

DODGE GAME

Nama : khaerul umam

Kelas : 2Ti03

Nim : 2222105106

Program ini membangun sebuah game sederhana yang disebut "Dodge Game". Tujuan dari game ini adalah untuk menghindari rintangan yang bergerak turun dari atas layar.

Kode program ini terdiri dari beberapa class dan fungsi berikut penjelasannya :

- Pertama, mari kita mulai dengan melihat kelas `Player.java`. Kelas ini adalah jantung dari karakter pemain dalam game ini. Ia bertanggung jawab atas semua aspek yang terkait dengan pemain, termasuk posisi, gerakan, dan tampilan. Kelas `Player` memiliki beberapa properti penting. Koordinat `x` dan `y` mewakili posisi saat ini dari pemain di layar. `WIDTH` dan `HEIGHT` adalah ukuran tetap dari pemain, dalam hal ini adalah persegi berukuran 30x30 piksel.
- Fungsi `move()` adalah yang paling krusial. Fungsi ini memperbarui posisi horizontal pemain dengan menambahkan nilai `dx` (perubahan posisi horizontal) ke koordinat `x`. Namun, fungsi ini juga memastikan pemain tidak keluar dari batas kiri (`x = 0`) atau kanan (`x = 770`) layar, mencegah pemain bergerak di luar area yang ditetapkan.
- Fungsi `draw(Graphics2D g2d)` bertanggung jawab untuk menggambar pemain di layar. Ia menggunakan objek `Graphics2D` untuk menetapkan warna hijau dan menggambar persegi sesuai dengan posisi dan ukuran pemain saat ini.
- Fungsi `keyPressed(KeyEvent e)` dan `keyReleased(KeyEvent e)` menangani input keyboard dari pemain. Saat tombol panah kiri ditekan, `dx` diubah menjadi -2 untuk menggerakkan pemain ke kiri. Saat tombol panah kanan ditekan, `dx` diubah menjadi 2 untuk menggerakkan pemain ke kanan. Saat tombol-tombol tersebut dilepas, `dx` diubah menjadi 0 untuk menghentikan pergerakan pemain.
- Fungsi `getBounds()` mengembalikan objek `Rectangle` yang mewakili batas-batas pemain. Ini berguna untuk mendeteksi tabrakan antara pemain dan rintangan dalam game.
- Selanjutnya, kelas `Obstacle.java` menangani rintangan yang harus dihindari oleh pemain. Kelas ini menyimpan informasi tentang posisi, ukuran, dan kecepatan rintangan. Properti `x` dan `y` mewakili koordinat saat ini dari rintangan. `WIDTH` dan `HEIGHT` adalah ukuran tetap dari rintangan, dalam hal ini juga adalah persegi berukuran 30x30 piksel. `DY` adalah kecepatan turun rintangan, yang dalam kasus ini adalah 2 piksel per iterasi.
- Fungsi `move()` bertanggung jawab untuk memperbarui posisi rintangan dengan menambahkan nilai `DY` ke koordinat `y`. Ini menyebabkan rintangan bergerak turun secara terus-menerus.
- Fungsi `draw(Graphics2D g2d)` menggambar rintangan di layar menggunakan objek `Graphics2D`. Ia menetapkan warna merah dan menggambar persegi sesuai dengan posisi dan ukuran rintangan saat ini.
- Fungsi `getY()` hanya mengembalikan koordinat `y` dari rintangan, yang berguna untuk memeriksa apakah rintangan sudah keluar dari batas bawah layar. Fungsi `getBounds()` mengembalikan objek `Rectangle` yang mewakili batas-batas rintangan, yang juga berguna untuk mendeteksi tabrakan.

- Kelas `'Main.java'` adalah titik masuk utama program. Kelas ini bertanggung jawab untuk membuat jendela game dan menambahkan panel game (`'GamePanel'`) ke dalamnya. Ia memastikan jendela game tidak dapat diubah ukurannya dan menempatkannya di tengah layar. Akhirnya, ia memanggil `'gamePanel.startGame()'` untuk memulai permainan.
- Inti dari game ini terletak di kelas `'GamePanel.java'`. Kelas ini bertanggung jawab atas semua logika dan tampilan game. Konstruktor `'GamePanel()'` mengatur ukuran panel, warna latar belakang, dan tata letak.
- Fungsi `'initGame()'` menginisialisasi pemain, rintangan, timer, dan tombol mulai. Fungsi `'startGame()'` memulai permainan dengan mengatur `'inGame'` menjadi true, menyembunyikan tombol mulai, menambahkan listener keyboard, dan memulai timer.
- Fungsi `'gameOver()'` menghentikan permainan, menampilkan pesan "Game Over", dan memanggil `'resetGame()'` untuk mengatur ulang game. Fungsi `'resetGame()'` mengembalikan pemain ke posisi awal dan membersihkan daftar rintangan, serta menampilkan tombol mulai kembali.
- Fungsi `'paintComponent(Graphics g)'` adalah yang paling penting untuk menggambar game. Saat permainan sedang berlangsung, ia memanggil `'draw(g2d)'` pada pemain dan rintangan untuk menggambarkan mereka di layar. Saat permainan belum dimulai, ia menampilkan pesan "Press Start to Begin".
- Fungsi `'actionPerformed(ActionEvent e)'` dipanggil setiap kali timer dijalankan. Ini memperbarui posisi pemain, memperbarui rintangan, memeriksa tabrakan, dan meminta panel untuk dirender ulang.
- Fungsi `'updateObstacles()'` menambahkan rintangan baru secara acak ke daftar rintangan dan menggerakkan rintangan yang ada. Fungsi `'checkCollisions()'` memeriksa tabrakan antara pemain dan rintangan, dan memanggil `'gameOver()'` jika terjadi tabrakan.
- Terakhir, kelas `'TAdapter'` adalah kelas internal dalam `'GamePanel'` yang bertindak sebagai listener keyboard. Ia mengimplementasikan `'keyPressed()'` dan `'keyReleased()'` untuk menangani input pemain dan mengubah gerakan pemain sesuai dengan itu.

Secara keseluruhan, program ini mengimplementasikan game sederhana di mana pemain harus menghindari rintangan yang terus-menerus jatuh dari atas layar. Kelas-kelas utama yang terlibat, seperti `'Player'`, `'Obstacle'`, `'Main'`, dan `'GamePanel'`, masing-masing memiliki tanggung jawab dan fungsi yang jelas dan saling terkait untuk menciptakan pengalaman game yang utuh.

HASIL

