

# Project Report: Infini-Platform

## I. Game Overview

### i. Genre:

Infini-Platform

### ii. Concept:

The player progresses up as the game generates more levels.

### iii. Credits:

This final project was made possible by coderslegacy.com with their tutorial on this game.

## II. Gameplay and Mechanics

### i. Objectives

The game has no objective, the player just needs to climb as high as possible.

### ii. Platforms

This class is self-explanatory with the name, this will be the basis for the platforms that the player will jump.

```
class platform(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.surf = pygame.Surface((random.randint(50,100), 12))
        self.surf.fill((0,255,0))
        self.rect = self.surf.get_rect(center = (random.randint(0,WIDTH-10),
                                                    random.randint(0, HEIGHT-30)))
```

### iii. Moving

Inside the player class, there's a function called move. The if statements for *pressed\_keys* are for the arrow keys to make the players move left and right. *Self.acc* is for the implementation of gravity.

```

def move(self):
    self.acc = vec(0,0.5)
    pressed_keys = pygame.key.get_pressed()

    if pressed_keys[K_LEFT]:
        self.acc.x = -ACC
    if pressed_keys[K_RIGHT]:
        self.acc.x = ACC

    self.acc.x += self.vel.x * FRIC
    self.vel += self.acc
    self.pos += self.vel + 0.5 * self.acc

    if self.pos.x > WIDTH:
        self.pos.x = 0
    if self.pos.x < 0:
        self.pos.x = WIDTH
    self.rect.midbottom = self.pos

```

#### iv. Jumping

The *jump* function in the *Player*, we assign a vertical value, so it goes up.

```

def jump(self):
    hits = pygame.sprite.spritecollide(self, platforms, False)
    if hits:
        self.vel.y = -15

```

Meanwhile the if statement in the *while True* statement in line 109 contains a key for making the block jump.

```

    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_SPACE:
            P1.jump()

```

### III. Level Generation

#### i. Level Generation

The code below is supposed to make more platforms when there are less than 7 platforms on the screen.

```
def plat_gen():
    while len(platforms) < 7 :
        width = random.randrange(50,100)
        p = platform()
        p.rect.center = (random.randrange(0, WIDTH - width),
                        random.randrange(-50, 0))
        platforms.add(p)
        all_sprites.add(p)
```

## ii. Infinite Scrolling Screen

The code below demonstrates that if the player has reached the  $HEIGHT / 3$ , the code will activate. It will also update the sprites and position of the player each time the player moves up, and destroys the platform below.

```
if P1.rect.top <= HEIGHT / 3:
    P1.pos.y += abs(P1.vel.y)
    for plat in platforms:
        plat.rect.y += abs(P1.vel.y)
        if plat.rect.top >= HEIGHT:
            plat.kill()
```