

5 3 2 1

Cannonball Fun

Aaarrgh!
These
cannonballs
are heavy.

Pirate ship on the horizon!
Tap to fire the cannons!

```
1 fill('ocean')
2 far = stamp('pirateship',82,350,50)
3 near = stamp('ship',800,850,300)
4
5 function tap() {
6   ball = stamp('bomb',near.x,900,50)
7   ball.aim(x,y)
8   ball.move(UP,1000,2000)
9   ball.size(0,1600)
10 }
11
12 function loop() {
13   far.move(RIGHT,6)
14   far.wrap()
15   near.move(LEFT,10)
16   near.wrap()
17   if (far.hits('bomb')) {
18     far.explode()
19   }
20 }
```

Wrap it up.

Aim your spyglass at line 16, matey! When the center of the boat **near** sails off the edge of the screen, the **.wrap()** command automatically moves it to the opposite side. The **.wrap()** command is usually used inside of a **loop()** function, just as it is here.

Sound effects would add a lot to this app. Can you add some?

DRAGONFIRE DRAW

Our homage to Trogdor.

```
1 fill('city2')
2 dragon = stamp('dragon6',0,300)
3 fire = stamp('flames',240,270,1)
4
5 function drag() {
6   stamp('crater',x,y,50)
7   burn = stamp('fire',x,y,200)
8   burn.size(0,2000)
9   fire.front().aim(x,y)
10  fire.size(distance(burn,fire)*2)
11 }
12
13 function untouch() {
14   fire.size(0,500)
15   dragon.dance()
16 }
```



We put the drag
in dragon.

Bitsbox's `drag()` function controls what happens when someone drags their cursor (or their finger) on the screen. `drag()` is an *interaction* that's really common in mobile apps: `tap()`, `touch()`, `untouch()`, and `touching()` are other types of interactions.

Click the sharing icon
on your screen to download
screenshots of your app!

Drag on the screen to
draw with dragonfire.
That's hot!

How would you make the dragon burn
more stuff with every breath? Hint: 50

Roller coaster

1 2 2 3

Draw a
crazy roller
coaster, then
watch it in
action!

Keep your arms
and legs inside
the vehicle at
all times.

```
1 fill('amusement')
2 track = []
3 car = stamp('coastercar',200).hide()
4
5 function drag(info) {
6   beam = stamp('oval',x,y,50)
7   beam.rotate(info.angle+90)
8   line(x,1100,x,y,'gold')
9   track.push(beam)
10 }
11
12 function untouch() {
13   car.front()
14   car.show()
15   loop = rollercoaster
16 }
17
18 i = 1
19 function rollercoaster() {
20   next = track[i]
21   car.aim(next)
22   car.move(next.x,next.y)
23   i = i + 1
24   if (i == track.length) {
25     loop = null
26   }
27 }
```

The untouchable loop

This app uses Bitsbox's built-in `untouch()` function to start the ride as soon as you've finished drawing the track. Lifting your finger off the screen triggers this interaction.

Then what happens? Line 15 assigns the function `rollercoaster()` to Bitsbox's built-in loop command, which turns `rollercoaster()` into a loop. Twenty times a second, the car moves to the next section of track you drew earlier. Wheeeee!

3 8 5 3

Trouble sleeping? Try counting sheep with this app!

Sleepy Sheep

Like a glass
of warm milk for
your eyeballs



Rotate and Move

Lines 7 and 8 use Bitsbox's `rotate()` and `move()` commands to make the sheep fly across the screen. Bitsbox code is *asynchronous*, so both actions happen at the same time (instead of one after the other).

```
1 fill('full moon')
2 count = 0
3 message = text(count,300,900,200)
4
5 function tap() {
6   animal = stamp('sheep2',800,200,300)
7   animal.rotate(RIGHT,400,3000)
8   animal.move(LEFT,1000,3000)
9   count = count + 1
10  message.change(count)
11 }
```

Try adding a line of code that makes the sheep get smaller as it flies. Hint: `animal.size()`