**Modul 3**

1. **Tujuan**

● Mahasiswa mampu memahami dan mengimplementasikan konsep Frame Rate pada pygame

● Mahasiswa mampu memahami dan mengimplementasikan pergerakan sebuah benda (karakter) pada pygame windows

● Mahasiswa mampu memahami dan mengimplementasikan konsep vektor pada pygame

1. **Dasar teori**
2. Frame Rate

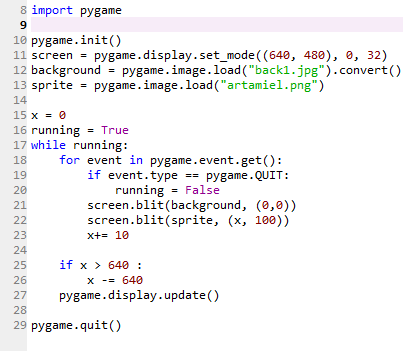
Hal pertama yang perlu kita ketahui tentang gerakan dalam permainan komputer adalah bahwa pergerakan benda (animasi) dalam layar tidak secara harafiah benda tersebut bergerak. Layar komputer atau pesawat televisi menyajikan kepada kita urutan gambar, dan ketika waktu antara gambar cukup pendek, otak kita memadukan gambar bersama-sama untuk menciptakan ilusi gerakan yang mengalir. Jumlah gambar, atau bingkai, yang diperlukan untuk menghasilkan gerakan halus dapat bervariasi dari orang ke orang. Film menggunakan 24 frame per detik, tetapi game komputer cenderung membutuhkan frame rate yang lebih cepat. Tiga puluh frame per detik adalah kecepatan yang umum digunakan, tetapi semakin tinggi frame rate, semakin halus gerakan akan terlihat. Meskipun setelah sekitar 70 frame per detik, hanya sedikit orang yang dapat mendeteksi peningkatan apa pun

Frame Rate untuk game dibatasi oleh berapa kali perangkat tampilan (seperti monitor Anda) dapat merefresh tiap detik. Misalnya, monitor LCD monitor memiliki kecepatan refresh 60 hertz, yang berarti monitor ini menyegarkan tampilan 60 kali setiap detik. Frame rate yang lebih cepat dari kecepatan refresh dapat menyebabkan apa yang dikenal sebagai "*Tearing*", di mana bagian dari frame berikutnya digabungkan dengan frame sebelumnya.

Mendapatkan frame rate yang baik biasanya dilakukan dengan mengorbankan efek visual, karena semakin banyak beban kerja yang dilakukan komputer, semakin lambat frame ratenya. Hal ini tentu saja dapat diatasi menggunakan perangkat komputer yang lebih canggih untuk menghasilkan visual yang diinginkan.

* 1. Pergerakan (Movement)
* Pergerakan Lurus (Moving in Straight Line)

Mari kita mulai dengan memeriksa gerakan garis lurus sederhana. Jika kita memindahkan gambar dengan jumlah yang tetap setiap frame, maka gambar itu akan tampak bergerak. Untuk memindahkannya secara horizontal, kita akan menambahkan koordinat x, dan untuk memindahkannya secara vertikal, kita akan menambahkan koordinat y.



Program diatas menunjukkan cara memindahkan gambar secara horizontal. Ini menggambar gambar pada koordinat x yang ditentukan dan kemudian menambahkan nilai 10,0 ke setiap frame, sehingga pada frame berikutnya akan bergeser sedikit ke kanan. Ketika koordinat x melewati tepi kanan layar, itu diatur kembali ke 0 sehingga tidak hilang sepenuhnya.

Permasalahan dari kode program diatas adalah kita tidak tahu berapa waktu yang diperlukan agar objek bergerak. Trik untuk memecahkan masalah ini adalah membuat gerakan berbasis waktu. Kita perlu mengetahui berapa banyak waktu yang telah berlalu sejak frame sebelumnya sehingga kita dapat memposisikan semuanya di layar sesuai dengan itu. Modul pygame.time berisi objek yang dapat kita gunakan untuk melacak waktu. Untuk membuat objek jam, panggil konstruktornya pygame.time.Clock:



Setelah memiliki objek clock, kita harus memanggil fungsi *tick()* satu kali tiap frame, dimana fungsi tersebut mengembalikan waktu yang berlalu sejak panggilan sebelumnya dalam milidetik (ada 1.000 milidetik dalam satu detik):

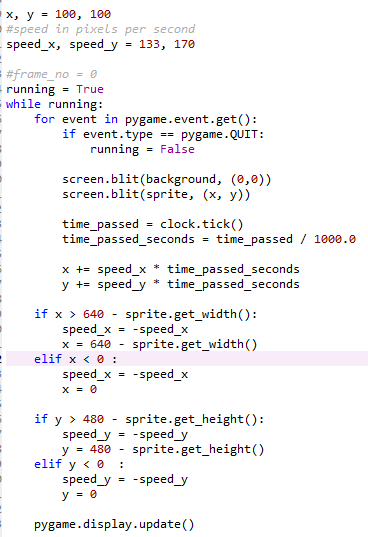


Fungsi tick() mengambil parameter opsional untuk kecepatan frame yang maksimum. Kita dapat mengatur parameter ini sehingga game/aplikasi yang berjalan di desktop komputer tidak memakan banyak sumber daya.



Milidetik sering digunakan untuk menghitung waktu *event* dalam game karena lebih mudah menangani nilai bilangan bulat daripada waktu pecahan. Penggunaan milidetik biasanya cukup akurat untuk sebagian besar tugas dalam games.

* Pergerakan Diagonal



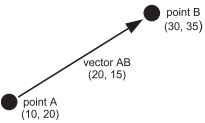
Untuk membuat objek bergerak secara diagonal, pertama-tama kita harus mendeteksi bahwa kita telah mencapai tepi layar. Jika koordinat x kurang dari 0, kita tahu bahwa kita telah melewati sisi kiri layar karena koordinat tepi kiri adalah 0. Jika x ditambah lebar sprite lebih besar dari lebar layar, kita tahu bahwa tepi kanan sprite telah mengenai tepi kanan layar. Kode untuk koordinat y serupa, tetapi kami menggunakan tinggi sprite daripada lebar.

* Exploring Vector

Vektor merupakan salah satu metode yang digunakan oleh pengembang game untuk membuat sebuah pergerakan objek, baik dalam game 2D maupun 3D. Konsep vektor mirip dengan titik koordinat karena keduanya memiliki nilai untuk x dan y (dalam 2D), tetapi memiliki tujuan yang berbeda. Titik pada koordinat (10, 20) akan selalu menjadi titik yang sama di layar. Tetapi vektor (10, 20) berarti menambahkan 10 ke koordinat x dan 20 ke koordinat y dari posisi saat ini.

* Create Vector

Anda dapat menghitung vektor dari dua titik mana pun dengan mengurangkan nilai di titik pertama dari titik kedua.



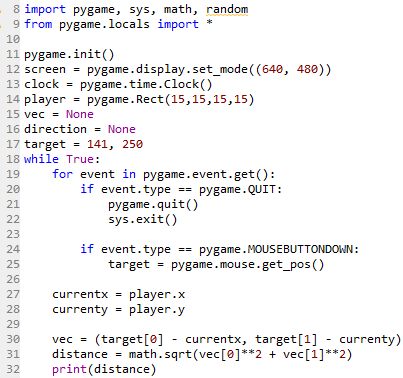
Materi lanjutan terkait vector dapat dilihat pada dokumentasi pygame dan Materi 1 di spada

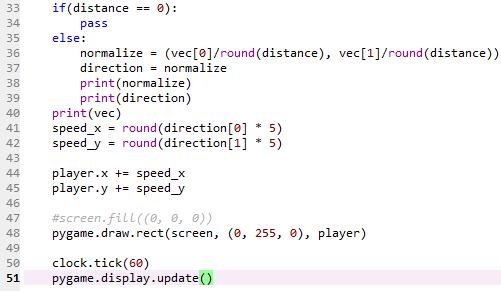
<https://www.pygame.org/docs/ref/math.html>

1. Kesimpulan

Menggerakkan objek, atau apa pun di layar, mengharuskan kita menambahkan nilai ke koordinat pada setiap frame. Tetapi jika menginginkan gerakan yang halus dan konsisten, pergerakan karakter harus didasarkan pada waktu tertentu (waktu awal ke waktu sejak frame terakhir).

1. **Contoh Program**
   * Menggambar garis bebas menggunakan vektor

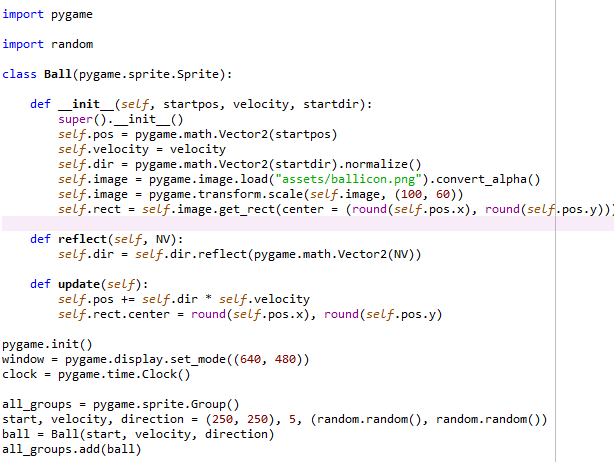


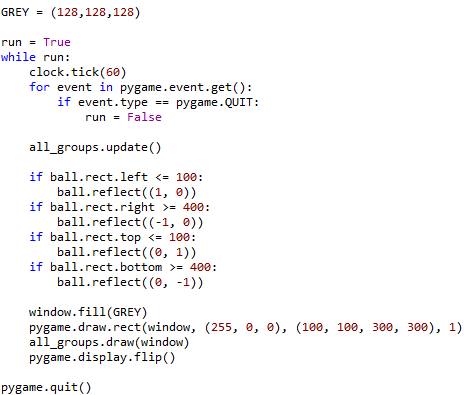


Hasil program :

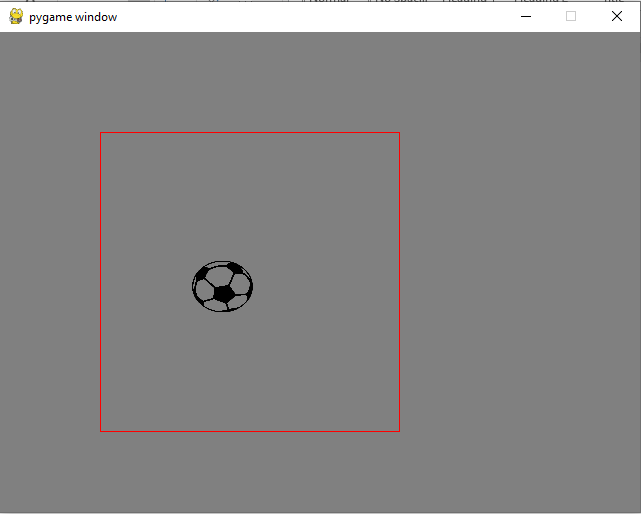


* + Vector Random Movement





Hasil Program :



**D. TUGAS INDIVIDU**

1. Cobalah program pada poin C. Kemudian analisis kedua program menggunakan bahasa sendiri di Ms. Word.

2. Pengumpulan Tugas Praktikum.

* **Untuk kelas TI E** Tugas Praktikum di kumpulkan di SPADA paling lambat tanggal **14 September 2021 jam 23.59**

**Untuk kelas TI D** Tugas Praktikum di kumpulkan di SPADA paling lambat tanggal **15 September 2021 jam 23.59**

* Format penamaan file SKD\_namakelas\_nim\_nama