

LAPORAN UAS
PEMROGRAMAN BERORIENTASI OBJEK
Dosen Pengampu : Sayekti Harits Suryawan, S.kom., M.Kom.



KELOMPOK 17

Nama Anggota :

- | | |
|--------------------------------|------------------------|
| 1. Zidane Arryo | : 2211102441095 |
| 2. Dion Ikhzanza Rabbil | : 2211102441097 |
| 3. Muhammad Ulil Albab | : 2211102441104 |

S1 TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR
2023

Elemen-elemen yang mencakup pada Game bounce-battle :

1. Kelas dan Objek:

Setiap kelas seperti Level, PongWorld, background, bgl1, bgl2, restartWorld, Ball, BallMedium, Ballhard, Easy, Medium, Hard, Paddle, Player1, Player2, dan Scoreboard merepresentasikan objek. Ini memungkinkan untuk pemodelan yang jelas dan terstruktur, memisahkan fungsionalitas masing-masing.

2. Pewarisan (Inheritance):

Dalam kode ini, konsep pewarisan digunakan dengan bijak, terutama pada kelas Player1 dan Player2, yang merupakan turunan dari kelas Paddle. Pewarisan memudahkan penggunaan kembali penyusun kode dan penyusunan kelas.

3. Polimorfisme:

Polimorfisme terlihat jelas dalam penggunaan metode act() pada beberapa kelas seperti Ball, BallMedium, Ballhard, Player1, dan Player2. Meskipun semua kelas ini memiliki metode dengan nama yang sama, perilaku yang sebenarnya bervariasi tergantung pada kelas objek tersebut. Hal ini meningkatkan fleksibilitas dan memungkinkan perubahan perilaku dengan mudah.

- **Ball Class:**
Metode act() pada kelas Ball akan mengandung perilaku umum untuk bola dalam permainan, seperti pergerakan, deteksi tabrakan, dan respons terhadap interaksi dengan objek lain.
- **BallMedium Class:**
Metode act() di kelas BallMedium merupakan contoh polimorfisme karena, meskipun nama metodenya sama dengan act() di kelas induk Ball, perilaku aktualnya berbeda. Dalam konteks ini, kami mengganti pergerakan bola agar sedikit lebih cepat.
- **Ballhard Class:**
Metode act() di kelas Ballhard juga mengimplementasikan polimorfisme. Dalam hal ini, perilaku yang terkait dengan bola dengan menambah kecepatan yang lebih tinggi daripada BallMedium Class .
- **Player1 dan Player2 Classes:**
Metode act() di kelas Player1 dan Player2 menunjukkan polimorfisme. Meskipun keduanya memiliki metode dengan nama yang hampir sama, perilaku sebenarnya berbeda. Dalam konteks ini mencerminkan interaksi player1 menggunakan tombol “w” untuk bergerak ke atas dan “s” untuk bergerak kebawah Sedangkan Player 2 menggunakan tombol “up” untuk bergerak ke atas dan “down” untuk bergerak kebawah.

4. Enkapsulasi:

Beberapa properti dan metode dienkapsulasi dalam kelas-kelas. Misalnya, variabel `velX`, `velY`, `deltaT`, `player1Count`, `player2Count`, dan metode `scorePoints()` pada kelas `Ball`, `BallMedium`, dan `Ballhard` dienkapsulasi untuk melindungi data dan memastikan akses yang aman. Enkapsulasi ini mendukung prinsip informasi tersembunyi dan pemisahan antara antarmuka publik dan implementasi internal.

- **Ball Class:**

Variabel `velX` dan `velY` mungkin merupakan atribut enkapsulasi. Ini berarti bahwa data kecepatan pergerakan bola dienkapsulasi di dalam kelas `Ball`, dan akses langsung dari luar kelas ke variabel tersebut tidak diizinkan. Penggunaan metode `get` dan `set` khusus, seperti `getVelocityX()`, `getVelocityY()`, `setVelocityX()`, dan `setVelocityY()`, dapat digunakan untuk mengakses dan memodifikasi nilai-nilai ini secara aman.

- **BallMedium dan Ballhard Classes:**

Konsep enkapsulasi pada variabel dan metode juga dapat diterapkan di kelas ini. Mungkin ada atribut tambahan yang dienkapsulasi untuk mendukung perilaku bola medium dan bola sulit, dan metode `act()` juga dapat dianggap sebagai bagian dari enkapsulasi untuk menjaga privasi implementasi internal.

- **Scoreboard Class:**

Dalam kelas `Scoreboard`, variabel `player1Count` dan `player2Count` dienkapsulasi. Akses langsung dari luar untuk memodifikasi atau membaca skor pemain dapat dihindari dengan menggunakan metode seperti `getPlayer1Count()` dan `getPlayer2Count()`.

- **Paddle, Player1, dan Player2 Classes:**

Properti seperti posisi pemain yang dienkapsulasi di dalam kelas-kelas ini. Metode yang berhubungan dengan pergerakan pemain dapat dianggap sebagai metode enkapsulasi.

5. Interaksi Antar Objek:

Terdapat interaksi yang dinamis antar objek dalam permainan. Bola berinteraksi dengan pemain (`Player1` dan `Player2`) dan dinding. Interaksi ini memungkinkan untuk menciptakan lingkungan permainan yang dinamis dan menarik.

6. Overriding dan Overloading:

Overriding terjadi pada metode `act()` di kelas-kelas turunan seperti `BallMedium` dan `Ballhard`, yang mengubah perilaku dari metode `act()` pada kelas induk `Ball`. Meskipun overloading tidak kami gunakan, penggunaan overriding menggantikan perilaku metode dari superclass dengan perilaku yang lebih spesifik.