

# Pygame - Dev Notes - Game Extras - Repetitive Firing

A brief intro on adding fun options such as repetitive firing to a game window with Pygame.

## Contents

- Intro
- Repetitive firing sequence
- References
- Demo

## Intro

For an example game, such as *Space Invaders* or a similar classic STG, we may wish to add a repetitive firing sequence for the player's sprite object.

## Repetitive firing sequence

In our current game logic, as a user presses down on the space bar, a laser beam will be fired from the top of the player's ship. However, one press is equal to one firing sequence.

We need to still check that the spacebar has been pressed down. However, it will now continue to fire a laser beam until the key is released.

In our `Player` class, we can add a couple of new variables. The first is to specify the delay in milliseconds between each firing of the laser beam, and the second allows us to check the time, the number of ticks, since the last beam was fired, e.g.

```
...
# firing delay between laser beams
self.firing_delay = 200
# time in ms since last fired
self.last_fired = pygame.time.get_ticks()
```

We may then add a listener for the space bar event to the `update()` method in the `Player` class.

```
# check space bar for firing projectile
if key_state[pygame.K_SPACE]:
    # fire laser beam
    self.fire()
```

Then, we can update our `fire()` method to reflect this repetitive firing sequence, e.g.

```
...
# get current time
time_now = pygame.time.get_ticks()
if time_now - self.last_fired > self.firing_delay:
    self.last_fired = time_now
...
```

So, our `fire()` method has now been updated as follows,

```
# fire projectile from top of player sprite object
def fire(self):
    # get current time
    time_now = pygame.time.get_ticks()
    if time_now - self.last_fired > self.firing_delay:
```

```
self.last_fired = time_now
# set position of projectile relative to player's object rect for centerx and
top
projectile = Projectile(self.rect.centerx, self.rect.top)
# add projectile to game sprites group
game_sprites.add(projectile)
# add each projectile to sprite group for all projectiles
projectiles.add(projectile)
# play laser beam sound effect
laser_effect.play()
```

We can now also remove the listener for a spacebar event in the events section of the game loop.

## References

- [pygame.key](#)
- [pygame.time](#)

## Demo

- [repetitivefiring.py](#)
- [shooter1.1.py](#)
  - add repetitive firing sequence for player's laser beam
    - move keypress check for space bar to player class
    - fire laser beam whilst space pressed down
    - set interval in ms for firing sequence
    - check time between now and last firing