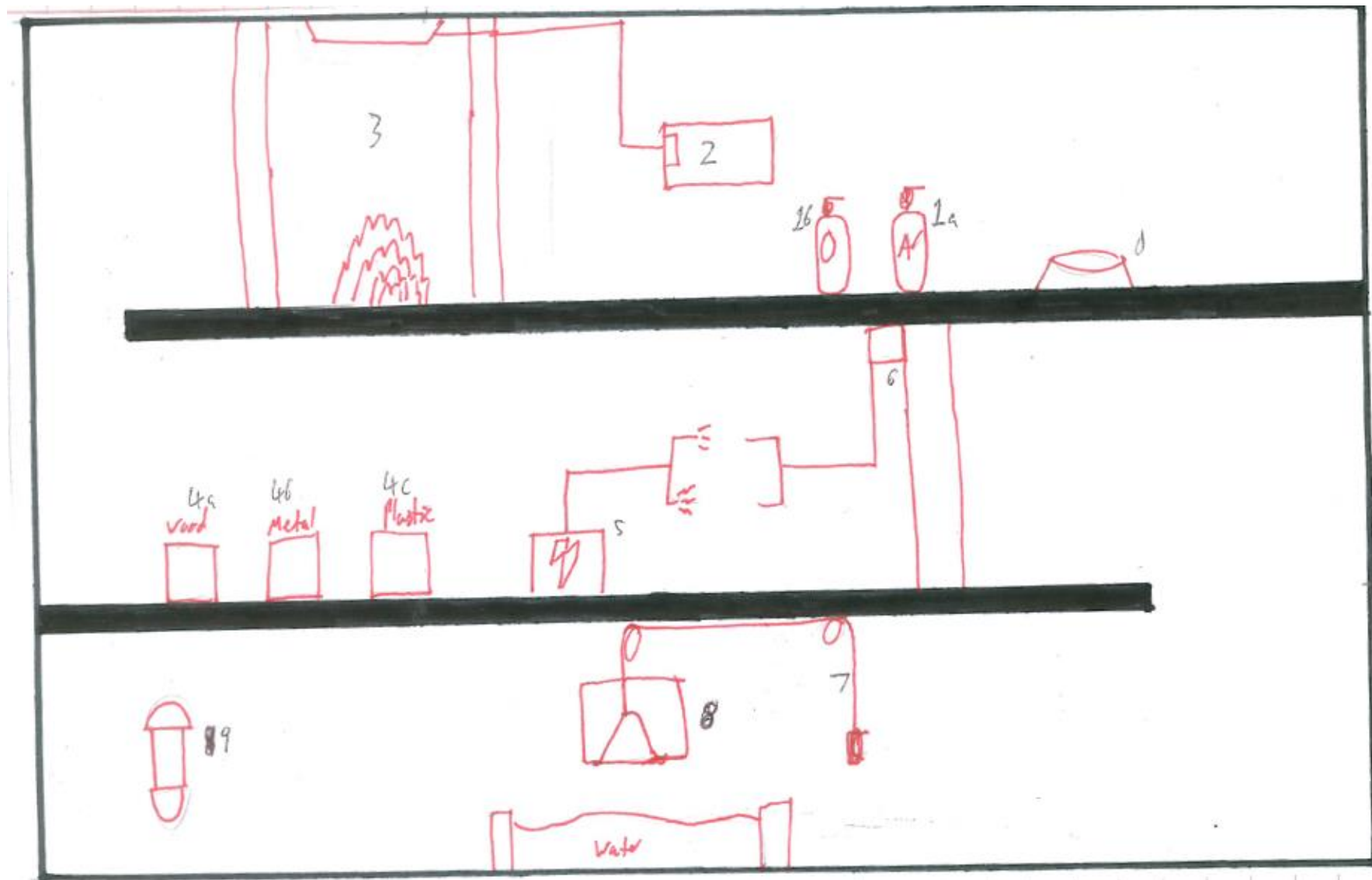
States: Off, on

Can transfer states: no

Starting state: Off

Algorithm(s):

```
animation: teleporter  
If [C2]: {Off};  
Load [Player] in +0 spaces.
```



1. *Gas canisters*: Object 1 consists of 2 gasses, the player can interact with any of the four to enter a hold state (states: O and Ar). The player can transfer either of the two states back by interacting with object 1.

Name: Gas canisters

Class: Hold Object

States: Idle, O, Ar

Can transfer states: Yes (can hold multiple states)

Starting state: Idle

Algorithm(s):

While: {Idle}; animation: Idle.

1a

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {O} to [Player].

While: {O}; animation: O.

While: {O},

If {interacting} occurs within +-1 spaces,

Transfer {O} from [Player].

Enter {Idle}

While: {Idle}; animation: Idle.

1b

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {Ar} to [Player].

While: {Ar}; animation: Ar.

While: {Ar},

If {interacting} occurs within +-1 spaces,

Transfer {Ar} from [Player].

Enter {Idle}

2. *Gas feed*: Starts in an off state, the player can transfer states on to object 2 (states O and Ar). The player can transfer either of the two states back by interacting with object 2 again.

Name: Gas feed

Class: Hold Object

States: Idle, O, Ar

Can transfer states: Yes (can hold multiple states)

Starting state: Idle

Algorithm(s):

While: {Idle}; animation: Idle.

2a

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {O} from [Player].

While: {O}; animation: O.

While: {O},

If {interacting} occurs within +-1 spaces,

Transfer {O} to [Player].

Enter {Idle}

While: {Idle}; animation: Idle.

2b

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {Ar} from [Player].

While: {Ar}; animation: Ar.

While: {Ar},

If {interacting} occurs within +-1 spaces,

Transfer {Ar} to [Player].

Enter {Idle}

3. *Fire room*: Object 3 starts in a closed state. The player can transfer states onto object 3. When in and closed state, object 3 will function as a vertical platform (collision box will be larger than its visuals). If the player tries to interact with object 3 while in an O state, then it will say (Last time I added oxygen to a fire it burned faster!). When in an Ar state object 3 will change its interactions and animations to allow the player to pass.

Name: Fire room

Class: Interactive Object

States: Idle, O, Ar, On

Can transfer states: Yes

Starting state: Off

Algorithm(s):

While: {Idle}; animation: Idle,

Mimic {[platform]}.

If [Fire room]: {Ar},

enter {On}.

While: {On}; animation: On,

Mimic {[none]}.

3a

While: {Idle},

If {interacting} occurs within +-1 spaces,

While [Gas feed]: {O},

[Player] speak (Last time I added oxygen to a fire it burned faster!).

3b

While: {Idle},

If {interacting} occurs within +-1 spaces,

While [Gas feed]: {Ar},

Enter {Ar}.

4. *Blocks*: Object 4 consists of three blocks the player can interact with any of the three to enter a hold state (states: wood, metal and plastic (they will be textured as such)). The player can transfer any of the three states back by interacting with object 4.

Name: Blocks

Class: Hold Object

States: Idle, wood, metal, plastic

Can transfer states: Yes (can hold multiple states)

Starting state: Idle

Algorithm(s):

While: {Idle}; animation: Idle.

4a

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {wood} to [Player].

While: {wood}; animation: wood.

While: {wood},

If {interacting} occurs within +-1 spaces,

Transfer {wood} from [Player],

Enter {Idle}.

While: {Idle}; animation: Idle.

4b

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {metal} to [Player].

While: {metal}; animation: metal.

While: {metal},

If {interacting} occurs within +-1 spaces,

Transfer {metal} from [Player],

Enter {Idle}.

While: {Idle}; animation: Idle.

4c

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {plastic} to [Player].

While: {plastic}; animation: plastic.

While: {plastic},

If {interacting} occurs within +-1 spaces,

Transfer {plastic} from [Player],

Enter {Idle}.

While: {Idle}; animation: Idle.

5. *Open circuit*: Object 5 starts in an open state. The player can transfer states onto object 6. If the player tries to interact with object 6 while in a wood or plastic state, then it will say (Nothing is happening). When in a metal state object 5 will change on a closed state.

Name: Open Circuit

Class: Interactive Object

States: Idle, wood, metal, plastic, closed

Can transfer states: Yes

Starting state: Off

Algorithm(s):

While: {Idle}; animation: Idle,

If [blocks]: {metal},

enter {closed}.

5a

While: {Idle},

If {interacting} occurs within +-1 spaces,

While [blocks]: {wood},

[Player] speak (Nothing is happening).

5b

While: {Idle},

If {interacting} occurs within +-1 spaces,

While [blocks]: {plastic},

[Player] speak (Nothing is happening).

5c

While: {Idle},

If {interacting} occurs within +-1 spaces,

While [blocks]: {metal}.

6. *Gate*: When in an off state, it will function as a vertical platform. If object 5 enters a closed state, then object 6 will enter an on state, removing all its interactions and changing its visuals.

Name: Gate

Class: Reactive Object

States: Off, On

Can transfer states: no

Starting state: Off

Algorithm(s):

While: {Off}; animation: Off,

Mimic {[platform]}.

If [Open circuit]: {closed},

enter {On}.

While: {On}; animation: none.

Mimic {[none]}

7. *Pulley*: Starts in an off state. If the player interacts with object 7, it will enter an on state.

Name: Pulley

Class: Interactive Object

States: Off, On

Can transfer states: no

Starting state: Off

Algorithm(s):

While: {Off}; animation: Off

If {interacting} occurs within +-1 spaces,
enter {On}.

While: {On}; animation: On.

8. *Alkali drop*: When in a normal state, it will function as a vertical platform. If object 8 enters an on state, then object 8 will enter a broken state, removing all its interactions and changing its visuals.

Name: Alkali drop

Class: Reactive Object

States: On, Off

Can transfer states: no

Starting state: On

Algorithm(s):

While: {On}; animation: On,

Mimic {[platform]}.

If [Pulley]: {On},
enter {Off}.

While: {Off}; animation: broken.

Mimic {[none]}

9. *Battery*: when the player touches object 9 the level instance enters a win state. This will end and lock the instance while loading up the neutral area in a C2 clear state (This will change certain visuals of the neutral area).

Name: C2 Battery

Class: Stage Object

States: On, Off, Win

Can transfer states: no

Starting state: On

Algorithm(s):

While: {On}; animation: On.

If [Player] enters +-0 spaces,
enter {Off}.

While: {Off}; animation: Off.

If {Off},

Wait 5seconds

enter {Win}.