C3 Level Design -Mechanics and Layout

C3 Level Design -Mechanics and Layout

Unlike all the other sciences in every other section, C3 consists of many smaller topics. This meant that if I was to select just one topic for the C3 level like I have been doing for the rest of the stages; the user would be doing almost the same thing three times.

To avoid this, I just blended different topics into one level, making C3 the most varied level:

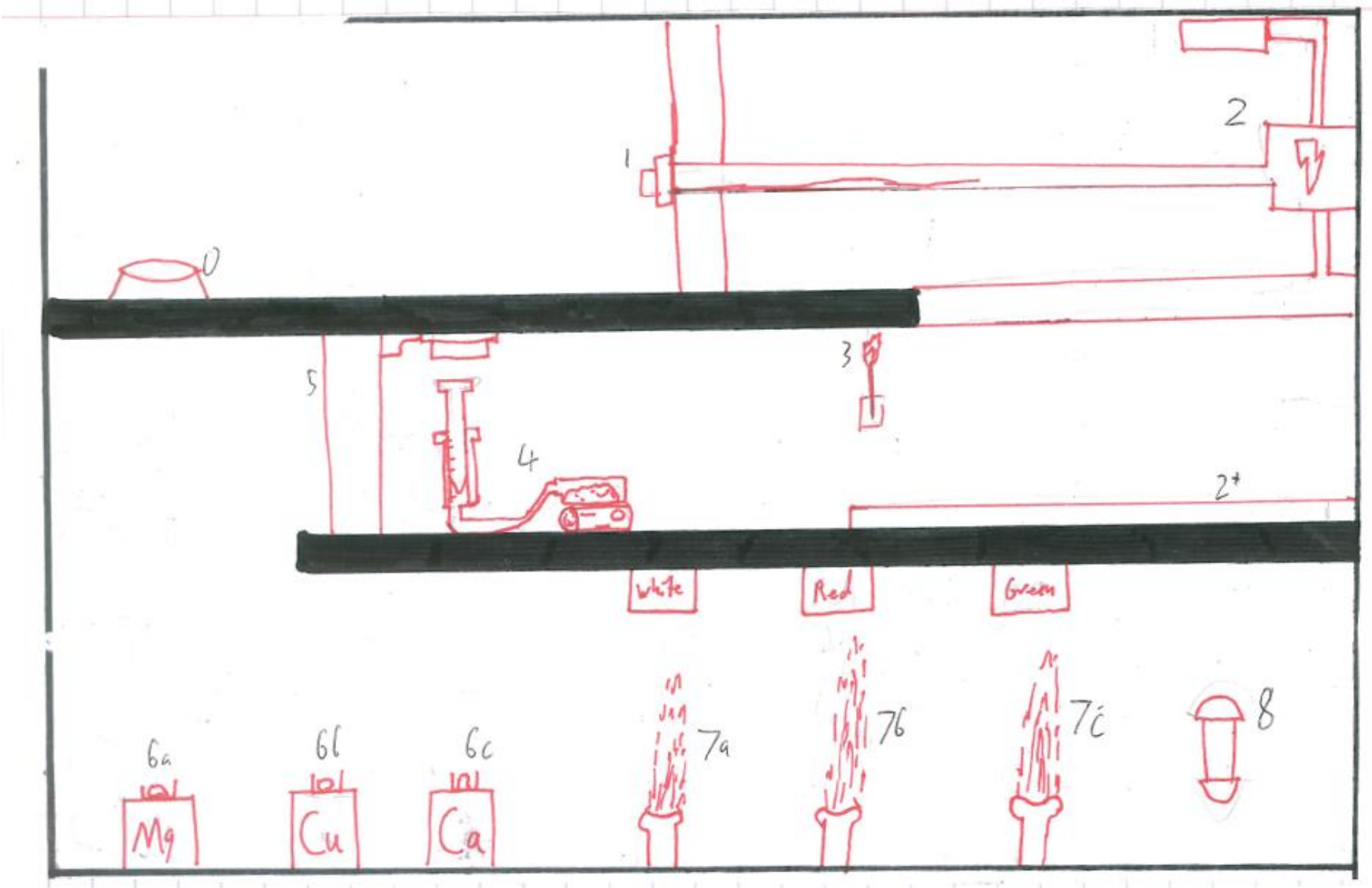
- Electroplated lock (Topic 3: Uses of electrolysis)

- High button (Topic 4: Molar volumes of gas)

- Rainbow flames (Topic 1: flame testing)

Name: C3
Class: Stage
States: Incomplete, Complete
Can transfer states: no
Starting state: Incomplete
Algorithm(s):
If [C.Teleporter]: {on},
And [C1]: not {Incomplete},
And [C2]: not {Incomplete}
enter {Incomplete}.
While: {Incomplete},
[C3] will be assigned to main instance.
If [Battery]: {Win},
enter {complete}.
While: {complete},
[C3] will be deleted.

Name: C3.Teleporter
Class: Stage object
States: Off, on
Can transfer states: no
Starting state: Off
Algorithm(s):
animation: teleporter
If [C3]: {Off};
Load [Player] in +0 spaces


0
1
2
3
4
5
2*
White
Red
Green
6a
6b
6c
Mg
Cu
Ca
7a
7b
7c
8

1. *Button*: Starts in an off state. If the player interacts with object 1, it will enter an on state.

Name: Button
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.

2. *Electroplating station*: Starts in an off state. When in an off state, it will function as a vertical platform. If object 1 enters an on state, then object 8 will enter an on state, removing all its interactions and changing its visuals.

Name: Electroplating station
Class: Reactive Object
States: On, Off
Can transfer states: no
Starting state: On
Algorithm(s):
While: {On}; animation: On,
Mimic {[platform]}.
If [Button]: {On},
enter {Off}.
While: {Off}; animation: Off.
Mimic {[none]}

3. Torch: When the player interacts with object 3 they will enter a hold state (state: fire). The player can transfer this state back by interacting with object 3.

Name: Torch
Class: Hold Object
States: Idle, fire
Can transfer states: Yes (can hold multiple states)
Starting state: Idle
Algorithm(s):
While: {Idle}; animation: Idle.
If {interacting} occurs within +-1 spaces,
Transfer {fire} to [Player].
While: {fire}; animation: none.
While: {fire},
If {interacting} occurs within +-1 spaces,
Transfer {fire} from [Player],

Enter {Idle}.
While: {Idle}; animation: Idle.

4. *Syringe:* Object 4 starts in a closed state. The player can transfer states onto object 4. When in fire state object 4 will change its interactions.

Name: Syringe

Class: Interactive Object
States: Idle, fire
Can transfer states: Yes
Starting state: Idle
Algorithm(s):
While: {Idle}; animation: Idle,
If {interacting} occurs within +-1 spaces,
While [Player]: {fire},
Transfer {fire}.
While: {fire}; animation: fire

5. *Gate*: When in an off state, it will function as a vertical platform. If object 4 enters a fire state, then object 6 will enter an on state, removing all its interactions and changing its visuals.

Name: Gate

Class: Reactive Object
States: On, Off
Can transfer states: no
Starting state: On
Algorithm(s):
While: {Off}; animation: Off,
Mimic {[platform]}.
If [Syringe]: {fire},
enter {On}.
While: {On}; animation: On.
Mimic {[none]}

6. *Chemicals*: Object 6 consists of three chemicals, the player can interact with any of the three to enter a hold state (states: mg, cu and ca). The player can transfer any of the three states back by interacting with object 6.

Name: Chemicals

Class: Hold Object
States: Idle, mg, cu, ca
Can transfer states: Yes (can hold multiple states)
Starting state: Idle
Algorithm(s):
While: {Idle}; animation: Idle.

6a
While: {Idle},
If {interacting} occurs within +-1 spaces,
Transfer red} to [Player].
While: {mg}; animation: mg.
While: {mg},
If {interacting} occurs within +-1 spaces,
Transfer {mg} from [Player].
Enter {Idle}
While: {Idle}; animation: Idle.

6b
While: {Idle},
If {interacting} occurs within +-1 spaces,
Transfer {cu} to [Player].
While: {cu}; animation: cu.
While: {cu},
If {interacting} occurs within +-1 spaces,
Transfer {cu} from [Player].
Enter {Idle}
While: {Idle}; animation: Idle.

6c
While: {Idle},
If {interacting} occurs within +-1 spaces,
Transfer {ca} to [Player].
While: {ca}; animation: ca.
While: {ca},
If {interacting} occurs within +-1 spaces,
Transfer {ca} from [Player].
Enter {Idle}
While: {Idle}; animation: Idle.

7. Flames: Object 7 actually consists of 3 separate elements. Each starts in a locked state. When in a locked state, it will function as a vertical platform. The player can transfer one state onto each flame (7a, 7b and 7c). When a flame enters a mg, ca or cu state, it will change animation to turn white, red or green (respectively). When object 7a enters a mg state, it will change to an unlocked state. When object 7b enters a ca, it will change to an unlocked state. When object 7c enters a cu state, it will change to an unlocked state. While in this state, object 7 changes it's interactions and its animation to allow the player to pass.

Name: Flames

Class: Interactive Object
States: Idle, mg, cu, ca, On
Can transfer states: Yes
Starting state: Off
Algorithm(s):

7a
While: {Idle}; animation: Idle
mimic {[platform]}
If {interacting} occurs within +-1 spaces,
Transfer {cu} from[Player].
While: {cu}; animation: green.
If {interacting} occurs within +-1 spaces,
Transfer {ca} from[Player].
While: {ca}; animation: green.
If {interacting} occurs within +-1 spaces,
Transfer {mg} from[Player].
While: {mg}; animation: white.
If [7a]: {mg},
mimic {[none]}.

7b
While: {Idle}; animation: Idle
mimic {[platform]}
If {interacting} occurs within +-1 spaces,
Transfer {cu} from[Player].
While: {cu}; animation: green.
If {interacting} occurs within +-1 spaces,
Transfer {ca} from[Player].
While: {ca}; animation: green.
If {interacting} occurs within +-1 spaces,
Transfer {mg} from[Player].
While: {mg}; animation: white.
If [7b]: {ca},
mimic {[none]}.

7c
While: {Idle}; animation: Idle
mimic {[platform]}
If {interacting} occurs within +-1 spaces,

Transfer {cu} from[Player].
While: {cu}; animation: green.
If {interacting} occurs within +-1 spaces,
Transfer {ca} from[Player].
While: {ca}; animation: green.
If {interacting} occurs within +-1 spaces,
Transfer {mg} from[Player].
While: {mg}; animation: white.
If [7c]: {cu},
mimic {[none]}.

8. *Battery*: when the player touches object 8 the level instance enters a win state. This will end and lock the instance while loading up the neutral area in a C3 clear state (This will change certain visuals of the neutral area).

Name: C3 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}.