

```
int dinosaurHeight = 94;
int dinosaurX = 50;
int dinosaurY = boardHeight - dinosaurHeight;
```

```
Block dinosaur;
```

```
//cactus
int cactus1Width = 34;
int cactus2Width = 69;
int cactus3Width = 102;
```

```
int cactusHeight = 70;
int cactusX = 700;
int cactusY = boardHeight - cactusHeight;
ArrayList<Block> cactusArray;
```

Variabel-variabel ini digunakan untuk mengatur dimensi dan posisi awal dari dinosaurus dan kaktus. Dinosaurus diwakili oleh objek Block dengan gambar dinosaurImg, sedangkan kaktus-kaktus ditampung dalam ArrayList cactusArray.

#### 4. Fisika Permainan

```
//physics
int velocityX = -12; //cactus moving left speed
int velocityY = 0; //dinosaur jump speed
int gravity = 1;
```

Variabel-variabel ini mengatur dinamika permainan:

- velocityX menentukan kecepatan pergerakan kaktus ke kiri.
- velocityY menentukan kecepatan lompatan dinosaurus.
- gravity adalah percepatan gravitasi yang digunakan untuk membuat dinosaurus jatuh saat tidak melompat.

#### 5. Timer dan Pengaturan Permainan

```
Timer gameLoop;
Timer placeCactusTimer;
```

```
public ChromeDinosaur() {
// ... konstruktor lainnya ...
```

```
//game timer
gameLoop = new Timer(1000/60, this); //1000/60 = 60 frames per 1000ms (1s), update
gameLoop.start();
```

```
//place cactus timer
placeCactusTimer = new Timer(1500, new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
placeCactus();
}
});
placeCactusTimer.start();
}
```

Di konstruktor, dua Timer diinisialisasi:

- gameLoop adalah timer untuk memperbarui permainan setiap 1/60 detik (60 kali per detik).
- placeCactusTimer adalah timer untuk menempatkan kaktus secara acak di layar setiap 1.5 detik.

## 6. Metode placeCactus

```
void placeCactus() {
if (gameOver) {
return;
}

double placeCactusChance = Math.random(); //0 - 0.999999
if (placeCactusChance > .90) { //10% you get cactus3
Block cactus = new Block(cactusX, cactusY, cactus3Width, cactusHeight, cactus3Img);
cactusArray.add(cactus);
}
else if (placeCactusChance > .70) { //20% you get cactus2
Block cactus = new Block(cactusX, cactusY, cactus2Width, cactusHeight, cactus2Img);
cactusArray.add(cactus);
}
else if (placeCactusChance > .50) { //20% you get cactus1
Block cactus = new Block(cactusX, cactusY, cactus1Width, cactusHeight, cactus1Img);
cactusArray.add(cactus);
}

if (cactusArray.size() > 10) {
cactusArray.remove(0); //remove the first cactus from ArrayList
}
}
```

Metode placeCactus digunakan untuk menempatkan kaktus secara acak berdasarkan peluang tertentu (placeCactusChance). Jika gameOver tidak terjadi, sebuah objek Block dengan gambar kaktus yang sesuai ditambahkan ke cactusArray. Jika ukuran

cactusArray melebihi 10, kaktus pertama dihapus dari daftar.

## 7. Metode move dan Deteksi Tabrakan (collision)

```
public void move() {  
    // ... kode pergerakan dinosaurus dan kaktus ...  
  
    //score  
    score++;  
}  
  
boolean collision(Block a, Block b) {  
    return a.x < b.x + b.width &&  
        a.x + a.width > b.x &&  
        a.y < b.y + b.height &&  
        a.y + a.height > b.y;  
}
```

Metode move mengatur pergerakan dinosaurus (berdasarkan gravitasi dan kecepatan velocityY) dan kaktus (ke kiri dengan kecepatan velocityX). Pada saat yang bersamaan, skor permainan juga terus bertambah. Metode collision digunakan untuk mendeteksi tabrakan antara dinosaurus (a) dan kaktus (b) berdasarkan posisi dan ukuran mereka.

## 8. Metode paintComponent dan Gambar

```
public void paintComponent(Graphics g) {  
    super.paintComponent(g);  
    draw(g);  
}  
  
public void draw(Graphics g) {  
    // ... kode gambar dinosaurus, kaktus, dan skor ...  
  
    //score  
    g.setColor(Color.black);  
    g.setFont(new Font("Courier", Font.PLAIN, 32));  
    if (gameOver) {  
        g.drawString("Game Over: " + String.valueOf(score), 10, 35);  
    }  
    else {  
        g.drawString(String.valueOf(score), 10, 35);  
    }  
}
```

Metode paintComponent digunakan untuk menggambar komponen-komponen permainan. Metode draw menggambar dinosaurus, kaktus, dan skor ke layar. Jika



## 1. Deklarasi Variabel dan Gambar

```
int boardWidth = 750;
int boardHeight = 250;

//images
Image dinosaurImg;
Image dinosaurDeadImg;
Image dinosaurJumpImg;
Image cactus1Img;
Image cactus2Img;
Image cactus3Img;
```

Variabel `boardWidth` dan `boardHeight` digunakan untuk menentukan ukuran panel permainan.

Gambar-gambar yang diperlukan untuk permainan (`dinosaurImg`, `dinosaurDeadImg`, `dinosaurJumpImg`, `cactus1Img`, `cactus2Img`, `cactus3Img`) dimuat menggunakan `ImageIcon` dari sumber daya yang terletak dalam direktori `./img/`.

## 2. Kelas Block sebagai Representasi Objek dalam Permainan

```
class Block {
    int x;
    int y;
    int width;
    int height;
    Image img;

    Block(int x, int y, int width, int height, Image img) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
        this.img = img;
    }
}
```

Kelas `Block` digunakan untuk merepresentasikan objek-objek dalam permainan, seperti dinosaurus dan kaktus. Setiap `Block` memiliki posisi (`x`, `y`), ukuran (`width`, `height`), dan gambar (`img`) yang digunakan untuk menampilkannya di layar.

## 3. Inisialisasi Objek Dinosaurs dan Kaktus

```
//dinosaur
int dinosaurWidth = 88;
```

```
17 // frame.setVisible(true);  
18 frame.setSize(boardWidth, boardHeight);  
19 frame.setLocationRelativeTo(c:null);  
20 frame.setResizable(resizable:false);
```

Chrome Dinosaur

Game Over: 75



```
PS C:\Users\lenovo\Downloads\chrome-dinosaur-java-master> ^C  
PS C:\Users\lenovo\Downloads\chrome-dinosaur-java-master>  
PS C:\Users\lenovo\Downloads\chrome-dinosaur-java-master> c:: cd 'c:\Users\lenovo\Downloads\chrome-dinosaur-  
Java-master\Program Files\Java\jdk-17\bin\java.exe' -cp .\src\main\resources;.\src\main\classes
```

gameOver, skor ditampilkan dengan pesan "Game Over: " di depannya.

## 9. Handling Input Keyboard (KeyListener)

```
@Override
public void keyPressed(KeyEvent e) {
    if (e.getKeyCode() == KeyEvent.VK_SPACE) {
        if (dinosaur.y == dinosaurY) {
            velocityY = -17;
            dinosaur.img = dinosaurJumpImg;
        }
    }
}
```

```
if (gameOver) {
    //restart game by resetting conditions
    dinosaur.y = dinosaurY;
    dinosaur.img = dinosaurImg;
    velocityY = 0;
    cactusArray.clear();
    score = 0;
    gameOver = false;
    gameLoop.start();
    placeCactusTimer.start();
}
}
}
```

```
@Override
public void keyTyped(KeyEvent e) {}
```

```
@Override
public void keyReleased(KeyEvent e) {}
```

Implementasi dari KeyListener untuk menangani input dari keyboard. Saat tombol SPACE ditekan, dinosaurus melompat jika berada di tanah (dinosaur.y == dinosaurY). Jika permainan berakhir (gameOver), kondisi permainan di-reset untuk memulai permainan baru.