P3 Level Design -Mechanics and Layout

As the final stage in physics, this level may not boast any more puzzles than the previous levels, but it is definitely bigger than the last two and features some of the most imaginative puzzles yet.

-The smoke alarm (alpha radiation)

-The waterfall (beta radiation)

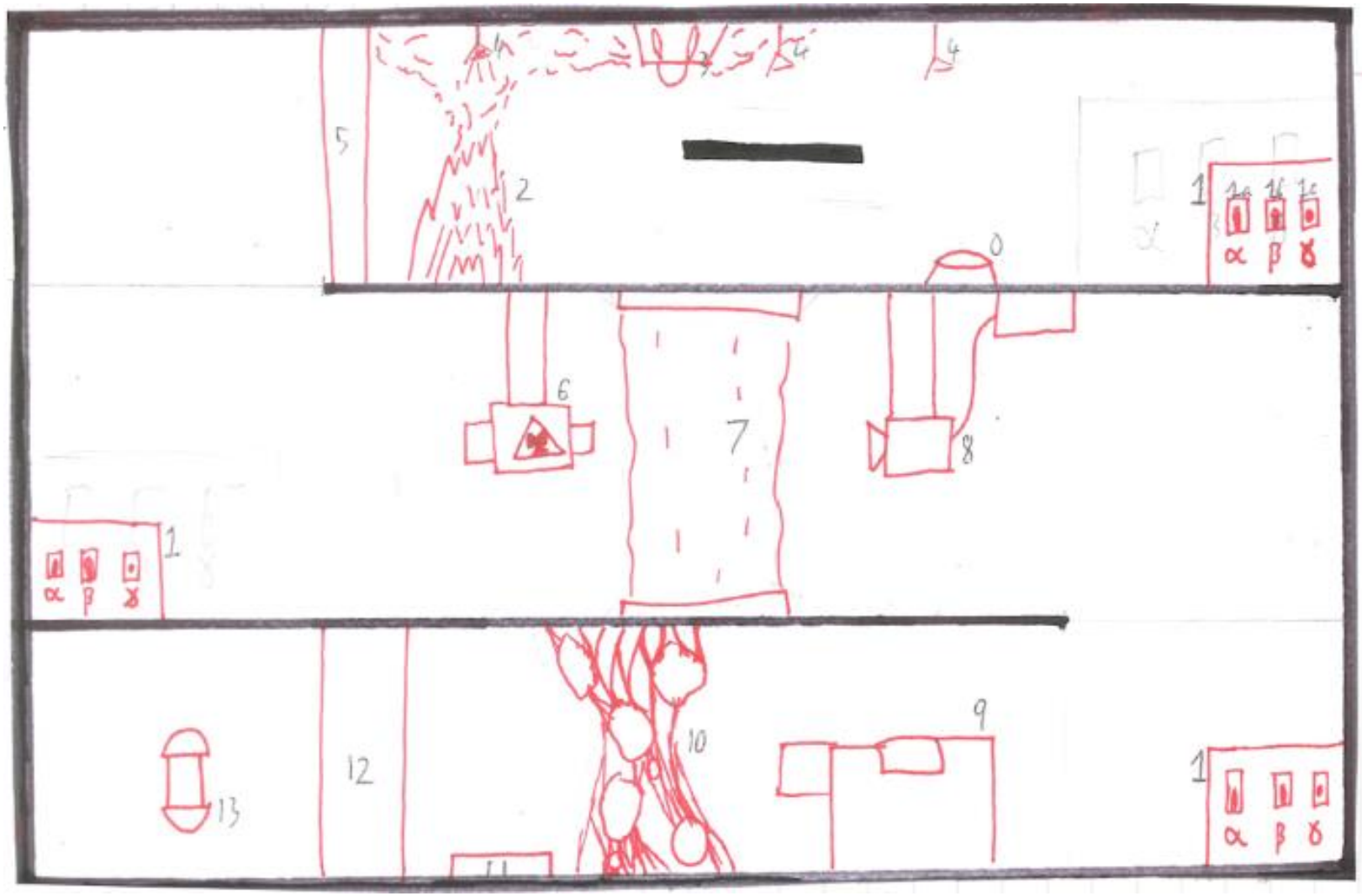
-The cancer (gamma radiation)

Name: P3

Class: Stage
States: Incomplete, Complete
Can transfer states: no
Starting state: Incomplete
Algorithm(s):
If [P.Teleporter]: {on},
And [P1]: not {Incomplete},
And [P2]: not {Incomplete}
enter {Incomplete}.
While: {Incomplete},
[P3] will be assigned to main instance.
If [Battery]: {Win},
enter {complete}.
While: {complete},
[P3] will be deleted.

Name: P3.Teleporter

Class: Stage object
States: Off, on
Can transfer states: no
Starting state: Off
Algorithm(s):
animation: teleporter
If [P3]: {Off};
Load [Player] in +0 spaces


5
2
4
1
1a
1b
1c
α
β
δ
0
6
7
8
1
α
β
δ
12
13
10
9
1
α
β
δ

Layout and Mechanics

1. *Source box*: This box is spilt into three different sections. If the player interacts with 1a on the far left, the player will hold the alpha source and enter an alpha state. If the player interacts with 1b in the middle, the player will hold the beta source and enter a beta state. If the player interacts with 1c on the far right, the player will hold the gamma source and enter a gamma state. To remove these states, the player must interact again with object 1 or another object capable of holding sources. The player can only enter one state at a time.

Name: Source box

Class: Interactive Object
States: Idle, alpha, beta, gamma
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 {alpha} to [Player].
While: {alpha}; animation: alpha.
While: {alpha},
If {interacting} occurs within +-1 spaces,
Transfer {alpha} from [Player].
Enter {Idle}
While: {Idle}; animation: Idle.

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

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

2. *Fire*: When in an on state, it will function as a vertical platform (Collison box will be larger than its visuals). If the player touches object 2, then it will say (I am not fire proof! I need to stop the fire to continue.) If the stage enters a wet state, then object 2 will enter an off state, removing all its interactions and visuals.

Name: Fire

Class: Reactive Object
States: On, Off
Can transfer states: no
Starting state: On
Algorithm(s):
While: {On}; animation: On,
Mimic {[platform]}.
If {walking} or {jumping} occurs within +-1 spaces,
[Player] speak (I am not fire proof! I need to stop the fire to continue.) .
If [Sprinklers]: {on}
enter {off}.
While: {off}; animation: none.
Mimic {[none]}

3. *Smoke alarm*: Starts in an off state, if the player interacts with 3 while in a state, it will transfer its state to object 3 (is also works the other way around). If object 3 enters an alpha state, then it will change to an on state.

Name: Smoke alarm

Class: Interactive Object
States: Off, alpha, beta, gamma, On
Can transfer states: Yes
Starting state: Off
Algorithm(s):
While: {Off}; animation: Off,
If {interacting} occurs within +-1 spaces,
While [Player]: {beta},
Transfer {beta} from [Player].
While: {beta}; animation: beta.
While: {beta},
If {interacting} occurs within +-1 spaces,
Transfer {beta} to[Player].
Enter {Off}.
While [Player]: {gamma},
Transfer {gamma} from [Player].
While: {gamma}; animation: gamma.
While: {gamma},
If {interacting} occurs within +-1 spaces,
Transfer {gamma} to[Player].
Enter {Off}.
If {interacting} occurs within +-1 spaces,
While [Player]: {alpha},

Transfer {alpha} from [Player].
While: {alpha}; animation: alpha.
If: {alpha}, enter {On}.

4. *Sprinklers*: Starts in an off state. If object 3 is in an on state, then object 4 enters an on state. When in an on state the stage enters a wet state, this will change object 4's animation for 10 seconds.

Name: Sprinklers

Class: Reactive Object
States: Off, On
Can transfer states: no
Starting state: Off
Algorithm(s):
While: {Off}; animation: Off
If [Smoke alarm]: {On},
enter {On}.
While: {On}; animation: On,
Wait 10 seconds,
animation: Off

5. *Fire door*: Starts in a closed state. If the stage enters a wet state than five seconds later, object 5 will enter an open state. When in an open state, object 5 will change visuals.

Name: Fire door

Class: Reactive Object
States: Closed, Open
Can transfer states: no
Starting state: Unpowered
Algorithm(s):
While: {Closed}; animation: Closed,
Mimic {[platform]}.
If [Sprinklers]: {powered},
enter {Open}.
While: {Open}; animation: none.
Mimic {[none]}

6. *Beta gun*: Starts in an off state, if the player interacts with object 6 while in a state, it will transfer its state to object 6 (is also works the other way around). If object 6 enters a beta state, then it will change to an on state.

Name: Beta gun

Class: Interactive Object
States: Off, alpha, beta, gamma, On
Can transfer states: Yes
Starting state: Off

Algorithm(s):
While: {Off}; animation: Off,
If {interacting} occurs within +-1 spaces,
While [Player]: {alpha},
Transfer {alpha} from [Player].
While: {alpha}; animation: beta.
While: {alpha},
If {interacting} occurs within +-1 spaces,
Transfer {alpha} to[Player].
Enter {Off}.
While [Player]: {gamma},
Transfer {gamma} from [Player].
While: {gamma}; animation: gamma.
While: {gamma},
If {interacting} occurs within +-1 spaces,
Transfer {gamma} to[Player].
Enter {Off}.
If {interacting} occurs within +-1 spaces,
While [Player]: {beta},
Transfer {beta} from [Player].
While: {beta}; animation: beta.
If: {beta}, enter {On}.

7. *Waterfall:* When in an on state, it will function as a vertical platform (the collision box will be larger than its visuals). If the player touches object 7, then it will say (The water is too strong for me to continue.) If object 8 enters an on state, object 7 will enter an off state, removing all its interactions and changing its visuals.

Name: Waterfall

Class: Reactive Object
States: On, Off
Can transfer states: no
Starting state: On
Algorithm(s):
While: {On}; animation: On,

Mimic {[platform]}.
If [Beta sensor]: {On},
enter {Off}.
While: {Off}; animation: Off.
Mimic {[none]}

8. *Beta sensor*: Starts in an off state. When object 6 is in an on state, then object 6 will change to an on state.

Name: Beta sensor

Class: Reactive Object
States: Off, On
Can transfer states: no
Starting state: Off
Algorithm(s):
While: {Off}; animation: Off
If [Beta gun]: {On},
enter {On}.

9. *Gamma gun*: Starts in an off state, if the player interacts with object 9 while in a state, it will transfer its state to object 9 (is also works the other way around). If object 9 enters a gamma state, then it will change to an on state.

Name: Gamma gun

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

While: {alpha},
If {interacting} occurs within +-1 spaces,
Transfer {alpha} to[Player].
Enter {Off}.
While [Player]: {beta},
Transfer {beta} from [Player].
While: {beta}; animation: beta.
While: {beta},
If {interacting} occurs within +-1 spaces,
Transfer {beta} to[Player].
Enter {Off}.
If {interacting} occurs within +-1 spaces,
While [Player]: {gamma},
Transfer {gamma} from [Player].
While: {gamma}; animation: gamma.
If: {gamma}, enter {On}.

10. *Cancer*: When in an on state, it will function as a vertical platform (Collison box will be larger than its visuals). If the player touches object 10, then it will say (I'm not touching this! Find a way to get rid of it.) If object 9 enters an on state, then object 10 will enter an off state, removing all its interactions and visuals.

Name: Cancer

Class: Reactive Object
States: On, Off
Can transfer states: no
Starting state: On
Algorithm(s):
While: {On}; animation: On,
Mimic {[platform]}.
If {walking} or {jumping} occurs within +-1 spaces,
[Player] speak (I'm not touching this! Find a way to get rid of it.)
If [Gamma gun]: {on}
enter {off}.
While: {off}; animation: none.

Mimic {[none]}

11. *Button*: Starts in an off state. When the player walks over object 11, it will enter an on state.

Name: Button
Class: Interactive Object
States: On, Off
Can transfer states: no
Starting state: Off [Player] 0
Algorithm(s):
While: {Off}; animation: Off,
If [Player] occurs within +-0 spaces,
enter {On}.
While: {On}; animation: On.

12. *Protective wall*: When in an of state, it will function as a vertical platform (Collison box will be larger than its visuals). When object 11 enters an on state, object 12 enters an on state, removing all its interactions and visuals.

Name: Protective Wall
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]}

13. *Battery*: When the player touches object 13 the level instance enters a win state. This will end and lock the instance while opening the neutral area in a P3 clear state (This will change certain visuals of the neutral area).

Name: P3 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}.

Notes:

- The order of the puzzles makes this area overly easy, as they are in the order of the radiations, therefore the floors will be swapped in the final product.
- While I am aware of the age of the target audience, I believe that I don't need to baby them. While the idea of a giant pulsating wall of cancer seems like nightmare fuel, I believe it is nothing that need to be desperately removed, as they will only be introduced into even more horrific tumors in the book.