

Pygame - Dev Notes - Events - Interaction

A few notes on detecting player interaction events with Pygame.

Contents

- Intro
- Detecting keyboard events

Intro

We may detect player interaction events with Pygame, and then allow a player to control shapes, animations, &c.

As the *game loop* is executed, Pygame keeps a record of interaction events for the game window. Regardless of the execution state of the *game loop*, each event is added to `events`.

We may then check `events` to see if a particular key has been pressed, or perhaps a controller button clicked.

Detecting keyboard events

We can detect interaction events for keys pressed by a player whilst the Pygame window is running.

As with detecting when a player wants to exit the Pygame windows, we can import `pygame.events` to check keys pressed.

For example, if we wanted to check for a given key press, we may add a generic listener for `KEYDOWN`, `KEYUP`, or perhaps `KEY_ESCAPE`.

```
...
# check keyboard events - keydown
if event.type == pygame.KEYDOWN:
...
```

We may then check a specific key event relative to keydown, perhaps a player request to move a shape left or right,

```
# check keyboard events - keydown
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_LEFT:
        leftDown = True
    if event.key == pygame.K_RIGHT:
        rightDown = True
```

We may also check specific lettered keys, such as the `f` character, again as part of a key press down

```
if event.key == pygame.K_f:
```

This is simply listening for a key press on the `f` key on the player's keyboard.

There are many more examples listed on the Pygame website,

```
* [Pygame - key](https://www.pygame.org/docs/ref/key.html)
```