PROJECT UAS
Pemrograman Berorientasi Objek

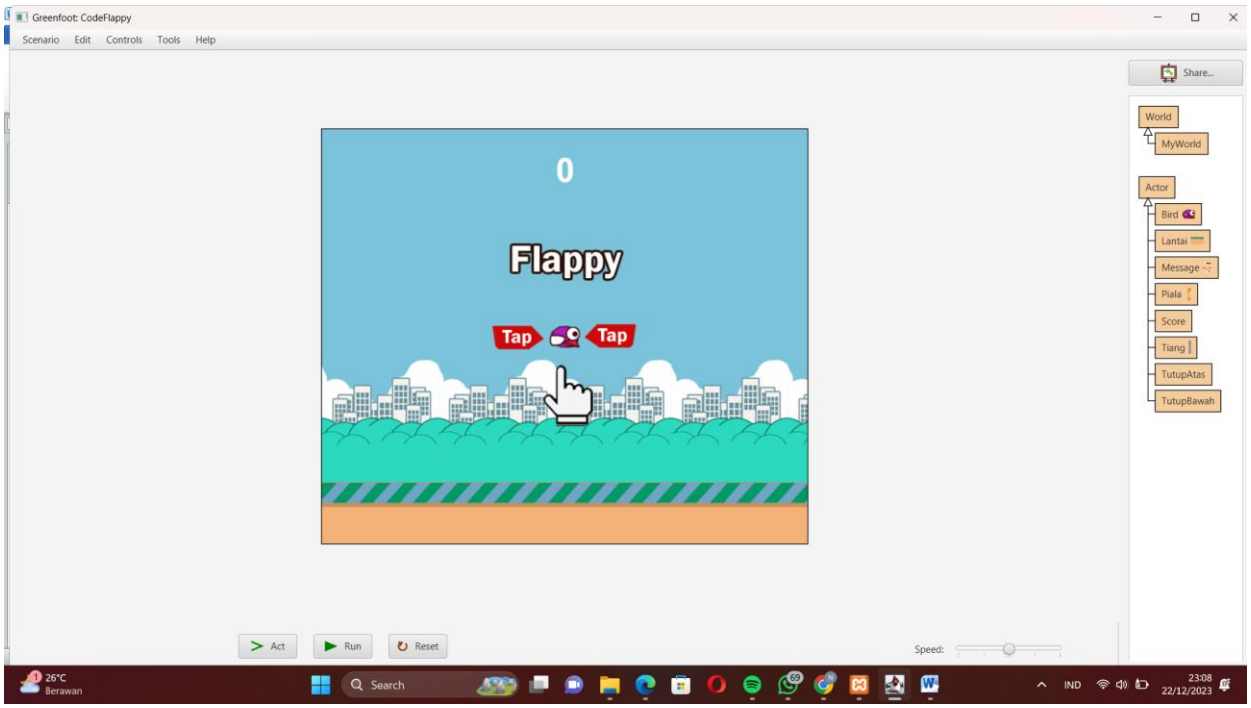
Oleh kelompok :

1. *Nurul Marisya Yastria* *(2211102441124)*
2. *Masni* *(2211102441100)*
3. *Fitriayana* *(2211102441066)*

Teknik Informatika Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2023

PENGERTIAN



CodeFlappy adalah permainan yang terinspirasi dari permainan populer "Flappy Bird". Dalam permainan ini, pemain mengontrol pergerakan burung atau objek yang harus melewati rintangan dengan cara menekan tombol atau layar untuk membuat objek tersebut terbang lebih tinggi atau turun secara terkendali. Tujuan utama pemain adalah bertahan sebanyak mungkin dengan melewati rintangan-rintangan yang muncul.

PENJELASAN CODE PADA GAME

1. **MyWorld :**

   Codingan tersebut merupakan bagian dari program yang menggunakan Greenfoot, sebuah framework untuk pembuatan permainan 2D. Ini adalah kelas MyWorld yang merupakan dunia (world) dalam permainan tersebut.

- Variabel dan Konstanta: Ada beberapa variabel seperti KECEPATAN, KONDISI, HIGHSCORE, dan lain-lain yang mengatur kondisi permainan seperti kecepatan, status permainan, serta skor tertinggi.
- Constructor MyWorld(): Method ini menginisialisasi dunia permainan. Mengatur latar belakang, menambahkan objek seperti Bird, Score, TutupBawah, TutupAtas, dan lainnya ke dalam dunia.

- addScore(): Method yang memperbarui dan menampilkan skor pemain ketika berhasil melewati rintangan.
- setNewCelah(int id): Method yang menetapkan posisi dan parameter untuk setiap rintangan yang harus dilewati oleh Bird.
- setKemenangan() & gameover(): Method yang menangani kondisi kemenangan atau kekalahan dalam permainan.
- generateCelah(): Method yang seharusnya membangkitkan posisi rintangan, tetapi tampaknya hanya mencetak posisi acak rintangan ke konsol dan tidak digunakan dalam implementasi utama.
- act(): Method ini dipanggil secara terus menerus oleh Greenfoot. Didalamnya, terdapat logika utama permainan seperti pergerakan rintangan, penambahan skor, serta penanganan klik mouse untuk memulai dan mengakhiri permainan.
- Class ini terdapat penerapan **INHERITANCE & OVERRIDING**

2. Bird :

kelas Bird dalam permainan Flappy Bird-style yang menggunakan Greenfoot framework. Ini mengatur perilaku burung atau objek yang dikontrol oleh pemain dalam permainan.

- Variabel :

  - img0, img1, img2: Gambar-gambar untuk berbagai posisi burung (terbang ke atas, tengah, dan ke bawah).
  - arah: Variabel yang mengatur arah perubahan posisi burung.
  - posisi: Menyimpan posisi saat ini dari gambar burung.
  - jeda: Mengatur jeda antara perubahan gambar burung.
  - rotate: Menyimpan nilai rotasi burung.
  - py: Menyimpan posisi vertikal burung.
  - vy: Menyimpan kecepatan vertikal burung.
  - gy: Gravitasi yang memengaruhi pergerakan burung.
  - dt: Interval waktu.
  - DIE: Menandakan jika burung telah mati (menabrak rintangan atau tidak).
- Method addedToWorld(World latar):

  - Dipanggil ketika objek burung ditambahkan ke dalam dunia.
  - Menginisialisasi gambar-gambar yang digunakan untuk berbagai posisi burung.

- Method act():

  - Method ini dipanggil secara terus-menerus oleh Greenfoot dan mengatur logika utama dari perilaku burung.

  - Jika burung menabrak objek Tiang (rintangan), burung berhenti dan DIE diatur menjadi true, suara "hit.wav" diputar.

  - Mengatur perubahan gambar burung secara periodik sesuai dengan posisinya.

  - Mengatur perilaku burung saat pemain mengklik mouse (terbang ke atas dengan kecepatan vy yang diubah sesuai).

  - Menghitung pergerakan vertikal burung berdasarkan gravitasi (gy) dan waktu (dt).

  - Mengontrol pergerakan burung dan menyesuaikan rotasinya saat bergerak naik atau turun.

  -Jika burung mencapai batas bawah layar permainan, burung berhenti, DIE diatur menjadi true, dan method gameover() dijalankan jika kondisi permainan memungkinkan, suara "die.wav" diputar.

- Class ini terdapat penerapan **INHERITANCE, POLYMORFORME & OVERRIDING.**

3. Lantai :

Kelas Lantai adalah bagian dari elemen-elemen visual dalam permainan yang bertujuan untuk membuat lantai yang bergerak ke kiri dengan kecepatan yang telah ditentukan.

- Public class Lantai extends Actor: Mendefinisikan kelas Lantai yang merupakan subclass dari kelas Actor dalam Greenfoot. Artinya, objek Lantai dapat dimanipulasi dan ditempatkan di dunia permainan Greenfoot.
- Public static int plg=0;: Mendeklarasikan variabel statik plg yang menyimpan nilai posisi lantai. Nilai ini akan berubah ketika lantai mencapai batas tertentu sehingga lantai bisa kembali ke posisi awal atau memindahkan lokasi lantai.
- Method act()

  - int px=getX()-MyWorld.KECEPATAN; :Mendapatkan posisi x saat ini dari objek Lantai dan mengurangi kecepatan dari kelas MyWorld. Hal ini dilakukan untuk menggerakkan Lantai ke kiri.

- if(px<-0.5*getImage().getWidth()): Mengecek apakah Lantai telah melewati batas kiri layar. Jika iya, artinya Lantai sudah berada di luar layar dan harus diposisikan kembali ke posisi awal.

- px=plg;: Jika Lantai sudah melewati batas kiri layar, posisi Lantai akan dipindahkan ke posisi plg, yang merupakan posisi awalnya.

- setLocation(px,getY());: Mengatur ulang lokasi Lantai ke posisi baru yang telah dihitung sebelumnya.

- Class ini terdapat penerapan **INHERITANCE.**

4. Message :

Message, yang digunakan untuk membuat efek animasi pesan yang bergerak naik dan turun. Berikut adalah penjelasan dari kode tersebut:

- Variabel dan Atribut Kelas:

  - arah: Menunjukkan arah pergerakan pesan. Nilai -1 mengarah ke atas, sementara nilai 1 mengarah ke bawah.

  - hitung: Menghitung langkah-langkah pergerakan pesan.

  - panjang: Menyimpan panjang langkah maksimum yang akan dijalankan sebelum perubahan arah.

  - jeda: Menyimpan nilai jeda atau delay untuk menunda pergerakan pesan setelah sejumlah langkah tertentu.

- Metode act():

  - Pengecekan Langkah: Memeriksa apakah jumlah langkah yang dihitung (hitung) sudah mencapai panjang maksimum (panjang). Jika sudah, nilai arah akan dibalik agar pesan bergerak ke arah yang berlawanan.

  - Pergerakan Pesan: Melakukan pergerakan pesan dengan mengubah posisi vertikalnya (setLocation(getX(), getY() + arah)). Pergerakan dilakukan dengan interval waktu yang ditentukan oleh nilai jeda.

- Class ini terdapat penerapan **INHERITANCE.**

5. Piala :

Ini adalah kelas yang belum diimplementasikan aksinya (act() method masih kosong), dan tidak memiliki atribut atau perilaku khusus yang ditentukan di dalamnya.

Dalam kode tersebut:

- Kelas Piala merupakan turunan dari kelas Actor dalam Greenfoot.
- Komentar yang diberikan menjelaskan bahwa kelas ini adalah untuk representasi objek piala dalam permainan atau dunia yang sedang dibuat.
- Metode act() adalah tempat di mana aksi atau perilaku dari objek piala akan diimplementasikan. Namun, saat ini metode ini kosong, yang berarti belum ada aksi spesifik yang dijalankan ketika objek piala tersebut berinteraksi dengan dunia permainan.
- Class ini terdapat penerapan **INHERITANCE & OVERRIDING**.

6. Score :

Kelas Score yang digunakan untuk menampilkan skor dalam lingkungan Greenfoot.

Berikut penjelasan dari kode tersebut:

- Kelas Score merupakan turunan dari kelas Actor dalam Greenfoot, yang digunakan untuk menampilkan elemen di layar permainan.
- Terdapat satu metode yang disebut setScore(int a). Metode ini bertugas untuk mengatur tampilan skor dengan mengubah gambar (image) yang ditampilkan oleh objek Score.

  - setImage(new GreenfootImage(""+a, 54, Color.WHITE, null)):

  * Membuat gambar baru menggunakan kelas **GreenfootImage**.
  * **""+a** mengonversi nilai a ke dalam string.
  * **54** adalah ukuran font yang digunakan.
  * **Color.WHITE** menentukan warna teks.
  * **null** pada parameter terakhir adalah untuk latar belakang kosong, sehingga hanya teks skor yang ditampilkan tanpa latar belakang.
- Class ini terdapat penerapan **INHERITANCE**.

7. Tiang :

Berikut adalah penjelasan dari kode tersebut:

- Kelas Tiang adalah turunan dari kelas Actor pada Greenfoot, yang berfungsi sebagai representasi dari objek tiang dalam permainan.
- Terdapat sebuah atribut publik yaitu SETNILAI yang memiliki nilai awal false. Atribut

ini nantinya dapat digunakan untuk mengatur suatu nilai yang akan digunakan dalam logika permainan.

- Metode act() adalah metode yang dipanggil secara terus-menerus oleh Greenfoot selama permainan berjalan.

  - Kode di dalamnya melakukan pengecekan kondisi **MyWorld.KONDISI > 0**.

  - Jika kondisi tersebut terpenuhi, maka posisi tiang akan berpindah seiring dengan pergerakan permainan ke arah yang berlawanan dengan **MyWorld.KECEPATAN**. Artinya, ketika kondisi permainan dalam keadaan tertentu (dalam hal ini ketika **MyWorld.KONDISI** lebih besar dari 0), tiang akan bergerak ke kiri dengan kecepatan yang ditentukan oleh **MyWorld.KECEPATAN**.
- Class ini terdapat penerapan **INHERITANCE**.

8. TutupAtas :

Berikut penjelasan dari kode tersebut:

- Kelas TutupAtas adalah turunan dari kelas Actor dalam Greenfoot, yang berfungsi sebagai elemen tutup layar atas dalam permainan.
- Terdapat beberapa atribut privat:

  - **jeda, trans, posx, posy, persen, arah** dan **mode** untuk mengatur logika dari efek visual dan pergerakan elemen ini.

  - **img** adalah objek GreenfootImage yang digunakan untuk menampung gambar yang akan ditampilkan.
- Terdapat beberapa konstruktor:

  - TutupAtas() adalah konstruktor kosong tanpa implementasi.

  - TutupAtas(int m) adalah konstruktor yang menerima parameter m yang mengatur mode tutup atas (0 atau 1).
- Metode **setMode(int m)** digunakan untuk mengatur mode dari tutup atas dan memanggil fadein(mode) untuk mengatur gambar yang akan ditampilkan.
- Metode **addedToWorld(World latar)** dipanggil saat objek ini ditambahkan ke dalam dunia permainan. Metode ini menginisialisasi posisi dan memanggil fadein(0) untuk mengatur tampilan awal elemen tersebut.
- Metode **fadein(int a)** digunakan untuk mengatur transparansi gambar. Gambar akan berubah sesuai dengan mode dan transparansi yang diatur berdasarkan parameter a.

- Metode **act()** merupakan metode yang dipanggil terus menerus oleh Greenfoot selama permainan berlangsung. Di dalamnya terdapat logika pergerakan elemen tutup atas, pengaturan transparansi gambar, dan efek visual perubahan posisi elemen tersebut.
- Class ini terdapat penerapan **INHERITANCE & OVERRIDING.**

9. TutupBawah :

Kelas TutupBawah dalam lingkungan Greenfoot bertanggung jawab atas tampilan dan efek visual dari elemen tutup layar bawah dalam permainan. Berikut penjelasan dari kode tersebut:

- Terdapat beberapa atribut privat seperti **jeda, trans, posx, posy, persen,** dan **arah** yang mengatur logika efek visual dan pergerakan elemen ini.
- Metode **addedToWorld(World latar)** dipanggil saat objek ini ditambahkan ke dalam dunia permainan. Metode ini menginisialisasi posisi dan memanggil **fadein(0)** untuk menyiapkan tampilan awal elemen tersebut.
- Metode **fadein(int a)** dan **fadeout(int a)** digunakan untuk mengatur tampilan gambar yang ditampilkan di elemen ini berdasarkan nilai parameter **a**. Mereka mengatur gambar, transparansi, dan teks yang akan ditampilkan pada tutup bawah.
- Metode **act()** merupakan metode yang dipanggil terus menerus oleh Greenfoot selama permainan berlangsung. Di dalamnya terdapat logika pergerakan elemen tutup bawah, pengaturan transparansi gambar, efek visual, dan pergantian dunia permainan ketika permainan selesai.
- Class ini terdapat penerapan **INHERITANCE & OVERRIDING**.

KESIMPULAN

"CodeFlappy" adalah sebuah permainan sederhana yang dibuat menggunakan lingkungan pengembangan permainan Greenfoot. Game ini mengadaptasi konsep yang mirip dengan game populer "Flappy Bird". Pemain mengendalikan sebuah karakter burung yang harus melewati rintangan berupa tiang-tiang dengan cara melompati mereka.

Tujuan pemain adalah bertahan sebanyak mungkin dengan melewati sebanyak mungkin tiang tanpa menabraknya. Setiap kali burung berhasil melewati tiang, pemain mendapatkan poin/skor. Namun, jika burung menabrak tiang atau jatuh ke tanah, permainan berakhir.