C1 Level Design -Mechanics and Layout

Chemistry is a pretty mathematics heavy subject, mostly focused with the idea of dealing with chemicals. This does not help when trying to make chemistry obstacles as everything is either mathematics or text heavy. It took a lot longer than planned to come up with level designs

The first level in chemistry focuses on C1- Topic: Fuels.

- Hydrocarbon wall (cracking).

-Fan a fire (incomplete and complete combustion).

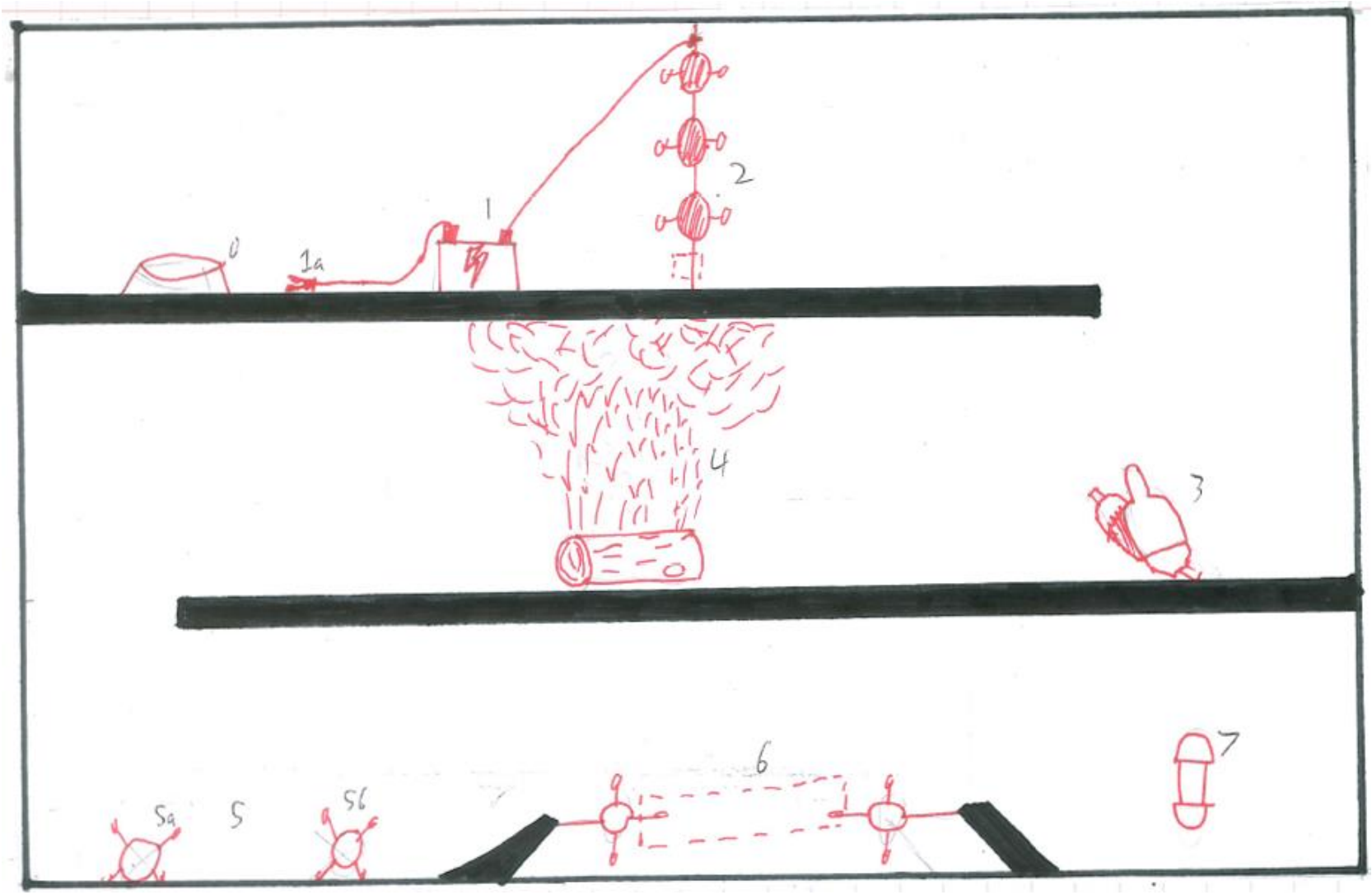
- Alkane wall (polymerization)

Name: C1

Class: Stage
States: Incomplete, Complete
Can transfer states: no
Starting state: Incomplete
Algorithm(s):
If [C.Teleporter]: {on},
enter {Incomplete}.
While: {Incomplete},
[C1] will be assigned to main instance.
If [Battery]: {Win},
enter {complete}.
While: {complete},
[C1] will be deleted.

Name: C1.Teleporter

Class: Stage object
States: Off, on
Can transfer states: no
Starting state: Off
Algorithm(s):
animation: teleporter
If [C1]: {Off};
Load [Player] in +0 spaces


0
1a
1
2
3
4
5
5a
5b
6
7

1. *Power box*: When the player interacts with object 1 they will enter a hold sate (state: powered). The player can transfer this state back by interacting with object 1.

Name: Power Box

Class: Hold Object
States: Idle, powered
Can transfer states: Yes
Starting state: Idle
Algorithm(s):
While: {Idle}; animation: Idle.
If {interacting} occurs within +-1 spaces,
enter {powered}.
While: {powered}; animation: none,
Transfer {fuel} to player.
While: {powered},
If {interacting} occurs within +-1 spaces,
Enter {Idle},
Transfer {powered} from player.

2. *Hydrocarbon wall*: Starts in a locked state. The player can transfer states onto object 2. When object 2 enters a powered state, it will enter an unlocked state. While in this state, object 2 will change its interactions and animations to allow the player to pass.

Name: Hydrocarbon wall

Class: Interactive Object
States: Locked, Unlocked
Can transfer states: Yes
Starting state: Locked
Algorithm(s):
While: {Locked}; animation: Locked,
Mimic {[platform]}.
If {interacting} occurs within +-1 spaces,
While [Player]: {powered},
Transfer {powered}.
If: {powered}, enter {Unlocked}.
While: {Unlocked}; animation: powered
Wait 4 seconds; animation: broken.

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

Name: Bellows

Class: Hold Object
States: Idle, bellow
Can transfer states: Yes
Starting state: Idle
Algorithm(s):
While: {Idle}; animation: Idle.
If {interacting} occurs within +-1 spaces,
enter {bellow}.
While: {bellow}; animation: none,
Transfer {bellow} to player.
While: {bellow},
If {interacting} occurs within +-1 spaces,
Enter {Idle},
Transfer {bellow} from player.

4. *Bonfire*: Object 4 starts in a closed state. The player can transfer states onto object 4. When in and closed state, object 4 will function as a vertical platform (Collison box will be larger than its visuals). If the player touches object 4, then it will say (There is a lot of black smoke, maybe it's not getting enough air?). When in a bellow state object 4 will change its interactions and animations to allow the player to pass.

Name: Bonfire

Class: Interactive Object
States: On, bellow, Off
Can transfer states: Yes
Starting state: Locked
Algorithm(s):
While: {On}; animation: One
Mimic {[platform]}.
If {walking} or {jumping} occurs within +-1 spaces,
[Player] speak (There is a lot of black smoke, maybe it's not getting enough air?).
If {interacting} occurs within +-1 spaces,
While [Player]: {bellow},
Transfer {bellow}.
If: {bellow}, enter {Off}.
While: {Off}; animation: Bellows,
Wait 5 seconds, animation: ash,
Mimic {[none]}. none

5. *Hydrocarbons*: Object 5 consists of two hydrocarbons; the player can interact with either of the two to enter a hold state (state: carbon). The player can transfer the state back by interacting with object 5.

Name: Hydrocarbon

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

5a
While: {Idle},
If {interacting} occurs within +-1 spaces,
Transfer {carbon} to [Player].
While: {carbon}; animation: carbon
While: {carbon},
If {interacting} occurs within +-1 spaces,
Transfer {carbon} from [Player].
Enter {Idle}
While: {Idle}; animation: Idle.

5b
While: {Idle},
If {interacting} occurs within +-1 spaces,
Transfer {carbon} to [Player].
While: {carbon}; animation: carbon
While: {carbon},
If {interacting} occurs within +-1 spaces,
Transfer {carbon} from [Player].
Enter {Idle}
While: {Idle}; animation: Idle.

6. *Polymer wall*: Starts in an off state, the player can transfer states into the two parts of object 6. When object 6 has two carbon states stacked then it will enter an on state. While in this state, object 6changes it's interactions and its animation to behave like a platform.

Name: Polymer wall

Class: Interactive Object
States: Idle, carbon
Can transfer states: Yes
Starting state: Off
Algorithm(s):
While: {Off}; animation: Off,
Mimic {[none]}.
If {walking} occurs within +-1 spaces,
Cancel {walking}.
If {jumping} occurs within +-^1 spaces,
Cancel {Jumping}.

6a
While: {Idle},
If {interacting} occurs within +-1 spaces,
Transfer {carbon} from[Player].
While: {carbon}; animation: carbon1.
Mimic {[platform]}.

6b
While: {Idle},
If {interacting} occurs within +-1 spaces,
Transfer {carbon} from[Player].
While: {carbon}; animation: carbon2.
Mimic {[platform]}.

7. *Battery*: when the player touches object 7 the level instance enters a win state. This will end and lock the instance while loading up the neutral area in a C1 clear state (This will change certain visuals of the neutral area).

Name: C1 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}.