

B3 Level Design -Mechanics and Layout

The biology levels have been pretty creative due to their difficult implementation into levels (I wanted to try avoid the overused setting of inside a human body). Therefore, as almost a cool down, rather than being massive or featuring complex puzzles- level B3 is more of a relaxing level that features some light humor with funny interactions.

The B3 level is based on Topic 2: Behavior

- Pavlov's friend/ Hungry dog (learned behavior)
- Busy bees (plant communication)
- The bird (Courtship and parenting)

Name: B3

Class: Stage

States: Incomplete, Complete

Can transfer states: no

Starting state: Incomplete

Algorithm(s):

If [P.Teleporter]: {on},
And [B1]: not {Incomplete},
And [B2]: not {Incomplete}
enter {Incomplete}.
While: {Incomplete},
[B3] will be assigned to main instance.
If [Battery]: {Win},
enter {complete}.
While: {complete},
[B3] will be deleted.

Name: B3.Teleporter

Class: Stage object

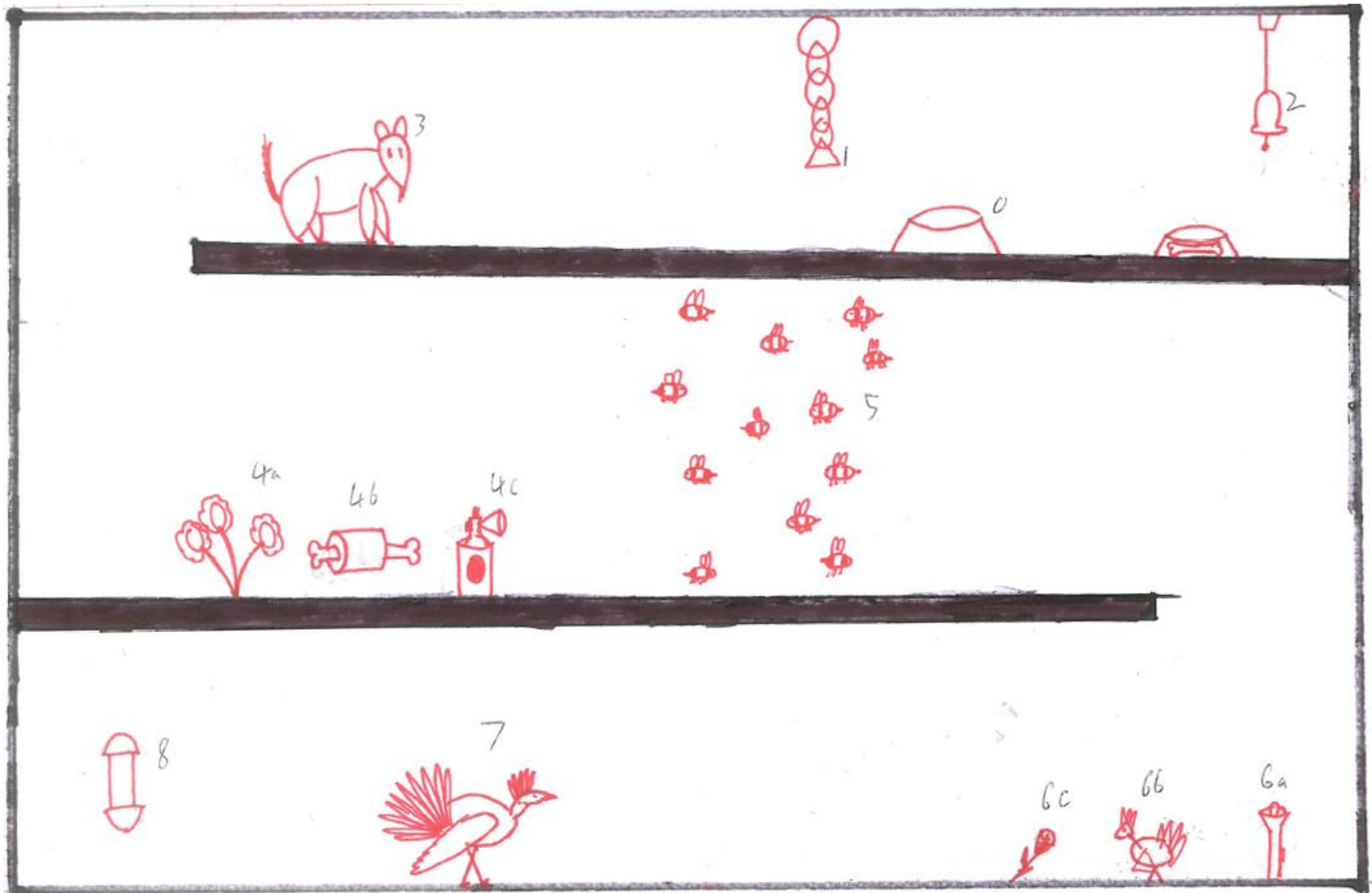
States: Off, on

Can transfer states: no

Starting state: Off

Algorithm(s):

animation: teleporter
If [B3]: {Off};
Load [Player] in +0 spaces



1. *Chain*: Object 1 starts in an off state, when the player interacts with object 1 it will enter an on state.

Name: Chain

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. *Bell and bowel*: Starts in an off state. When object 1 is in an on state, object 2 will enter an on state. When in an on state object 2 will enter an animation for 2 seconds.

Name: Bell and bowel

Class: Reactive Object

States: Off, On

Can transfer states: no

Starting state: Off

Algorithm(s):

While: {Off}; animation: Off

If [Chain]: {on},
enter {On}.

3. *Hungry Dog*: Starts in an on state. When in an on state, it will function as a vertical platform (Collision box will be larger than its visuals). If the player touches object 3, then it will say (The dog looks hungry, it may be dangerous to try pass it.) If object 2 is on, then object 3 will enter an off state, removing all its interactions and changing its visuals.

Name: Hungry Dog

Class: Reactive Object

States: On, Off

Can transfer states: no

Starting state: On

Algorithm(s):

While: {On}; animation: On,

Mimic {[platform]}.

If [Bell and bowel]: {On},
enter {Off}.

While: {Off}; animation: Off.

Mimic {[none]}

4. *Distractions*: Object 4 consists of three distractions, the player can interact with any of the three to enter a hold state (states: flower, meat and horn). The player can transfer any of the three states back by interacting with object 4 (or the space the objects started in).

Name: Distractions

Class: Hold Object

States: Idle, flower, meat, horn

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 {flower} to [Player].

While: {flower}; animation: flower.

While: {flower},

If {interacting} occurs within +-1 spaces,

Transfer {flower} from [Player].

Enter {Idle}

While: {Idle}; animation: Idle.

4b

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {meat} to [Player].

While: {meat}; animation: meat.

While: {meat},

If {interacting} occurs within +-1 spaces,

Transfer {meat} from [Player].

Enter {Idle}

While: {Idle}; animation: Idle.

4c

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {horn} to [Player].

While: {horn}; animation: horn.

While: {horn},

If {interacting} occurs within +-1 spaces,

Transfer {horn} from [Player].

Enter {Idle}

While: {Idle}; animation: Idle.

5. *Busy Bees*: Starts in an on state. When in an on state, it will function as a vertical platform (Collision box will be larger than its visuals). If the player touches object 5, then it will say (Would you try to walk past them?). If the player tries to interact with object 5 while in a meat state the player will say (Nothing's happening, maybe they're vegetarian?). If the player tries to interact with object 5 while in a horn state, the player will say (This only seems to make them angry, I should stop.) If the player interacts with object 5 while in a flower state, object 5 will enter an off state. When in an off state, object 5 changes it's animations and loses all interactions.

Name: Busy bees

Class: Interactive Object

States: Idle, flower, meat, horn, On

Can transfer states: Yes

Starting state: Idle

Algorithm(s):

While: {Off}; animation: Off,

Mimic {[platform]}.

If [busy bees]: {flower},

enter {On}.

While: {On}; animation: On.

Mimic {[none]}.

4a

While: {Idle},

If [Player]: {horn}

When {interacting} occurs within +-1 spaces,

[Player] speak (This only seems to make them angry, I should stop.).

4b

While: {Idle},

If [Player]: {meat}

When {interacting} occurs within +-1 spaces,

[Player] speak (Nothing's happening, maybe they're vegetarian?).

4c

While: {Idle},

If {interacting} occurs within +-1 spaces,

When [Player]: {meat} When

Transfer {flower} from [Player].

6. *Decoys*: Object 6 consists of three decoys, the player can interact with any of the three to enter a hold state (states: rose, toy and light). The player can transfer any of the three states back by interacting with object 6 (or the space the objects started in).

Name: Decoys

Class: Hold Object

States: Idle, rose, toy and light

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 {rose} to [Player].

While: {rose}; animation: rose.

While: {rose},

If {interacting} occurs within +-1 spaces,

Transfer {rose} from [Player].

Enter {Idle}

While: {Idle}; animation: Idle.

4b

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {toy} to [Player].

While: {toy}; animation: toy.

While: {meat},

If {interacting} occurs within +-1 spaces,

Transfer {toy} from [Player].

Enter {Idle}

While: {Idle}; animation: Idle.

4c

While: {Idle},

If {interacting} occurs within +-1 spaces,

Transfer {light} to [Player].

While: {light}; animation: light.

While: {light},

If {interacting} occurs within +-1 spaces,

Transfer {light} from [Player].

Enter {light}

While: {light}; animation: Idle.

7. *Bird*: Starts in an on state. When in an on state, it will function as a vertical platform (Collision box will be larger than its visuals). If the player touches object 7, then it will say (I think I would lose this fight). If the player tries to interact with object 7 while in a rose state the player will say (It doesn't seem interested, maybe roses aren't its thing?). If the player tries to interact with object 7 while in a light state, the player will say (It look ready to strike, this wasn't a good idea). If the player interacts with object 7 while in a bird state, object 7 will enter an off state. When in an off state, object 7 changes its animations and loses all interactions.

Name: Bird

Class: Interactive Object

States: Idle, rose, toy and light

Can transfer states: Yes

Starting state: Idle

Algorithm(s):

While: {Off}; animation: Off,

Mimic {[platform]}.

If [Bird]: {flower},

enter {On}.

While: {On}; animation: On.

Mimic {[none]}.

4a

While: {Idle},

If [Player]: {rose}

When {interacting} occurs within +-1 spaces,

[Player] speak (It doesn't seem interested, maybe roses aren't its thing?).

4b

While: {Idle},

If [Player]: {light}

When {interacting} occurs within +-1 spaces,

[Player] speak (It look ready to strike; this wasn't a good idea).

4c

While: {Idle},

If {interacting} occurs within +-1 spaces,

When [Player]: {toy} When

Transfer {toy} from [Player].

8. *Battery*: When the player touches 8 the level instance enters a win state. This will end and lock the instance while opening the neutral area in a P2 clear state (This will change certain visuals of the neutral area).

Name: B3 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:

- Unlike other levels, the player is very talkative. There is the possibility that the player will also say a line of text every time it picks up an item as well. This happens to compensate for the lack of impressive graphics to show the animals reactions.
- While this stage may seem lackluster as a subject to make a level on, I feel like it isn't really mentioned in biology. As it is a bigger part of biology as a whole, I felt that I would personally represent in this game.