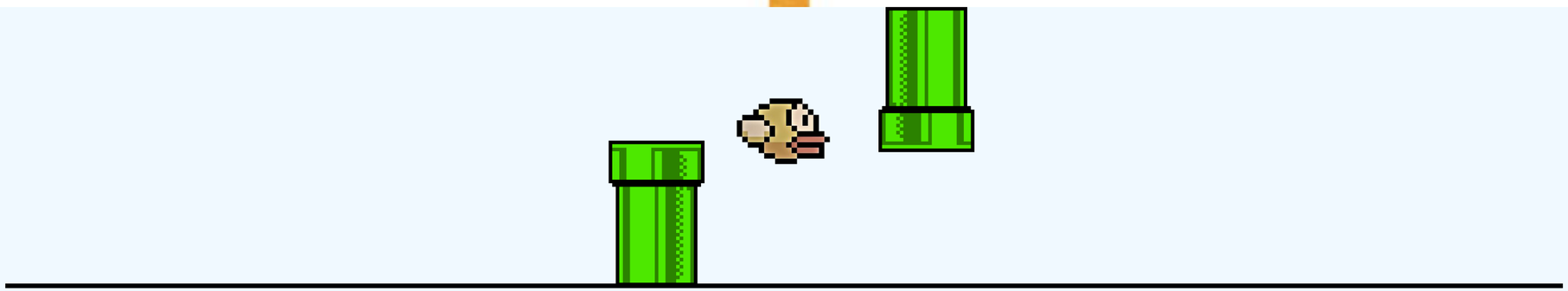
FLAPPY
BIRD
START

MEMBER

- Marsella Yesi
- Natalia S
- Adriel Gideon
- Natanael Argajova
- M Farhan Annnaufal
- M Umar Basyir
- Ghaza Muhammad Al-Ghiffari

KONSEP



Konsep dalam Game ini Sederhana, yaitu Pemain mengendalikan seekor burung yang harus terbang melintasi rintangan-rintangan pipa

OBJEK

BIRD
OBJEK UTAMA DALAM GAME FLAPPY BIRD, MERUPAKAN KARAKTER YANG DIKENDALIKAN OLEH PEMAIN UNTUK TERBANG MELALUI RINTANGAN-RINTANGAN PIPA.

PIPE
OBJEK RINTANGAN UTAMA DALAM GAME FLAPPY BIRD, TERDIRI DARI PIPA-PIPA YANG TERLETAK DI ATAS DAN DI BAWAH LAYAR. BURUNG HARUS TERBANG MELALUINYA DENGAN PRESISI YANG TEPAT UNTUK TIDAK MENABRAK PIPA.

BACKGROUND
LATAR BELAKANG LAYAR PADA GAME FLAPPY BIRD, BIASANYA TERDIRI DARI LANGIT DAN AWAN YANG BERGERAK SECARA LAMBAT.

SCOREBOARD
OBJEK YANG MENUNJUKKAN SKOR PEMAIN SELAMA BERMAIN GAME.MENU:

MENU
OBJEK YANG MENUNJUKKAN OPSI MENU PADA GAME FLAPPY BIRD, SEPERTI OPSI UNTUK MEMULAI GAME, MENAMPILKAN SKOR TERBAIK, ATAU KELUAR DARI GAME.

GROUND
LANTAI PADA GAME FLAPPY BIRD, BURUNG AKAN JATUH ATAU MATI JIKA MENABRAK LANTAI.

C. Japan

PENERAPAN KONSEP PBO

PERWARISAN

KELAS PIPE BISA MENJADI SUBKELAS DARI KELAS OBSTACLE ATAU KELAS BIRD BISA MENJADI SUBKELAS DARI KELAS ANIMAL.

POLIMORFISME

KELAS BIRD DAN KELAS OBSTACLE MASING-MASING MEMILIKI METODE MOVE() YANG BERBEDA.

ENKAPSULASI

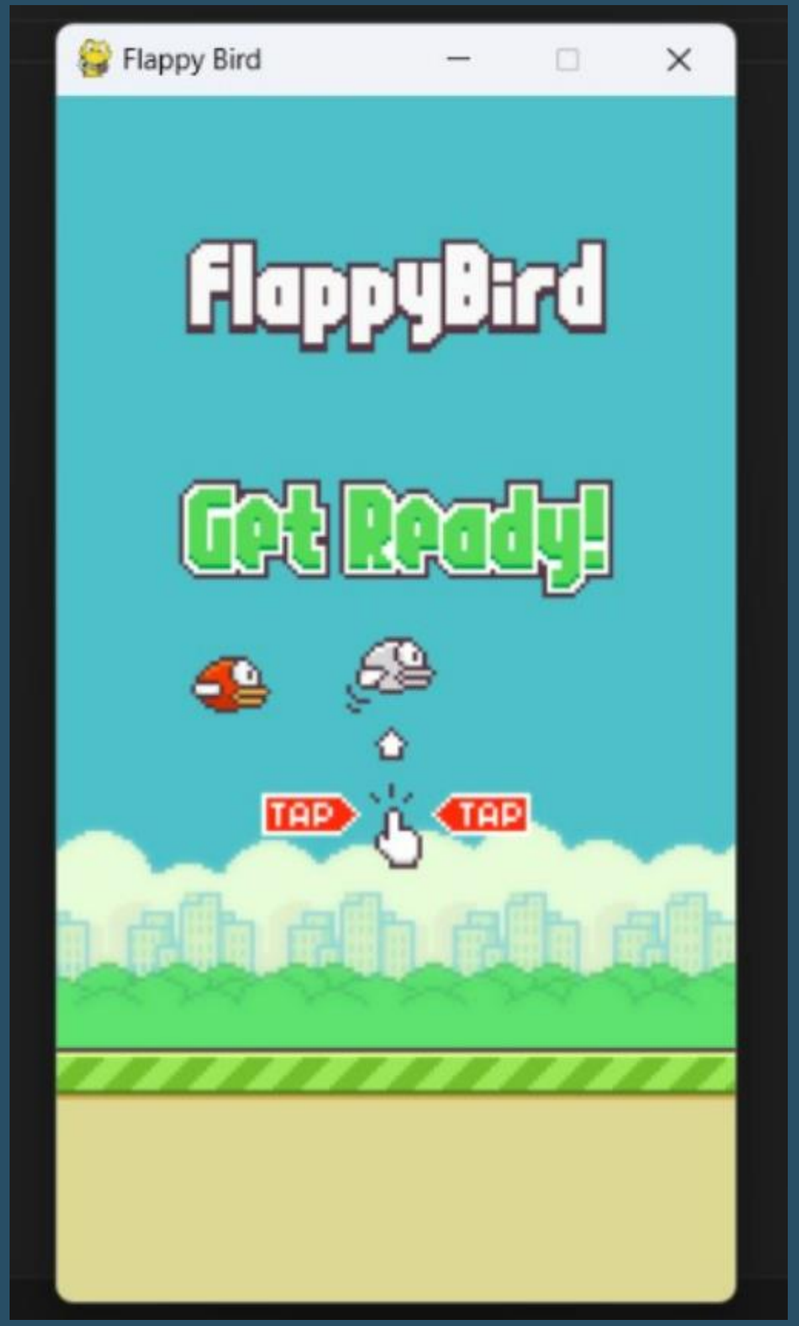
VARIABEL SCORE PADA KELAS SCOREBOARD TIDAK BISA DIAKSES ATAU DIMODIFIKASI DARI LUAR KELAS TERSEBUT.

ABSTRAKSI

KELAS MENU BISA MENGEKSPOS OPSI YANG TERSEDIA PADA MENU NAMUN TIDAK MENUNJUKKAN DETAIL IMPLEMENTASI DARI SETIAP OPSI TERSEBUT.

TAMPILAN MENU

TAMPILAN MENU UTAMA PADA GAME FLAPPYBIRD. KLIK LAYAR UNTUK MEMULAI PERMAINAN.

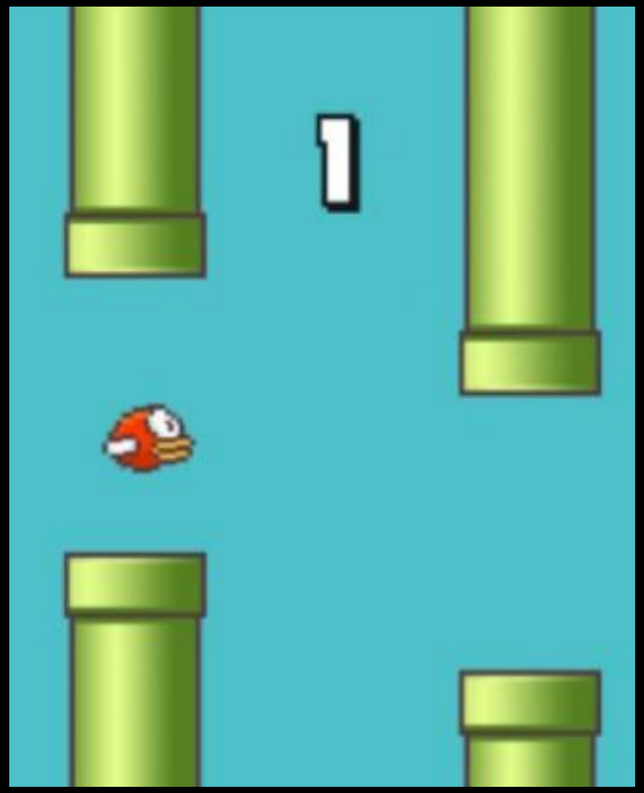


DISPLAY

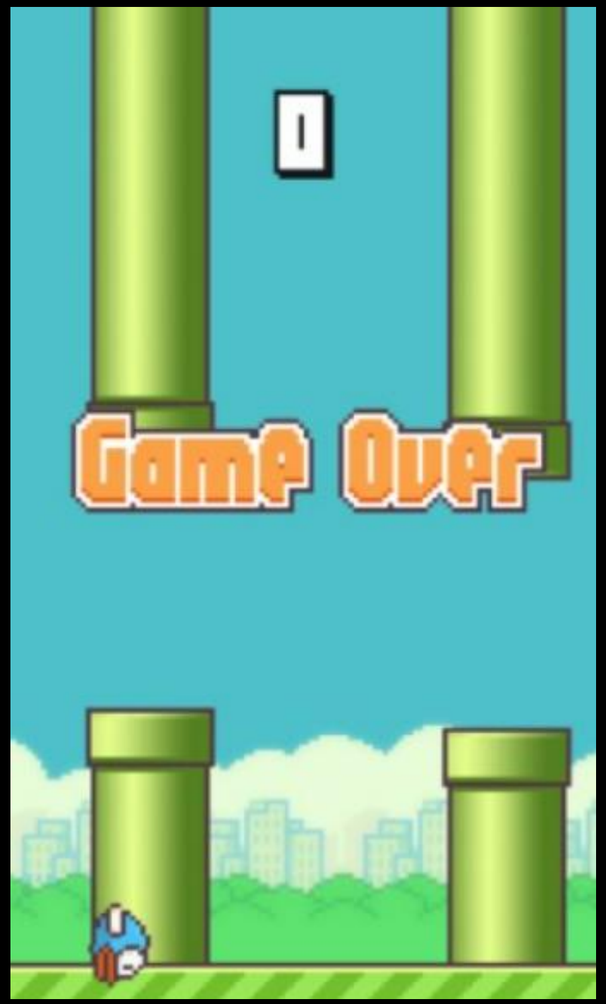
TAMPILAN MENU PADA GAME FLAPPYBIRD. KLIK LAYAR UNTUK MEMULAI PERMAINAN.

GAMEPLAY

TAMPILAN SAAT USER MENDAPATKAN SCORE



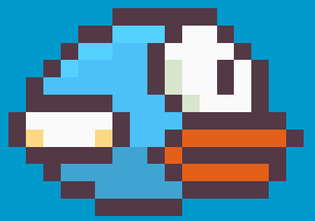
TAMPILAN SAAT USER MENABRAK PIPA ATAU TANAH..



RANDOMIZER

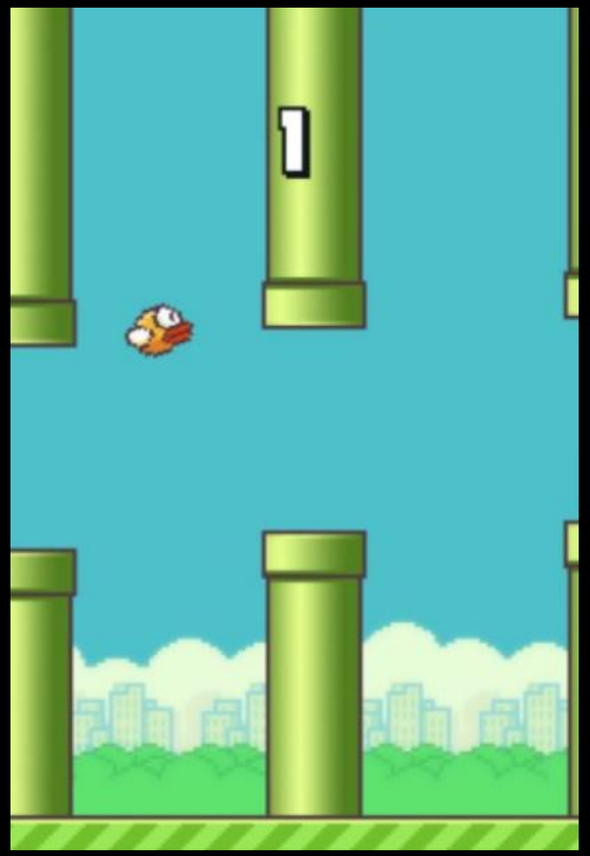
RED BIRD

YELLOW BIRD



BLUE BIRD

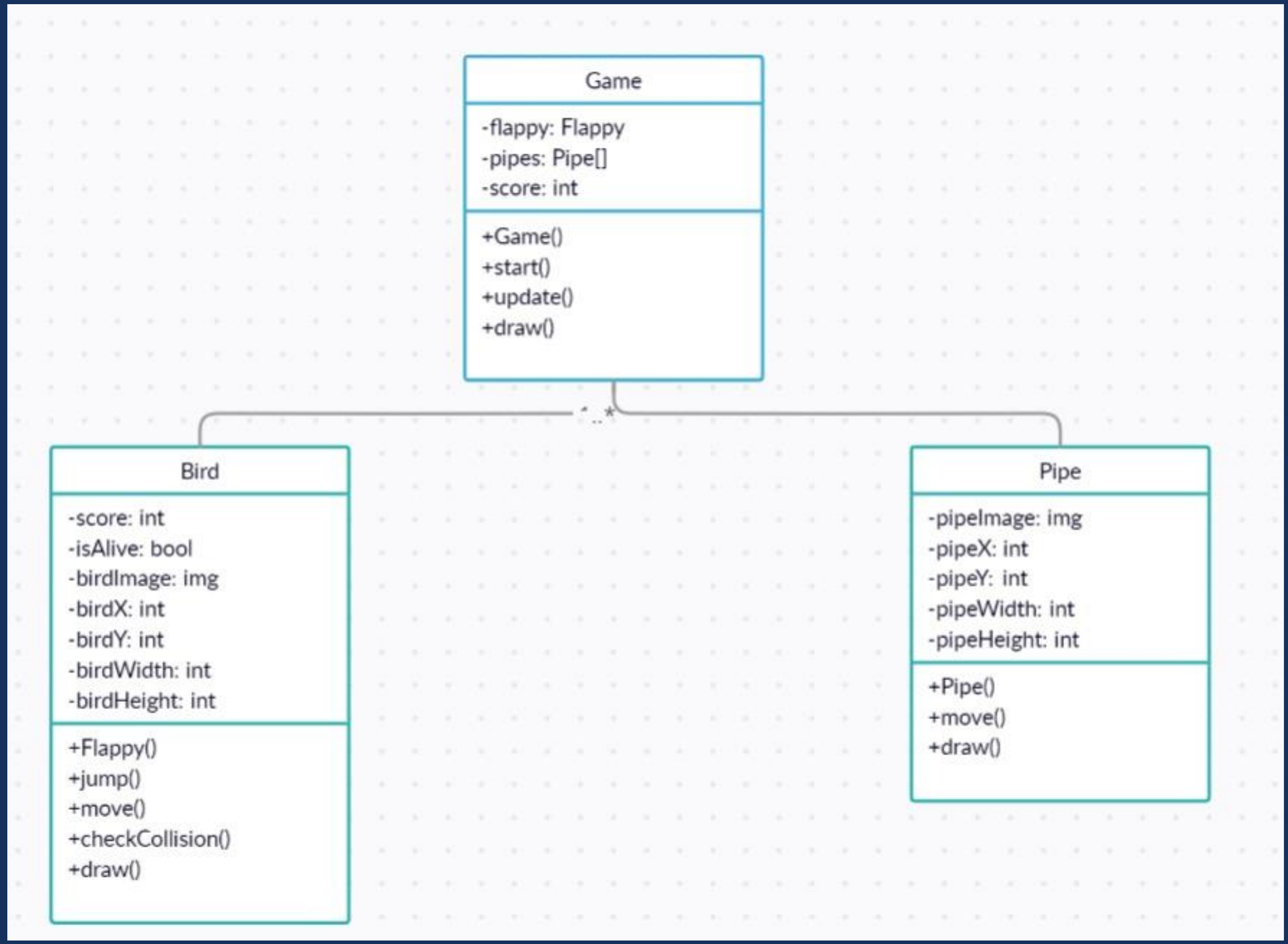
RANDOMIZER



TAMPILAN PAGI HARI

TAMPILAN MALAM HARI

UML


Game
-flappy: Flappy
-pipes: Pipe[]
-score: int
+Game()
+start()
+update()
+draw()
1
*
Bird
-score: int
-isAlive: bool
-birdImage: img
-birdX: int
-birdY: int
-birdWidth: int
-birdHeight: int
+Flappy()
+jump()
+move()
+checkCollision()
+draw()
Pipe
-pipeImage: img
-pipeX: int
-pipeY: int
-pipeWidth: int
-pipeHeight: int
+Pipe()
+move()
+draw()

ENKAPSULASI

```
def playerShm(playerShm):
    """oscillates the value of playerShm['val'] between 8 and -8"""
    if abs(playerShm['val']) == 8:
        playerShm['dir'] *= -1

    if playerShm['dir'] == 1:
        playerShm['val'] += 1
    else:
        playerShm['val'] -= 1
```

main.py 9+

C: > Users > annau > OneDrive > Pictures > flappy bird > main.py > ...

```python

def showWelcomeAnimation():
    """Shows welcome screen animation of flappy bird"""

    playerIndex = 0
    playerIndexGen = cycle([0, 1, 2, 1])

    loopIter = 0

    playerx = int(SCREENWIDTH * 0.2)
    playery = int((SCREENHEIGHT - IMAGES['player'][0].get_height()) / 2)

    messagex = int((SCREENWIDTH - IMAGES['message'].get_width()) / 2)
    messagey = int(SCREENHEIGHT * 0.12)

    basex = 0

    baseShift = IMAGES['base'].get_width() - IMAGES['background'].get_width()


    playerShmVals = {'val': 0, 'dir': 1}

    while True:
        for event in pygame.event.get():
            if event.type == QUIT or (event.type == KEYDOWN and event.key == K_ESCAPE):
                pygame.quit()
                sys.exit()
            if event.type == KEYDOWN and (event.key == K_SPACE or event.key == K_UP):
                SOUNDS['wing'].play()
                return {
                    'playery': playery + playerShmVals['val'],
                    'basex': basex,
                    'playerIndexGen': playerIndexGen,
                }

```

```
376  def playerShm(playerShm):
377      """oscillates the value of playerShm['val'] between 8 and -8"""
378      if abs(playerShm['val']) == 8:
379          playerShm['dir'] *= -1
380
381      if playerShm['dir'] == 1:
382          playerShm['val'] += 1
383      else:
384          playerShm['val'] -= 1
385
386
387  def getRandomPipe():
388      """returns a randomly generated pipe"""
389      gapY = random.randrange(0, int(BASEY * 0.6 - PIPEGAPSIZE))
390      gapY += int(BASEY * 0.2)
391      pipeHeight = IMAGES['pipe'][0].get_height()
392      pipeX = SCREENWIDTH + 10
393
394      return [
395          {'x': pipeX, 'y': gapY - pipeHeight},  # upper pipe
396          {'x': pipeX, 'y': gapY + PIPEGAPSIZE}, # lower pipe
397      ]
398
```

INHERITANCE

```python
def mainGame(movementInfo):
    score = playerIndex = loopIter = 0
    playerIndexGen = movementInfo['playerIndexGen']
    playerx, playery = int(SCREENWIDTH * 0.2), movementInfo['playery']

    basex = movementInfo['basex']
    baseShift = IMAGES['base'].get_width() - IMAGES['background'].get_width()

    newPipe1 = getRandomPipe()
    newPipe2 = getRandomPipe()

    upperPipes = [
        {'x': SCREENWIDTH + 200, 'y': newPipe1[0]['y']},
        {'x': SCREENWIDTH + 200 + (SCREENWIDTH / 2), 'y': newPipe2[0]['y']},
    ]

    lowerPipes = [
        {'x': SCREENWIDTH + 200, 'y': newPipe1[1]['y']},
        {'x': SCREENWIDTH + 200 + (SCREENWIDTH / 2), 'y': newPipe2[1]['y']},
    ]

    pipeVelX = -4

    # player velocity, max velocity, downward accleration, accleration on flap
    playerVelY    =  -9   # player's velocity along Y, default same as playerFlapped
    playerMaxVelY =  10   # max vel along Y, max descend speed
    playerMinVelY =  -8   # min vel along Y, max ascend speed
    playerAccY    =   1   # players downward accleration
    playerRot     =  45   # player's rotation
    playerVelRot  =   3   # angular speed
    playerRotThr  =  20   # rotation threshold
    playerFlapAcc =  -9   # players speed on flapping
    playerFlapped = False # True when player flaps
```

POLI
MORF
ISME

TERIMA KASIH