Agate Mobile Game Developer Camp

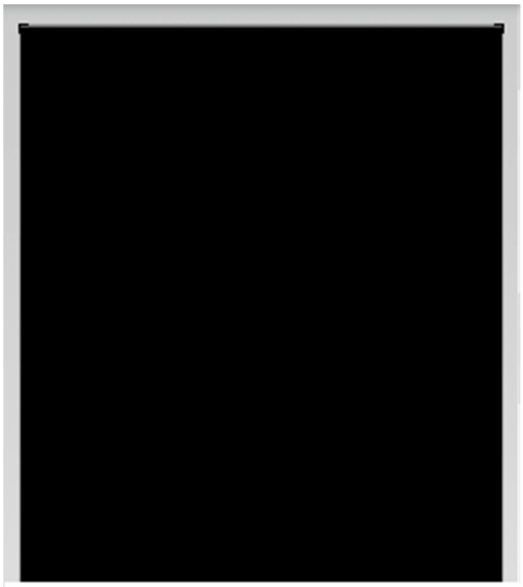
Programmer Handout – Day #2

Pada kesempatan kali ini kita akan mempelajari tentang Collision Detection dan Simple AI. Untuk mempermudahnya, maka akan kita coba membuat game sederhana, yaitu Pong. Pong merupakan game 2D dimana ada 2 buah pemukul dan sebuah bola. 1 pemukul akan kita kendalikan, sedangkan pemukul yang lain akan dikendalikan oleh AI.

Aturan Pong:

1. Pemukul harus menjaga agar bola tidak melewati daerahnya.
2. Bola akan terus memantul hingga bola melewati daerah lawan.
3. Daerah samping selain tempat yang harus dijaga oleh pemukul, hanya akan memantulkan bola saja.

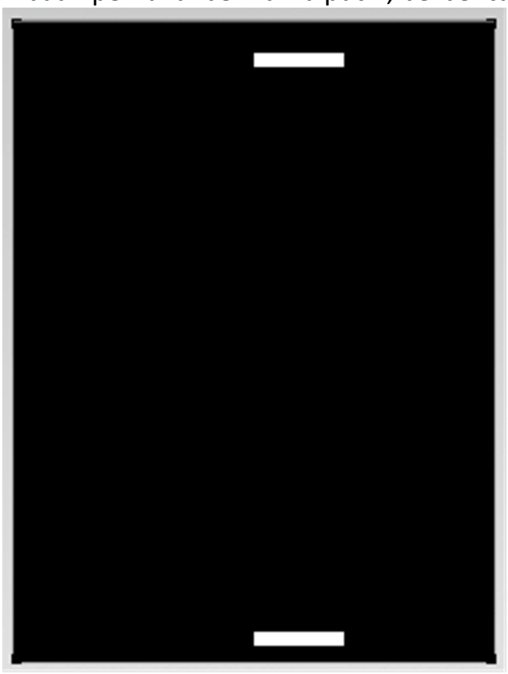
Pertama - tama, mari kita identifikasikan kebutuhan gambarnya.



```
g.setColor(0, 0, 0);
g.fillRect(0, 0, getWidth(), getHeight());
```

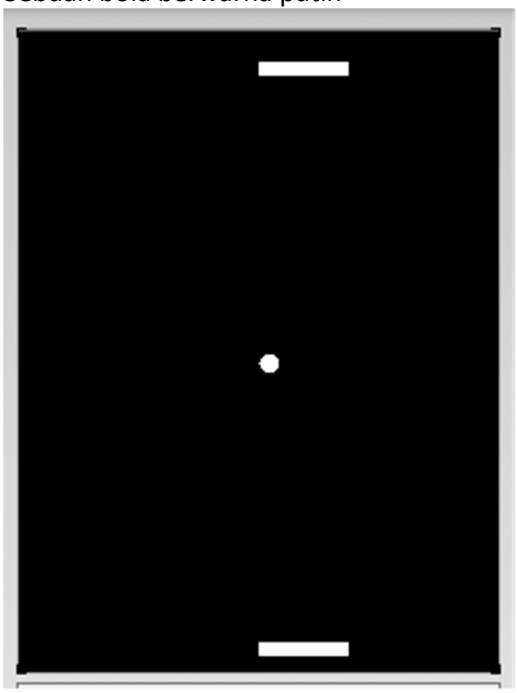
1. Background berwarna hitam.

2. 2 buah pemukul berwarna putih, berbentuk persegi panjang.



```
//Warna putih
g.setColor(255, 255, 255);
//Pad musuh
g.fillRect(getWidth()/2, 15, 45, 7);
//Pad kita
g.fillRect(getWidth()/2, getHeight()-15, 45, 7);
```

3. Sebuah bola berwarna putih



```
//Bola berwarna putih
g.fillArc(getWidth()/2, getHeight()/2, 10, 10, 0, 360);
```

Setelah membuat gambar statis seperti di atas, mari kita kembangkan agar pad dan bola kita bisa bergerak. Bagaimana caranya? Pertama – tama kita harus membuat agar posisi dari pad dan bola dinamis.

1. Pad

   Karena kita hanya akan menggerakkan Pad secara horizontal, maka posisi awal X yang akan kita manipulasi. Kita akan membuat variable baru yaitu **padX**, nantinya pada fillRect untuk menggambar Pad kita, akan seperti ini.

```
//Variabel posisi Pad kita
private int pad1X1 = getWidth()/2;
private int pad1Y1 = getHeight();
private int pad1X2 = 45;
private int pad1Y2 = 7;
//Variabel posisi Pad musuh
private int pad2X1 = getWidth()/2;
private int pad2Y1 = 15;
private int pad2X2 = 45;
private int pad2Y2 = 7;
```

   Lalu kita buat kelas Pad1() dan Pad2()

```
//Pad kita
private void Pad1(){
    g.setColor(255, 255, 255);
    g.fillRect(pad1X1, pad1Y1, pad1X2, pad1Y2);
}
//Pad musuh
private void Pad2(){
    g.setColor(255, 255, 255);
    g.fillRect(pad2X1, pad2Y1, pad2X2, pad2Y2);
}
```

   Sehingga ketika kita tambah code berikut di game loop

```
public void run() {
    while(true) {
        createBackground(g);
        Pad1();
        Pad2();
        padXKita+=1;
        padXMusuh+=1;
        System.out.println(padXKita);
        try {
            Thread.sleep(sleepTime);
        } catch (Exception e) {
        }
        flushGraphics();
    }
}
```

Maka sudah dipastikan pad kita posisinya akan selalu bertambah, yang artinya akan bergerak ke kanan terus. Sekarang mari kita coba mix dengan user inputnya. Jika kita tekan kanan, maka pad akan bergerak ke arah kanan, dan jika kita tekan kiri, maka pad akan bergerak ke arah kiri.

Pertama – tama, deklarasikan dahulu kecepatan dari pad,

```
private final static int padXVel = 5;
```

Lalu kita buat kelas gerakPad(),

```
private void gerakPad() {
    int keyState = getKeyStates();

    if ((keyState & LEFT_PRESSED) != 0) {
        pad1X1 -= padXVel;
    } else if ((keyState & RIGHT_PRESSED) != 0) {
        pad1X1 += padXVel;
    }
}
```

Ketika kita tekan kiri, maka **padXKita** akan dikurangi dengan **padXVel**, sehingga lokasi dari pad akan berubah. Begitu pula sebaliknya ketika kita tekan kanan.

Nah lho, ketika pad kita sudah mencapai ujung, ternyata pad akan terus berjalan. Bahkan terus hingga tak hingga. Untuk mencegah hal itu, maka akan kita buat pengecekan ketika pad sudah mencapai ujung. Sehingga pad akan berhenti. Kita akan simpan kode berikut di dalam fungsi gerakPad(),

```
if (pad1X1 <= 0) {
    pad1X1 = 0;
}else if ((pad1X1 + 45) >= getWidth()) {
    pad1X1 = getWidth() - 45;
}
```

Yup, selesailah untuk pad kita. Untuk pengembangan pad musuh, nanti kita akan tambahkan simple AI di dalamnya.

2. Bola

Tugas sebuah bola adalah bergerak secara acak, memiliki efek pantulan, dan ketika melewati daerah lawan, maka akan menambah skor kita. Berarti, posisi X dan Y dari bola kita harus bersifat dinamis. Pertama – tama, deklarasikan posisi X dan Y dari bola, kita buat variable global bolaX dan bolaY.

```
private int bolaX = getWidth() / 2;
private int bolaY = getHeight() / 2;
```

Lalu buat kelas Bola() yang isinya definisi warna dan fillArc()

```
//Bola
private void Bola(){
    g.setColor(255, 255, 255);
    g.fillArc(bolaX, bolaY, 7, 7, 0, 360);
}
```

Lalu kita buat kelas agar si Bola bergerak. Terlebih dahulu definisikan **bolaXVel** dan **bolaYVel** untuk menentukan arah dan kecepatan bolanya.

```
private int bolaXVel = 3;
private int bolaYVel = 1;
```

Buat kelas gerakBola()

```
private void gerakBola() {
    bolaY+=bolaYVel;
    if(bolaX + 7> getWidth() || bolaX < 0)
    {
        bolaXVel*=-1;
    }
    bolaX+=bolaXVel;
    if(bolaY + 7> getHeight() || bolaY < 0)
    {
        bolaYVel*=-1;
    }

    if(checkCollision()){
        bolaYVel*=-1;;
    }
}
```

Tadi kita sudah menetapkan **bolaX += bolaXVel** dan **bolaY -= bolaYVel,** sehingga bola akan bergerak sesuai dengan **bolaXVel** dan **bolaYVel**. Kode di atas juga telah menangani ketika bola menabrak dinding, jika bola > getWidth (nabrak ke batas kanan) atau bola < 0 (nabrak batas kiri) maka **bolaXVel** *= -1, dengan kata lain bola akan berubah geraknya ke arah yang berlawanan.

Lalu mari kita buat pantulan terhadap Pad kita, buat kelas baru **checkCollision()**,

```
private boolean checkCollision(){
    boolean collide = false;
    if( (bolaX + 7 >pad1X1) &&
        (bolaX + 7 <=(pad1X1 + pad1X2)) &&
        (bolaY + 7 > pad1Y1) &&
        (bolaY + 7 <= (pad1Y1 + pad1Y2))
        ){
        collide=true;
    }
    return collide;
}
```

Statement IF di atas berguna untuk mengecek apakah bola memasuki daerah Pad kita atau tidak, jika YA maka collide = **true**. Lalu kita taruh sedikit sintax dibawah ke dalam fungsi **gerakBola()**.

```
if(checkCollision()){
    bolaYVel*=-1;
}
```

Jika collide tadi bernilai true, maka **bolaYVel** akan dikalikan dengan -1, sehingga bola pun akan memantul.

Nah setelah membuat pad kita bergerak dan bola kita memantul kesana kemari, saatnya kita membuat Pad AI. Ya ini hanya simple AI, dimana pad ini akan bergerak mengikuti arah dari bola. Dan kita juga akan mempersempit jarak pandang dari AI, agar permainan tidak dikuasai oleh AI terus :D. Ingat pad yang terletak di atas? Nah itu adalah pad AI kita.

Pertama – tama, mari kita deklarasikan kebutuhan variable kita.

```
private final static int padAiVel = 5;
```

Sudah jelas dilihat dari nama variable nya. **padAiVel** berguna untuk mendefinisikan kecepatan dari pad AI kita. Kita bisa menambahkan atau menurunkan kecepatannya sesuka kita. Lalu kita buat fungsi **gerakAI()**

```
private void gerakAI() {
    //pengecekan
    if (bolaX < pad2X1 + 23 && bolaY < getHeight() - getWidth())
        pad2X1 -= padAiVel;
    if (bolaX > pad2X1 + 23 && bolaY < getHeight() - getWidth())
        pad2X1 += padAiVel;

    //Cek apakah Pad sudah berada di ujung batas monitor
    if (pad2X1 <= 0) {
        pad2X1 = 0;
    }else if ((pad2X1 + 45) >= getWidth()) {
        pad2X1 = getWidth() - 45;
    }
}
```

Fungsi IF pertama, mengecek apakah posisi **bolaX** < **pad2X1** DAN **bolaY** < **getHeight**() – **getWidth**()? Jika YA, maka pad2X1 -= padAiVel, yang artinya jika bolaX (posisi X bola) < koordinat kiri atas pad AI (**pad2X1**, pad2Y1), (pad2X2, pad2Y2) DAN bolaY (posisi Y bola) < getHeight (mis. 320) – getWidth (mis. 240) dengan kata lain akan mengecek apakah bolaY berada dalam jarak pandang AI (320 – 240 = 80), jika YA

maka pad akan bergerak ke kiri sesuai dengan kecepatannya. Begitu pula sebaliknya dengan IF dibawahnya. Pada IF-ELSE IF berikutnya, memiliki fungsi yang sama dengan IF yang ada pada **gerakPad()**, yaitu untuk mengecek apakah Pad sudah berada di ujung atau tidak. Meskipun sekarang pad sudah bisa bergerak, tapi bolanya tetap belum bisa mantul ketika menyentuh pad AI. Kita perlu tambahkan kode berikut di fungsi **checkCollision()**,

```
if( (bolaX >pad2X1) &&
    (bolaX <=(pad2X1 + pad2X2)) &&
    (bolaY > pad2Y1) &&
    (bolaY <= (pad2Y1 + pad2Y2))
    ){
    collide=true;
}
```

Intinya sama dengan pengecekan collision pada pad kita. Nah beres sudah game Pong ini, tentu saja akan ada tantangan untuk kalian >:)

Tantangan di hari kedua.

Tambahkan fitur – fitur ini:

- Skor, ketika ada yang mencapai nilai 10, maka game end, restart dari permulaan game.
- Peningkatan level permainan, tambahkan kecepatan bola setelah bola menyentuh pad kita 3x dan kelipatannya.