CS Bridge, Lecture 14

Breakout Extra

Plan for Today

- "Sticky paddle"
- Keyboard events

How is Breakout going so far?

Post in the chat or unmute yourself for questions or comments!

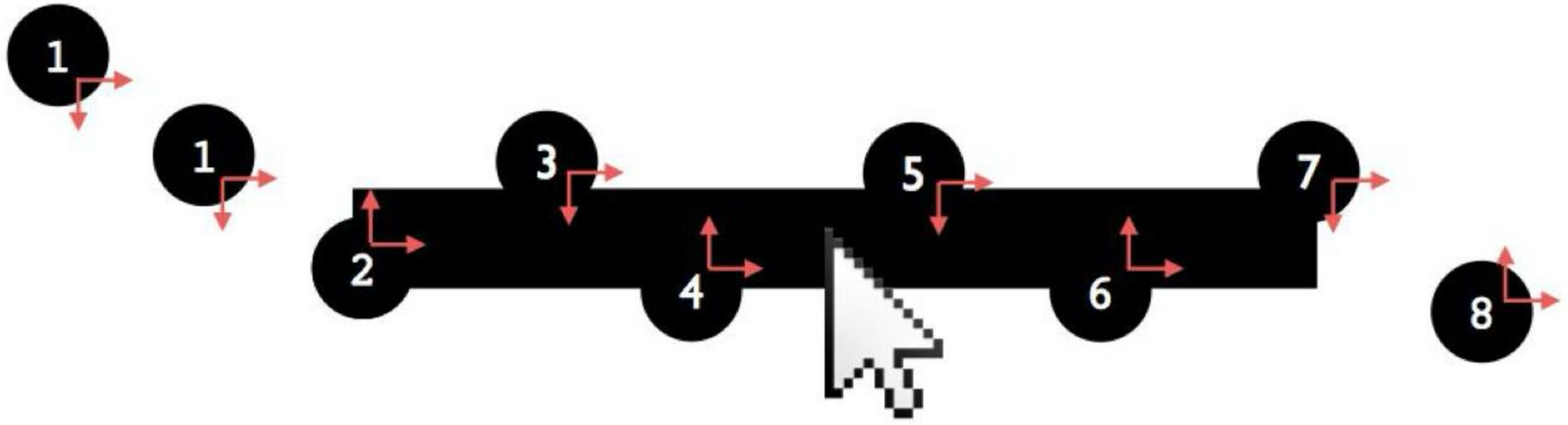
Plan for Today

- **“Sticky paddle”**
- Keyboard events

Sticky Paddle

Common Breakout bug: “sticky paddle” where ball sometimes gets stuck on the paddle.

To see it: try moving the paddle very quickly sideways into the ball so they overlap. Easier with larger paddle.

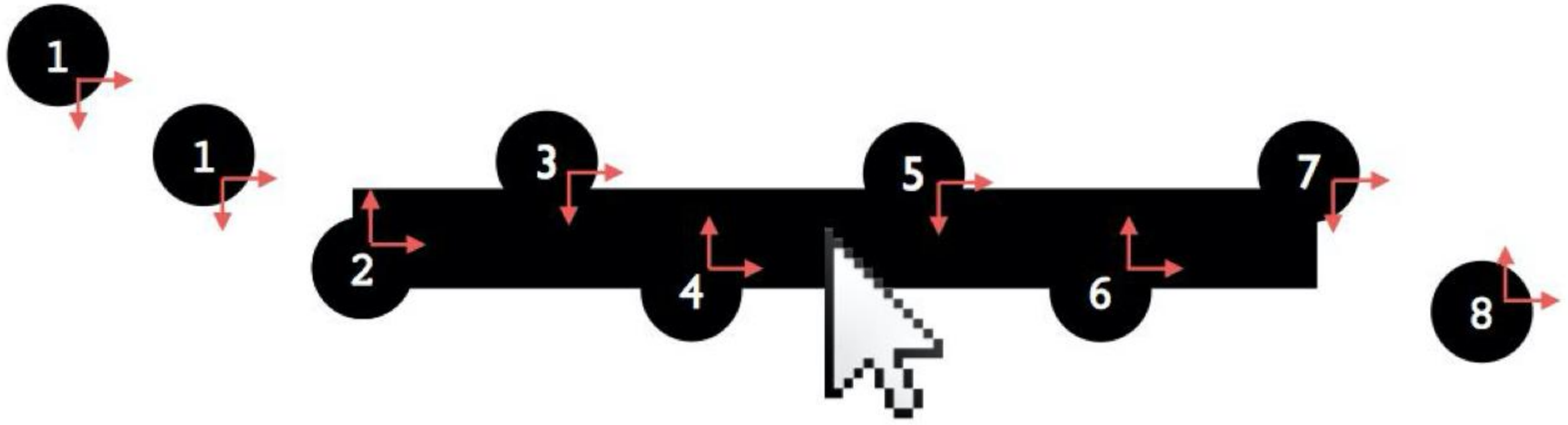


Demo: Sticky Paddle

Sticky Paddle

Common Breakout bug: “sticky paddle” where ball sometimes gets stuck on the paddle.

Hint: unlike for bricks, when bouncing off the paddle we always want the ball to then go *up*.

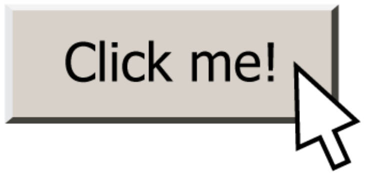


Plan for Today

- “Sticky paddle”
- **Keyboard events**

Responding To The Mouse

- **event**: Some external stimulus that your program can respond to.



Events

- Mouse clicking
- Keyboard keys pressed
- Etc.

Events

- In our programs, we can ask the canvas if any events have occurred since the last time we asked.
- If there are, then we do something.
- If there are not, we do nothing and check again later.

```
while True:
     # Handle any new keyboard events
     # ...
     canvas.update()
```

Mouse Clicks

At any time, we can ask the canvas for a list of mouse clicks that have happened since the last time we asked.

```
clicks = canvas.get_new_mouse_clicks()
```

Mouse Clicks

Each element in the list has an **x** and **y** coordinate of where that click happened.

```
clicks = canvas.get_new_mouse_clicks()
for click in clicks:
    print(click.x, click.y)
```

Events

Pattern: we make a loop (like for animation), and each time through the loop we check for new mouse clicks, and act on them.

```
while True:
     # Handle any new mouse clicks
     # ...
     canvas.update()
```

Example: Polka Dots

```
while True:
    clicks = canvas.get_new_mouse_clicks()

    # Add a circle each time the user clicks
    for click in clicks:
        circle = canvas.create_oval(click.x, click.y,
                                    click.x + CIRCLE_SIZE,
                                    click.y + CIRCLE_SIZE)
        canvas.set_color(circle, 'blue')
    canvas.update()
```

Example: Polka Dots

```
while True:
    clicks = canvas.get_new_mouse_clicks()

    # Add a circle each time the user clicks
    for click in clicks:
        circle = canvas.create_oval(click.x, click.y,
                                    click.x + CIRCLE_SIZE,
                                    click.y + CIRCLE_SIZE)
        canvas.set_color(circle, 'blue')
    canvas.update()
```

NEW: Key Presses

At any time, we can ask the canvas for a list of keyboard presses that have happened since the last time we asked.

```
presses = canvas.get_new_key_presses()
```

Key Presses

Each element in the list has a **keysym** which is the name of the key that was pressed.

```
presses = canvas.get_new_key_presses()
for press in presses:
    print(press.keysym)
```

Mouse Clicks

Each element in the list has a **keysym** which is the name of the key that was pressed.

- Letters are themselves (e.g. “a”, “b”, …), and can be uppercase if holding down Shift (e.g. “A”, “B”, …)
- Arrow keys are “Left”, “Right”, “Up” and “Down”
- Space is “space”
- …etc.

Demo: keysym

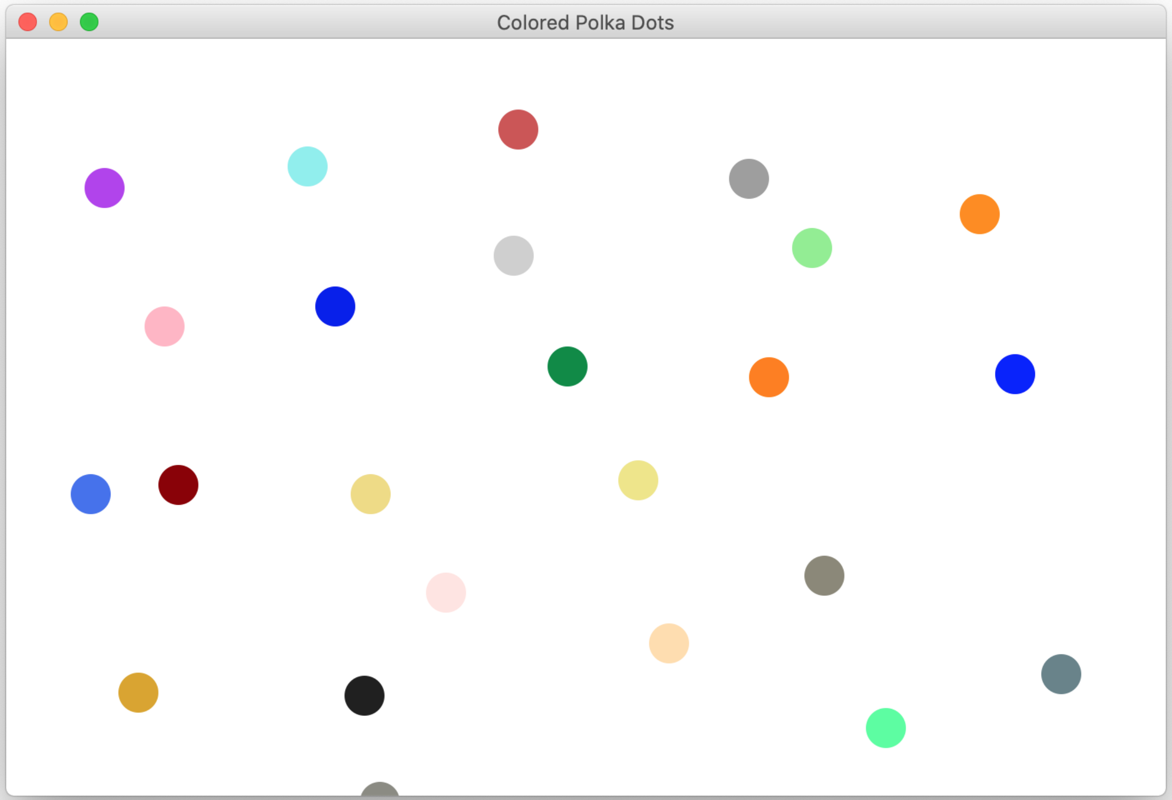
Key Presses

Each element in the list has a **keysym** which is the name of the key that was pressed.

We can check its keysym to know what key was pressed.

```
presses = canvas.get_new_key_presses()
for press in presses:
    if press.keysym == "a":
        # user hit ‘a’ key
        ...
```

Colored Polka Dots



<u>Challenge</u>: let's add to our polka dot program to randomize all the dot colors when the user hits the space bar.

Demo: Colored Polka Dots

Recap

- “Sticky paddle”
- Keyboard events

We hope you have fun working on Breakout! There are many pieces, so work step by step, and ask us for help!