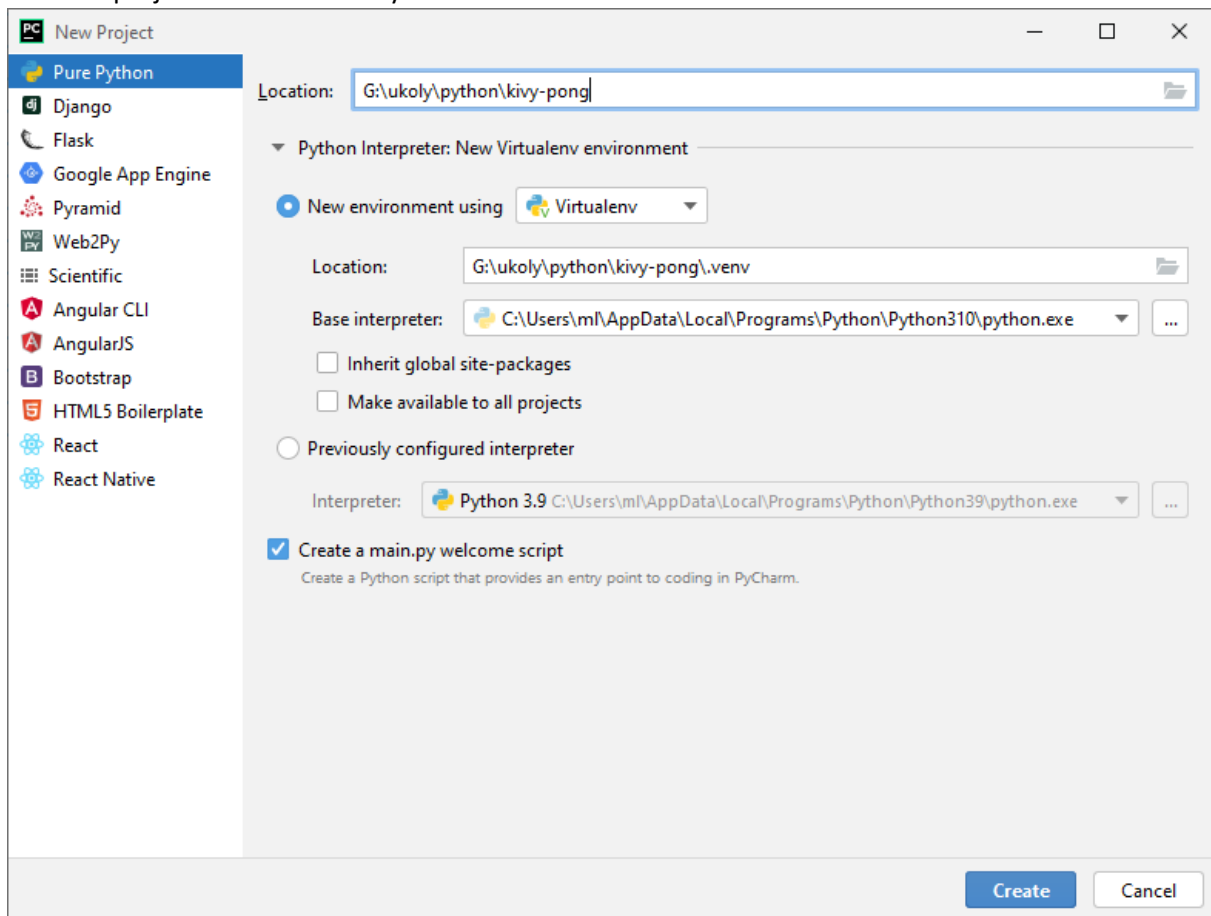
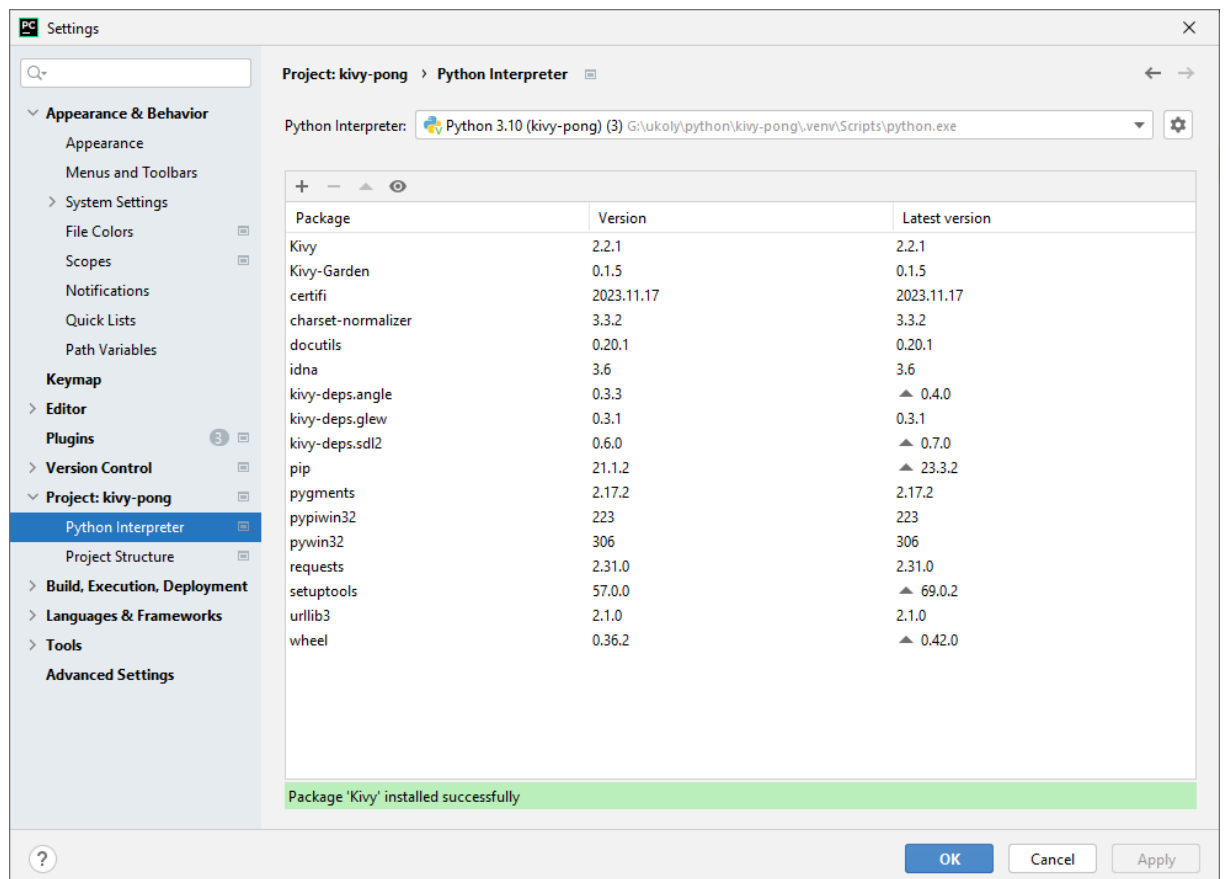
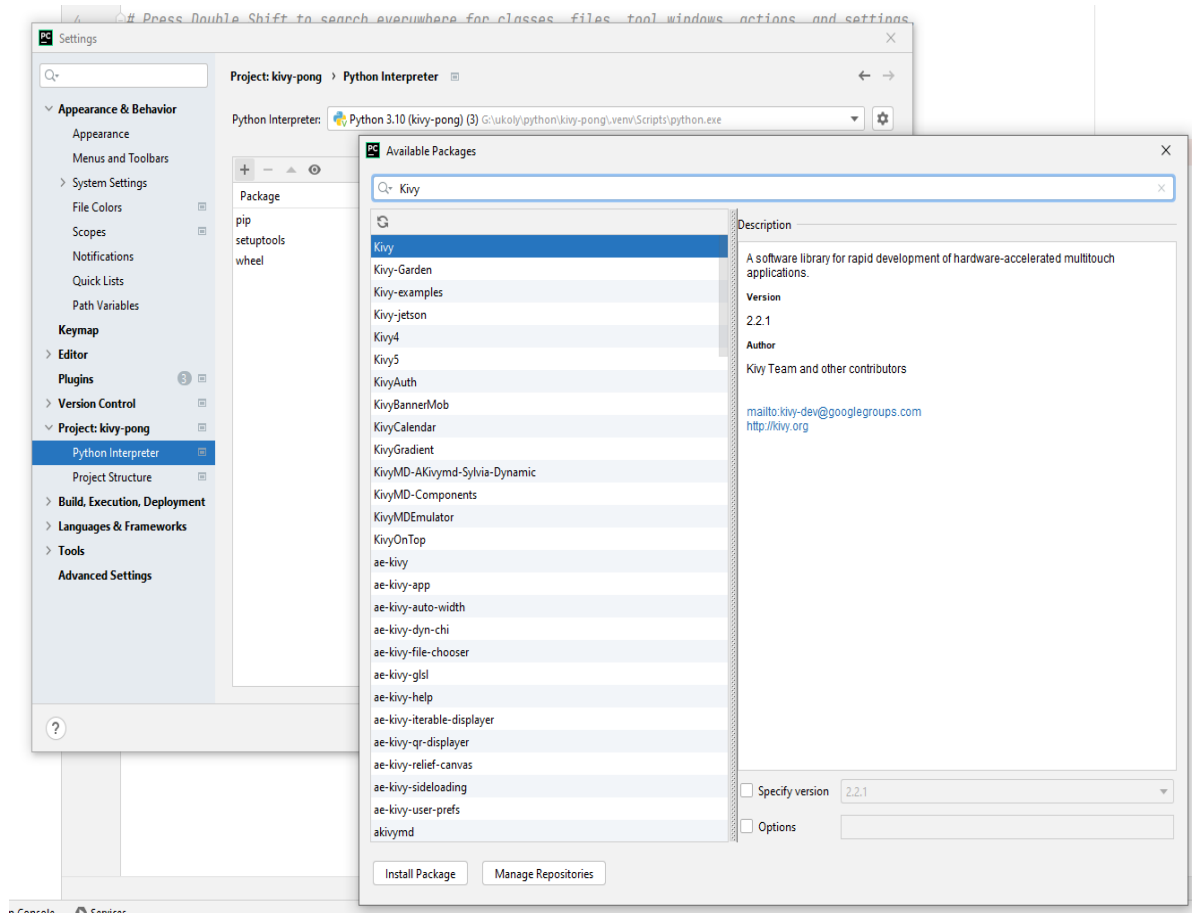


Hra Pong s využitím frameworku Kivy

1. Založení projektu a instalace Kivy





Základní kostra projektu:

```
import kivy
from kivy.app import App
from kivy.ui.widget import Widget

# Hlavní třída pro Pong hru
class PongGame(Widget):
    pass

# Třída pro spuštění hry
class PongApp(App):
    def build(self):
        return PongGame()

# Spuštění hry
if __name__ == '__main__':
    PongApp().run()
```

2. Vytvoření grafického rozhraní - soubor pong.kv

```
<Ball>:
    size: 25, 25
    canvas:
        Ellipse:
            pos: self.pos
            size: self.size

<Paddle>:
    size: 25, 100
    canvas:
        Rectangle:
            pos: self.pos
            size: self.size

<PongGame>:
    ball: pong_ball
    player1: player_left
    player2: player_right
    canvas:
        Color:
            rgba: 0, 1, 0, .5
        Rectangle:
            pos: self.pos
            size: self.size

        Color:
            rgba: 1, 1, 1, 1
        Rectangle:
            pos: self.center_x - 5, 0
            size: 10, self.height

    Label:
        font_size: 70
        center_x: root.width / 4
        top: root.top - 50
    #     text: str(root.player1.score)
        text: "0"

    Label:
        font_size: 70
        center_x: root.width * 3 / 4
        top: root.top - 50
    #     text: str(root.player2.score)
        text: "0"

    Ball:
        id: pong_ball
        center: self.parent.center

    Paddle:
        id: player_left
        x: 0
        center_y: root.center_y

    Paddle:
        id: player_right
        x: root.width - self.width
        center_y: root.center_y
```

```
# Třída pro míček
class Ball(Widget):
    pass
```

```
class Paddle(Widget):
    pass
```

3. Pohyb míčku

```
# Třída pro míček
class Ball(Widget):
    pass
    # Rychlost míčku ve dvou osách
    velocity_x = NumericProperty(0)
    velocity_y = NumericProperty(0)
    velocity = ReferenceListProperty(velocity_x, velocity_y)

    # Metoda pro aktualizaci pozice míčku
    def move(self):
        self.pos = Vector(*self.velocity) + self.pos

# Hlavní třída pro Pong hru
class PongGame(Widget):
    ball = ObjectProperty(None)

    def serve_ball(self, vel=(6, 0)):
        self.ball.center = self.center
        self.ball.velocity = vel

    #
    def update(self, dt):
        self.ball.move()

# Třída pro spuštění hry
class PongApp(App):
    def build(self):
        game = PongGame()
        game.serve_ball()
        Clock.schedule_interval(game.update, 1.0 / 60.0)
        return game
```

4. Pálky, odraz míčku

```
# Třída pro pátku
class Paddle(Widget):
    score = NumericProperty(0)  # Skóre hráče

    # Metoda pro pohyb pátky nahoru
    def move_up(self):
        self.y += 10  # Pohyb nahoru o 10 pixelů

    # Metoda pro pohyb pátky dolů
    def move_down(self):
        self.y -= 10  # Pohyb dolů o 10 pixelů

    def bounce_ball(self, ball):
        if self.collide_widget(ball):
            vx, vy = ball.velocity
            offset = (ball.center_y - self.center_y) / (self.height / 2)
            bounced = Vector(-1 * vx, vy)
            vel = bounced * 1
            ball.velocity = vel.x, vel.y + offset

class PongGame(Widget):
    ball = ObjectProperty(None)
    player1 = ObjectProperty(None)
    player2 = ObjectProperty(None)

    def serve_ball(self, vel=(6, 0)):
        self.ball.center = self.center
        self.ball.velocity = vel

    #
    def update(self, dt):
        self.ball.move()
        # bounce off paddles
        self.player1.bounce_ball(self.ball)
        self.player2.bounce_ball(self.ball)
```

5. Pohyb pátky, změna skóre

```
# Hlavní třída pro Pong hru
class PongGame(Widget):
    ball = ObjectProperty(None)
    player1 = ObjectProperty(None)
    player2 = ObjectProperty(None)

    def serve_ball(self, vel=(6, 0)):
        self.ball.center = self.center
        self.ball.velocity = vel

    #
    def update(self, dt):
        self.ball.move()
        # bounce off paddles
        self.player1.bounce_ball(self.ball)
        self.player2.bounce_ball(self.ball)

        # went off to a side to score point?
        if self.ball.x < self.x:
            self.player2.score += 1
            self.serve_ball(vel=(6, 0))
        if self.ball.right > self.width:
            self.player1.score += 1
            self.serve_ball(vel=(-6, 0))

    #
    def on_touch_move(self, touch):
        if touch.x < self.width / 3:
            self.player1.center_y = touch.y
        if touch.x > self.width - self.width / 3:
            self.player2.center_y = touch.y
```


6. Odraz míčku od horního/dolního mantinelu

```
def update(self, dt):
    self.ball.move()
    # bounce off paddles
    self.player1.bounce_ball(self.ball)
    self.player2.bounce_ball(self.ball)
    # bounce ball off bottom or top
    if (self.ball.y < self.y) or (self.ball.top > self.top):
        self.ball.velocity_y *= -1
```

7. Ovládání pálek pomocí klávesnice, omezení pohybu pálek

```
# Hlavní třída pro Pong hru
class PongGame(Widget):
    ball = ObjectProperty(None)
    player1 = ObjectProperty(None)
    player2 = ObjectProperty(None)
    #
    def __init__(self, **kwargs):
        super(PongGame, self).__init__(**kwargs)
        self._keyboard = Window.request_keyboard(self._keyboard_closed,
self)
        self._keyboard.bind(on_key_down=self._on_keyboard_down)

    def serve_ball(self, vel=(6, 0)):
        self.ball.center = self.center
        self.ball.velocity = vel

    #
    def update(self, dt):
        self.ball.move()
        # bounce off paddles
        self.player1.bounce_ball(self.ball)
        self.player2.bounce_ball(self.ball)
        # Omezení pohybu pátky
        self.player1.y = min(self.height - self.player1.height, max(0,
self.player1.y))
        self.player2.y = min(self.height - self.player2.height, max(0,
self.player2.y))

        # bounce ball off bottom or top
        if (self.ball.y < self.y) or (self.ball.top > self.top):
            self.ball.velocity_y *= -1

    #
    # went off to a side to score point?
    if self.ball.x < self.x:
        self.player2.score += 1
        self.serve_ball(vel=(6, 0))
    if self.ball.right > self.width:
        self.player1.score += 1
        self.serve_ball(vel=(-6, 0))

    #
    def on_touch_move(self, touch):
        if touch.x < self.width / 3:
            self.player1.center_y = touch.y
        if touch.x > self.width - self.width / 3:
            self.player2.center_y = touch.y

    #
    def _keyboard_closed(self):
        self._keyboard.unbind(on_key_down=self._on_keyboard_down)
        self._keyboard = None

    def _on_keyboard_down(self, keyboard, keycode, text, modifiers):
        if keycode[1] == 'w':
            self.player1.move_up()
        elif keycode[1] == 's':
            self.player1.move_down()
        # Přidat klávesy pro druhého hráče
        elif keycode[1] == 'up':
            self.player2.move_up()
        elif keycode[1] == 'down':
            self.player2.move_down()
```

Window.request_keyboard: Tato metoda je volána pro získání instance klávesnice z aktuálního okna. Metoda **request_keyboard** se používá k získání odkazu na klávesnici, aby mohl váš program reagovat na stisky kláves.

self._keyboard_closed: Toto je odkaz na metodu nebo funkci, která bude volána, když se klávesnice "uzavře" nebo "zruší", například když uživatel klikne mimo textové pole nebo když aplikace ztratí fokus. Je to druh "callbacku", který zajišťuje správné uvolnění zdrojů nebo ukončení naslouchání klávesnici, když už není potřeba.

self: Tento argument specifikuje, kdo je "requester" klávesnice, tedy objekt, který o klávesnici požádal. V tomto případě to je instance třídy, ve které je tento kód umístěn.

self._keyboard.bind(on_key_down=self._on_keyboard_down):

self._keyboard.bind(...): Tato metoda se používá k navázání události klávesnice na konkrétní funkci nebo metodu.

on_key_down=self._on_keyboard_down: Tento argument říká, že kdykoliv je stisknuta klávesa (událost **on_key_down**), měla by být volána metoda **self._on_keyboard_down**.

self._on_keyboard_down je metoda, kterou definujete v rámci své třídy, a která obsluhuje logiku toho, co se stane, když je stisknuta určitá klávesa.

Celkově tento kód říká: "Chci naslouchat klávesnici. Když je klávesa stisknuta, zavolej metodu **_on_keyboard_down**. Když klávesnice přestane být používána, zavolej metodu **_keyboard_closed**." Toto je základní pro zajištění interakce s klávesnicí ve vaší aplikaci Kivy.